



**Politecnico
di Torino**

ScuDo

Scuola di Dottorato - Doctoral School

WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation

Doctoral Program in Physics (38th cycle)

Interpretable and Controllable Statistical Inference Models for Protein Structure Prediction and Sequence Design

By

Francesco Caredda

Supervisor(s):

Prof. Andrea Pagnani

Doctoral Examination Committee:

Prof. Guido Tiana, Referee, Università degli Studi di Milano

Prof. Martin Weigt, Referee, Sorbonne Université Paris

Prof. Alfredo Braunstein, Internal Member, Politecnico di Torino

Politecnico di Torino

2026

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Francesco Caredda
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

ai miei genitori

Acknowledgements

I would like to deeply thank all the people who walked with me along this hard and uneven path.

In particular, to my supervisor, Andrea, for giving me the freedom to make my own decisions and the support I needed when some of those decisions proved to be wrong, both in research and in life.

To Paolo, who welcomed me to his lab and showed me how collaborations sometimes can give you new strengths and perspectives. To Lisa, Maria, Marco, and Toby, for making my stay in Lausanne lighter and full of memories.

To all my colleagues, old and new. To Mattia, Luca, Niccolò, Damiano, and Federico, for all the almost-serious discussions on whatever topic came up at lunchtime in the "Saletta". To the old guard that welcomed me into the group since my earliest days as a master student: Anna, Luca, Elisa, Matteo, Stefano, and Matteo.

To everyone I met at conferences, schools, and workshops throughout these years, for the many inspiring conversations and all the shared memories. In particular, Roberto, with whom I walked a parallel path since our days in Paris, and I almost murdered during a school in Cargese..

To all my friends in Torino and to all the people I met over the last three years who helped make this city feel a little more like home.

Finally, to my closest friends, the people who never fail to support me during my darkest moments: Eleonora, Michela, and Gabriele. Thank you.

Abstract

Proteins are among the fundamental molecular components of living systems, carrying out a wide range of structural, catalytic, and regulatory functions that are essential for life at any level. Understanding, predicting, and ultimately controlling the relationship between protein sequence, structure, and function is a central problem in modern biology, with profound implications for medicine, biotechnology, and fundamental science. Over the last two decades, the advent of high-throughput sequencing technologies and the explosion of genomic data have made it possible to approach this problem from a data-driven perspective, first through the tools of inferential statistical physics and, more recently, through machine learning. While these two frameworks have always been historically intertwined, they present different trade-offs: statistical physics models are typically interpretable and grounded in explicit probabilistic principles, whereas modern machine learning architectures offer greater expressive power at the cost of transparency and interpretability. In this thesis, we study this trade-off by developing and analysing extensions of statistical physics models that incorporate structural elements inspired by modern machine learning models, with the goal of improving expressivity while preserving a clear statistical interpretation.

After introducing the biological background in Chapter 1, we review in Chapter 2 the statistical physics approach to protein sequence modeling, focusing in particular on Direct Coupling Analysis (DCA) as a maximum-entropy framework for inferring evolutionary constraints from multiple sequence alignments. In Chapter 3, we turn to machine learning models, tracing the progression from early sequence-based approaches to transformer architectures and large-scale models such as AlphaFold2, which have revolutionized protein structure prediction.

Building on these foundations, Chapter 4 presents the first original contribution of this thesis: AttentionDCA, a model that bridges statistical physics and deep

learning by interpreting the attention mechanism as a structured interaction term analogous to Potts couplings. This perspective provides a low-rank parametrization of DCA, enabling a reduction in model complexity while preserving predictive performance, and offering a principled framework to analyze the internal representations of transformer-based architectures. In Chapter 5, this approach is extended by exploiting the flexibility of the attention-based parametrization to decouple and control different components of the interaction model, naturally enabling its application to protein–protein interaction modeling and paralog matching.

Finally, in Chapter 6, we introduce FeatureDCA, a second contribution that extends autoregressive DCA to a conditional generative framework. By coupling sequence probabilities to low-dimensional, biologically meaningful features, FeatureDCA enables controlled sampling of protein sequences in specific regions of sequence space, while maintaining the interpretability and statistical grounding of DCA-based models. This formulation provides a transparent alternative to latent-variable approaches commonly used in deep learning, and demonstrates that controllable generative modeling can be achieved within a shallow statistical framework.

Overall, this work shows that the apparent dichotomy between interpretability and expressivity can be partially reconciled by reinterpreting modern machine learning architectures through the lens of statistical physics and, conversely, by extending classical models with structured and controllable representations drawn from machine learning.

Contents

List of Figures	xii
------------------------	------------

List of Tables	xvi
-----------------------	------------

1 Introduction	1
1.1 The biology of proteins	1
1.1.1 Chemical composition	2
1.1.2 Levels of protein structure	3
1.1.3 Structure-function relationship	6
1.2 The protein folding and protein design problems	6
1.2.1 The protein folding problem	7
1.2.2 The protein design problem	8
1.3 Statistical Physics and Deep Learning Approaches	9
1.3.1 Statistical physics perspective	9
1.3.2 Deep learning perspective	10
1.4 Research objectives and aims	12
2 Statistical Inference Models for Protein Sequences	15
2.1 Evolutionary Constraints	15
2.1.1 Conservation and Co-evolution	15
2.1.2 Multiple Sequence Alignments	17

2.2	The Principle of Maximum Entropy	19
2.3	The Potts Model for Protein Sequences	21
2.3.1	Maximum Entropy Construction	22
2.3.2	Statistical Physics Interpretation	23
2.3.3	Gauge Freedom	24
2.4	Direct Coupling Analysis	26
2.4.1	Maximum Likelihood Inference	26
2.4.2	Mean-Field Approximation	28
2.4.3	Gaussian Approximation	29
2.4.4	Boltzmann Machine Learning	30
2.4.5	Pseudo-likelihood Maximisation	32
2.4.6	Autoregressive Inference	34
2.5	Regularization and Sequence Re-weighting	35
2.5.1	Sequence re-weighting	36
2.5.2	Regularization	37
2.6	Applications of Direct Coupling Analysis	38
2.6.1	Contact Prediction	38
2.6.2	Generative Modeling	40
2.6.3	Prediction of Mutational Effects	42
2.6.4	Other Applications	44
3	Protein Language Modeling and the Attention Mechanism	46
3.1	Natural Language Processing	46
3.1.1	Early statistical and shallow models	47
3.1.2	Neural sequence models before attention	49
3.1.3	Attention	52
3.1.4	Attention as a soft dictionary	55

3.1.5	Transformer	57
3.2	Protein Language Modeling	63
3.2.1	From languages to proteins	63
3.2.2	Early protein language models	64
3.2.3	AlphaFold	66
3.3	Attention and the Generalized Potts Model	70
3.3.1	Generalized Potts Model and Self-Attention	71
3.3.2	Factored Attention	73
4	AttentionDCA: Interpretable Attention-based Direct Coupling Analysis	75
4.1	Implementation	76
4.1.1	Standard version	76
4.1.2	Multi-family version	79
4.1.3	Autoregressive version	81
4.2	Results	82
4.2.1	Parameter reduction and hyper-parameters	83
4.2.2	Standard and multi-family contact prediction	84
4.2.3	Attention heads	87
4.2.4	Generativity	89
4.3	Conclusion and Perspectives	93
5	Applications and Extensions of Attention-Based Direct Coupling Analysis	97
5.1	Decoupling and Flexibility in AttentionDCA	97
5.1.1	Subdomain Decomposition and Controlled Decoupling	99
5.2	AttentionDCA in CLAPP: Class Pooling for Paralog Matching	101
5.2.1	Paralog Matching and Coevolutionary Inference	101
5.2.2	Class Pooling in CLAPP	102

5.2.3	AttentionDCA as an Interaction Score	104
5.3	Conclusion	106
6	FeatureDCA: Controllable Generative Modeling of Protein Families	108
6.1	Introduction and Motivation	108
6.2	Model	111
6.2.1	Autoregressive modeling and Gaussian approximation . . .	111
6.2.2	Discrete ansatz	114
6.2.3	Sampling procedure	116
6.3	Evaluation tasks	116
6.3.1	Reproducing global statistics	117
6.3.2	Conditional generation in feature space	118
6.3.3	Mutational effect prediction	119
6.4	Results	119
6.4.1	Global generativity	120
6.4.2	In-place generativity	121
6.4.3	The case of the Response Regulator family (PF00072) . . .	123
6.4.4	In-silico Deep Mutational Scanning	129
6.5	Conclusion and Perspectives	129
7	Conclusion and outlook	132
	References	137
	Appendix A Statistical Inference Models	146
A.1	Derivation of the Mean-Field Approximation	146
A.2	Derivation of the Gaussian Approximation	149
A.3	Gauge-Fixing Regularization in PlmDCA	151

A.4	Pseudo-likelihood versus Autoregressive Factorizations in Potts Models	153
A.4.1	Pseudo-likelihood factorization	153
A.4.2	Autoregressive factorization	154
Appendix B	AttentionDCA: Supplementary Information	156
B.1	Contact scores	156
B.1.1	Attention and Frobenius Scores	156
B.1.2	Epistatic Score	159
B.2	Attention matrix sparsity analysis	160
B.3	Implementation details	163
B.3.1	Convergence Noise	163
B.3.2	Learning Parameters	163
B.4	Figures	165
Appendix C	FeatureDCA: Supplementary Information	172
C.1	Mathematical foundation of the feature-conditioned autoregressive model	172
C.1.1	Gaussian Approximation	173
C.1.2	Discrete model	183
C.2	Implementation Details	185
C.3	Data Processing and MSA Construction	186
C.4	Wasserstein distance and Sinkhorn divergence	187
C.5	Principal components beyond the second	189
C.6	In-silico Deep Mutational Scanning	190
C.7	Supplementary Figures	191

List of Figures

1.1	The twenty canonical amino acids grouped according to the chemical properties of their side chains.	3
1.2	Hierarchical organization of protein structure.	5
2.1	Schematic representation of the architecture of a profile Hidden Markov Model (HMM)	18
3.1	Encoder-decoder architecture for sequence-to-sequence modeling. .	51
3.2	Additive attention in encoder–decoder models	54
3.3	Schematic overview of the Transformer architecture.	59
3.4	Schematic overview of the AlphaFold 2 architecture	67
3.5	The Evoformer block within the AlphaFold architecture	69
4.1	AttentionDCA: Positive Predictive Value for different values of the compression rate	85
4.2	AttentionDCA: Positive Predicted Value Curves for families PF00014, PF00072, PF00763 respectively	87
4.3	AttentionDCA: Various contact maps for families PF00014, PF00072, PF00763.	89
4.4	AttentionDCA: Results from Autoregressive AttentionDCA applied to families PF00014, PF00072 and PF00763	91
4.5	AttentionDCA: Principal Component Analysis on protein families PF00014, PF00763 and PF13354	92

4.6	AttentionDCA: Positive Predictive Values for different effective depths	93
5.1	Contact prediction for interacting domains in the histidine kinase-response regulator (HK-RR) system.	100
5.2	Paralog matching performance for different protein systems using AttentionDCA interaction scores, with and without class pooling. Results are shown for (a) histidine kinase-response regulator (HK-RR), (b) bacterial Hsp70-J-domain pairs, and (c) the ModBC (2ONK) complex. In all cases, matching is performed either within species or using the CLAPP pooling framework. Ground truth interactions are inferred from genomic proximity, and clustering is performed using k-means to define class labels. Performance is compared to the iterative pairing algorithm (IPA). Adapted from [96].	107
6.1	FeatureDCA: Panel with global statistics	122
6.2	FeatureDCA: In-place generativity for PF13354	123
6.3	FeatureDCA: Structural validation of in-place generativity	124
6.4	FeatureDCA: Representative homodimer structures of Response Regulator (RR) subclasses with distinct dimerization modes.	125
6.5	FeatureDCA: Principal component projection of RR sequences reveals structurally distinct clusters.	127
6.6	FeatureDCA: Structural class specificity of FeatureDCA-generated sequences in the RR family (PF00072).	128
B.1	SM AttentionDCA: Overlap matrix	158
B.2	SM AttentionDCA: Standard AttentionDCA and AttentionDCA with frozen Value matrices for protein families PF00014 and PF00763	158
B.3	SM AttentionDCA: Epistatic Score PPV@L for AttentionDCA using different initial wild-type sequences for the computation of the Epistatic Score	160
B.4	SM AttentionDCA: PPV curves of different protein families	161

B.5	SM AttentionDCA: PPV curves @L and @2L for the averaged and maximized sparse Attention Score as the value of k varies	162
B.6	SM AttentionDCA: Frequency of the top $2L$ predicted contacts across 20 independent inference runs for nine protein families . . .	164
B.7	SM AttentionDCA: Positive Predicted Value Curves for different families	166
B.8	SM AttentionDCA: Contact map built from the Frobenius Score applied to the interaction tensor inferred by AttentionDCA	167
B.9	SM AttentionDCA: Grayscale heat map of the averaged attention matrix	168
B.10	SM AttentionDCA: Grayscale heat map for the sparse attention matrix	168
B.11	SM AttentionDCA: Contact prediction results from Autoregressive AttentionDCA applied to different families	169
B.12	SM AttentionDCA: Generativity results from Autoregressive AttentionDCA applied to different families	170
B.13	SM AttentionDCA: Principal Component Analysis on different protein families.	171
C.1	SM FeatureDCA: Principal component projections of natural and generated sequences for PF13354.	193
C.2	SM FeatureDCA: Predictive performance of FeatureDCA and arDCA on experimental deep mutational scanning data for beta-lactamase (PF13354).	194
C.3	SM FeatureDCA: Comparison of statistical properties across protein families as a function of PCA feature dimensionality.	195
C.4	SM FeatureDCA: Wasserstein distance between the PCA-space distributions of natural and generated MSAs for different models. . . .	196
C.5	SM FeatureDCA: Projection onto the first two principal components of natural and generated MSAs for different protein families.	197
C.6	SM FeatureDCA: Conditional generativity across protein families. .	198

C.7 SM FeatureDCA: Structural similarity between generated and wild-type sequences across different protein families.	199
---	-----

List of Tables

3.1	Transformer encoder overview	61
3.2	Transformer decoder overview	62
4.1	Summary of the protein families used throughout the analysis and main results	96
B.1	SM AttentionDCA: Inference parameters and hyper-parameters of the model for each protein family	164

Chapter 1

Introduction

In this chapter, we outline the biological background of this thesis by introducing the fundamental principles of protein structure and function characterization, the challenges of predicting and designing protein structures, and the main computational frameworks developed to address them. We emphasize the complementary roles of statistical physics and deep learning, highlighting both their strengths and limitations. We conclude by presenting the research objectives that motivate the methods developed in the following chapters.

1.1 The biology of proteins

The starting point of any study in the rich and complex domain of statistical physics, inferential learning and computational biology applied to proteins is to recognize the theoretical and practical importance of studying their structure and function. Proteins are among the most fundamental biomolecules in living organisms. They participate in virtually every biological process within the cell, fulfilling a remarkably broad spectrum of functions. As enzymes, they catalyze chemical reactions with extraordinary efficiency, sustaining metabolism and enabling life under physiological conditions. Other proteins play central roles in signaling pathways, where they act as receptors, messengers, or regulators, ensuring the transmission and processing of information within and between cells. Structural proteins, such as those forming the cytoskeleton or extracellular matrix, provide stability and organization to cells and tissues. Motor proteins convert chemical energy into mechanical work, driving

processes such as intracellular transport or muscle contraction. In the immune system, antibodies and other protein complexes mediate recognition and defense against pathogens. The universality and diversity of these roles underline the central position of proteins in biology and biotechnology. Understanding the principles that determine their structure and function is therefore of paramount scientific and technological importance [1, 2].

1.1.1 Chemical composition

From a biochemical point of view, proteins are polymers built from basic building blocks called amino acids. Each amino acid shares a common chemical scaffold, consisting of an amino group ($-\text{NH}_2$), a carboxyl group ($-\text{COOH}$), and a central carbon atom (the α -carbon) to which a variable side chain, or *R*-group, is attached. The *R*-group determines the chemical identity and physicochemical properties of the amino acid, such as polarity, hydrophobicity, or electric charge. Although many amino acids exist in nature, only twenty of them are encoded by the genetic code and synthesized within living genomes. Rarely, non-canonical residues such as selenocysteine or pyrrolysine can also be incorporated in specific organisms [1, 2].

Proteins are synthesized in cells at the level of the ribosome, where messenger RNA (mRNA), containing the transcription of the relevant gene, is translated into a peptide chain, according to the central dogma of molecular biology [3]. In the ribosome, amino acids are linked sequentially through condensation reactions that form peptide bonds. A peptide bond is a covalent bond between the carbon of the carboxyl group of one amino acid and the nitrogen of the amino group of the next one, linked together during a condensation reaction that releases a molecule of water as a byproduct. This process results in a polypeptide chain composed of a repeating backbone, from which the diverse *R*-group side chains project outward. The linear sequence of amino acids defines the identity of the protein and largely determines its structural and functional properties [1].

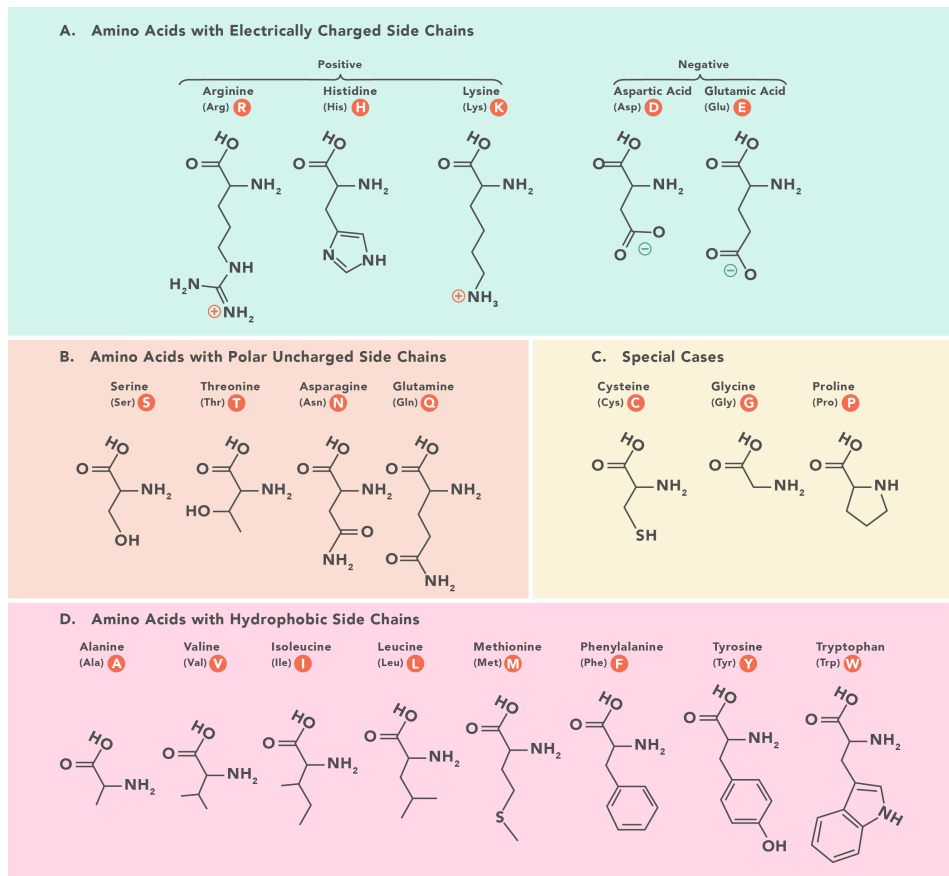


Fig. 1.1 The twenty canonical amino acids grouped according to the chemical properties of their side chains. Amino acids can be classified into four main categories: (A) amino acids with electrically charged side chains, which can be positively or negatively charged; (B) amino acids with polar but uncharged side chains; (C) special cases with unique structural or chemical features such as cysteine, glycine, and proline; and (D) amino acids with hydrophobic side chains. This chemical diversity underlies the wide range of structural and functional roles that proteins can adopt in biological systems. Source [4].

1.1.2 Levels of protein structure

The structural organization of proteins is commonly described across four hierarchical levels:

- **Primary structure:** the linear sequence of amino acids, directly encoded in the genome by a specific sequence of DNA called gene.

- **Secondary structure:** local conformations stabilized primarily by hydrogen bonds between backbone atoms. The most common motifs are α -helices and β -sheets, complemented by loops and turns [5–7].
- **Tertiary structure:** the overall three-dimensional arrangement of a single polypeptide chain, resulting from long-range interactions between residues. Hydrophobic packing, ionic bonds, hydrogen bonds, van der Waals interactions, and disulfide bridges all contribute to stabilizing the native fold and minimizing the Gibbs free energy of the entire molecule, [8].
- **Quaternary structure:** the assembly of multiple polypeptide chains, or subunits, into larger macromolecular complexes (e.g., hemoglobin or DNA polymerase).

Compared to other biomolecules such as DNA, RNA, or lipids, which generally adopt a limited set of stable conformations, proteins exhibit an extraordinary variety of three-dimensional structures, reflecting the wide range of possible biochemical functions.

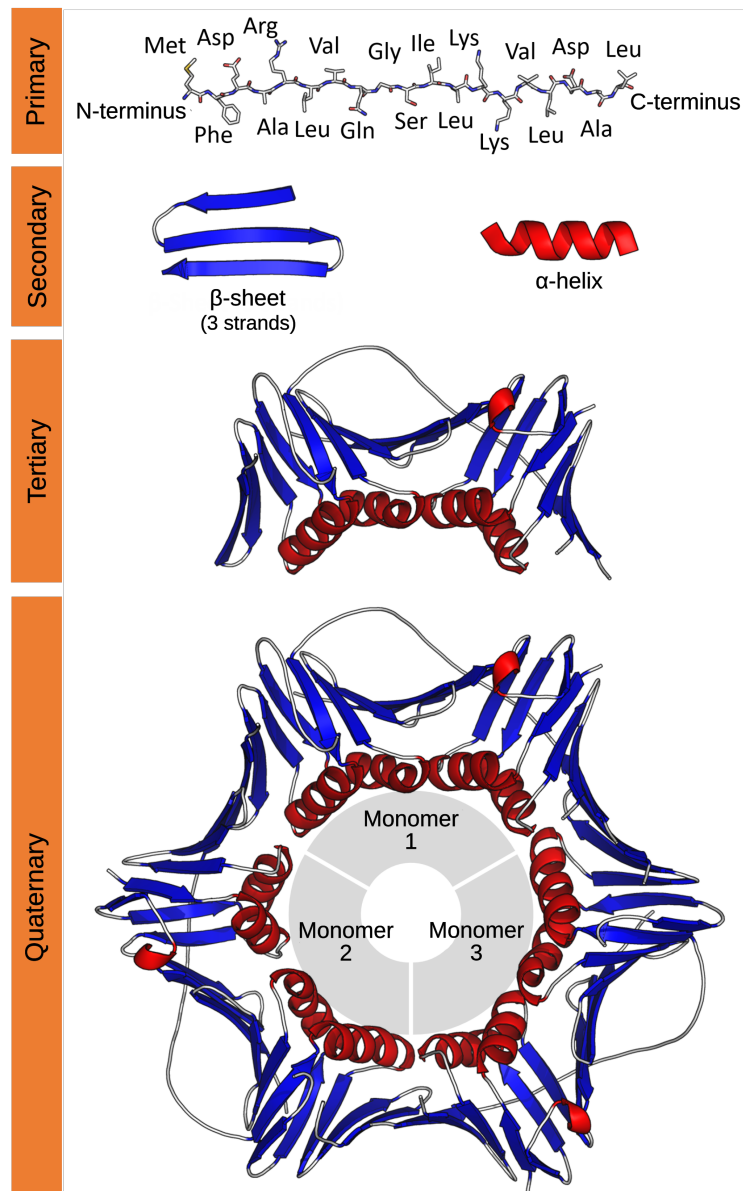


Fig. 1.2 Hierarchical organization of protein structure. Proteins are organized across four structural levels. The primary structure corresponds to the linear amino acid sequence that defines the polypeptide chain. The secondary structure arises from local hydrogen-bonding patterns forming motifs such as α -helices (red) and β -sheets (blue). The tertiary structure represents the overall three-dimensional conformation of a single polypeptide, determined by interactions between distant residues. Finally, the quaternary structure results from the association of multiple polypeptide subunits into a functional complex. Together, these hierarchical levels determine the physicochemical properties and biological roles of the protein. [Source](#)

1.1.3 Structure-function relationship

The function of a protein is intimately linked to its structure at every level of organization. Enzymes, for instance, depend on the precise spatial arrangement of residues in the catalytic site to bind substrates and lower activation energies. Antibodies recognize antigens with high specificity through variable loop regions, while structural proteins rely on large-scale assemblies to achieve mechanical stability. Even small conformational changes can drastically alter functionality, highlighting the delicate connection between form and activity [2]. In more extreme cases, protein misfolding or aggregation can result in a complete loss of function and is associated with numerous diseases. Prominent examples include Alzheimer's disease, where the aggregation of amyloid- β peptides into extracellular plaques and tau protein into intracellular neurofibrillary tangles are pathological hallmarks, and Parkinson's disease, which is linked to the accumulation of misfolded α -synuclein into Lewy bodies [9–11].

Connected to the structure-to-function paradigm, a particularly important concept is that of protein domains. Domains are modular subunits within a polypeptide chain, often carrying distinct structural or functional roles. Many proteins can be regarded as collections of domains, recombined and repurposed throughout evolution to yield new functionalities. This modularity facilitates evolutionary innovation and provides a natural unit for structural and functional analysis. As will be discussed in later chapters, domains are critical for defining protein families and for constructing multiple sequence alignments (MSAs), which are the foundation of statistical inference methods for protein sequence analysis.

1.2 The protein folding and protein design problems

In this section, we present the two central problems of protein modeling: predicting how sequences fold into native structures, and designing sequences that fold into specific structures with determined functions. These two challenges represent opposite directions of the same fundamental sequence-to-structure mapping and will be the fundamental motivation at the core of this thesis.

1.2.1 The protein folding problem

As described in the previous section, the intimate link between structure and function makes the determination of protein structures a problem of fundamental importance. Knowing the three-dimensional conformation of a protein often allows one to infer its biochemical role within the cell, to understand the molecular mechanism at the basis of a pathology, or to design targeted therapeutics. However, experimental methods for structure determination, such as X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and cryo-electron microscopy (cryo-EM), are extremely costly and time-consuming. Additionally, some classes of proteins present particular challenges during the purification and crystallization procedures required for structure determination. A notable example of this is that of membrane glycoproteins, whose amphipathic nature creates a strong dependence on the lipid environment for proper folding and stability [12, 13]. As a consequence, the number of experimentally solved protein structures remains orders of magnitude smaller than the number of known protein sequences, which continue to accumulate at an ever-increasing pace thanks to high-throughput sequencing technologies. A complete understanding of how sequence determines structure would therefore bridge this gap, allowing us to predict structure directly from sequence data and thus to study the functional information encoded in genomes.

The protein folding problem is classically formulated as the question of how a linear sequence of amino acids spontaneously acquires its native three-dimensional conformation [14]. From a thermodynamic point of view, the folded state corresponds to a free energy minimum in the conformational landscape of the polypeptide chain. Yet the number of possible conformations is astronomically large, and a purely random search would make folding essentially impossible within any biological time frame. This apparent paradox, first articulated by Cyrus Levinthal [15], emphasized that proteins must fold along specific pathways or be guided by energetic funnels in order to reach their native state efficiently.

Before the advent of data-driven approaches, biophysicists tried to understand folding using theoretical and physical models. Simplified lattice models and statistical mechanical descriptions treated folding as a phase transition between unfolded and folded states. Other approaches focused on diffusion and kinetics models, describing how polypeptides escape the unfolded state along a rugged energy landscape. While these models provided valuable insights into the thermodynamics and kinetics

of folding, they did not deliver experimentally-accurate predictive power. As a result, the protein folding problem remained one of the central open questions in molecular biology for decades [16, 17].

A major breakthrough arrived only recently with the development of deep learning methods [18]. In particular, the AlphaFold 2 [19] architecture demonstrated unprecedented accuracy in predicting protein structures from sequence data, reaching a level of performance that, in practice, solves much of the predictive aspect of the folding problem. However, while AlphaFold 2 provides highly accurate structural predictions, it does not explain the underlying folding mechanism, nor does it provide a transparent connection between sequence, energetics, and the folding pathways. Thus, from a practical perspective, it can be argued that the folding problem is close to being solved, in particular for the case of monomeric proteins with sufficient evolutionary information. However, from a mechanistic and interpretability standpoint, it remains an open challenge.

1.2.2 The protein design problem

Complementary to the protein folding problem is the protein design problem, which seeks to generate amino acid sequences that reliably fold into a desired three-dimensional structure. This task, often referred to as the inverse folding problem, is motivated by the fact that protein function is determined by structure, and therefore controlling structure allows one to engineer artificial proteins with chosen biological activities. Applications of protein design span across diverse fields: in medicine, to develop novel therapeutic antibodies or enzymes; in biotechnology, to create highly specific biocatalysts; and in synthetic biology and materials science, to engineer new molecular scaffolds or functional biomaterials.

Concretely, the challenge of protein design lies in the vastness of sequence space: for a protein of length L , there are 20^L possible amino acid combinations, only a minute fraction of which are compatible with a stable fold and a functional role. Exploring this astronomical space systematically is unfeasible without principled strategies, and thus protein design requires methods that can capture the essential constraints relating sequence, structure, and stability. Over the years, various computational and experimental approaches have been developed to guide design, but

achieving reliable and general solutions remains one of the grand challenges in molecular biology.

1.3 Statistical Physics and Deep Learning Approaches

The significant advances in protein modeling achieved over the past decades are rooted in two complementary yet deeply intertwined frameworks. On the one hand, statistical physics has offered principled and interpretable models to study the coevolutionary signals that emerge among related proteins. On the other hand, deep learning approaches have provided high expressivity and predictive power, though often at the expense of architectural transparency. In this section, we explore these two frameworks to motivate the need for hybrid strategies capable of merging their respective strengths.

1.3.1 Statistical physics perspective

The first strategy that we review originates from statistical physics. As many-body physical systems, proteins can be described by simple probabilistic models that balance descriptive power with interpretability. Instead of studying the microscopic details in an analytical way, which is computationally unfeasible, the goal is to construct effective models that capture the essential constraints observed in data [20].

In this sense, the mechanism of natural selection comes to our aid: evolution can be treated as a natural experiment that samples the space of possible protein sequences under strong structural and functional constraints [21–23]. Homologous proteins across different species accumulate mutations over time, and the resulting sequence variation contains signatures of both conservation at functionally critical sites and coordinated changes between residues that interact. Statistical modeling provides a principled way to extract these signals and encode them into the parameters of a probabilistic model. In a Bayesian approach, one starts with limited empirical information, such as amino-acid frequencies at single positions or correlations between positions, and constructs the simplest model consistent with these observations. The maximum entropy principle formalizes this idea, ensuring that no unwarranted assumptions are introduced beyond what the data dictate. The resulting models are typically energy-based: they assign a statistical energy to each sequence,

from which a probability distribution is derived. A common analogy can be made with spin systems in physics, where local fields describe the bias of each site toward a specific state (for instance, spin up or down) and coupling parameters describe interactions between sites.

Although these models are simplified, their parameters are interpretable: conservation translates into strong local fields, while couplings often correspond to structural contacts or functional dependencies. Beyond interpretability, statistical-physics-inspired models can be made generative: having defined an explicit probability distribution over sequence space, this can be sampled to produce new sequences consistent with evolutionary constraints. They can also be used predictively, for example by estimating the effect of mutations from changes in the energy of a sequence. Remarkably, these capabilities arise naturally from the probabilistic framework without the need for ad hoc assumptions [24–26].

At the same time, statistical models face limitations. They depend on the availability and diversity of homologous sequences, are often constrained to pairwise approximations, and can be affected by sampling and phylogenetic biases in the data. Nevertheless, their efficiency and interpretability explain their enduring influence over the past decade on more complex and less interpretable deep learning approaches.

In Chapter 2, we will formalize this perspective by introducing the concepts of multiple sequence alignments and the family of maximum entropy approaches used to extract evolutionary information from them to model the structure and function of a protein family, also known as Direct Coupling Analysis.

1.3.2 Deep learning perspective

In contrast to the statistical-physics view, which emphasizes simple probabilistic models with interpretable parameters, deep learning approaches pursue a different strategy. The aim is to learn complex, high-dimensional mappings directly from data, with minimal reliance on hand-crafted constraints or biological principles. Neural networks are used as flexible high-dimensional function approximators that can extract patterns in protein sequences and connect them to their structural or functional properties.

The first successes of deep learning in this domain were linked to the protein folding problem, though not by predicting full three-dimensional structures directly. Instead, researchers focused on predicting intermediate representations such as contacts or distance maps between residues, which could then be used as restraints in folding pipelines. Convolutional neural networks (CNNs) were among the earliest tools: for example, DeepCov [27, 28] applied CNNs directly to covariance matrices derived from multiple sequence alignments (MSAs) to predict contacts. Soon after, deeper architectures based on residual connections achieved major improvements. RaptorX-Contact [29] used deep residual networks to predict inter-residue contacts with much higher accuracy, while TripletRes [30] combined multiple coevolutionary features through a ResNet [31] architecture to predict not just binary contacts but full distance and orientation information. These predictions were then passed to energy minimization or gradient-based engines to reconstruct three-dimensional folds. A key milestone came with AlphaFold 1 [18], which competed in the 2018 Critical Assessment of protein Structure Prediction (CASP). It employed deep residual networks to predict inter-residue distance and torsion-angle distributions from MSAs and then optimized structures consistent with those distributions. This was the first clear demonstration that deep learning could significantly outperform traditional physics-based methods in structure prediction.

Parallel to these folding-oriented methods, another family of models addressed the protein design problem, i.e. the generation of sequences consistent with functional and structural constraints. Among the most influential ones there was DeepSequence [32], a variational autoencoder (VAE) trained on MSAs that learned a low-dimensional latent representation of protein families. This model was able to capture the effects of mutations and to sample novel sequences that remained consistent with evolutionary constraints. Energy-based approaches [33] were also explored, such as restricted Boltzmann machines (RBMs) [34], which provided a neural-network formulation for modeling sequence variability and higher-order dependencies beyond pairwise couplings. While these generative models were less accurate at predicting structure than CNN/ResNet-based approaches, they were important for demonstrating that deep architectures could capture nontrivial constraints in sequence space and could be used to propose new sequences for design.

As protein sequence databases continued to grow, a new generation of models sought to move beyond handcrafted features and alignment-derived statistics, learning directly from raw sequences at scale. Early examples included UniRep [35], an

LSTM-based [36] model trained on millions of sequences to produce embeddings useful for downstream tasks, and the TAPE benchmark suite [37], which standardized the evaluation of different architectures. These developments laid the groundwork for even larger architectures that would treat protein sequences analogously to natural languages. From 2019 onward, with the introduction of the attention mechanism and its integration into the transformer architecture, models such as ProtBERT, ESM, and the MSA Transformer [38, 39] demonstrated that large-scale pretraining on massive protein databases could learn structural and functional constraints directly from sequence context. The attention mechanism [40, 41], by allowing the model to dynamically weight relationships between all positions in a sequence, proved especially well suited to proteins, where long-range dependencies are essential for the correct three-dimensional folding of a sequence. This trajectory culminated in AlphaFold 2 [19] (CASP14, 2020), which introduced a transformer-based architecture, the Evoformer block, integrated with a structure refinement module, achieving unprecedented accuracy in end-to-end structure prediction and effectively solving the protein folding problem, from a predictivity perspective.

In summary, deep learning methods for proteins have developed along two complementary tracks: supervised predictors for folding that evolved from CNNs to ResNets and, eventually, to transformers, and generative models for design, such as VAEs and RBMs. Both tracks have converged in recent years toward large-scale pretrained models that blur the line between folding and design applications. These methods offer unparalleled expressive power and predictive accuracy, but at the cost of interpretability, data hunger, and computational resources. In Chapter 3, we will return to the transformer implementation of the attention mechanism and examine in detail its connections with statistical physics models that underlie modern advances in protein structure prediction and design.

1.4 Research objectives and aims

In Section 1.2 we have outlined one of the grand challenges in computational protein biology: understanding the sequences-to-structure paradigm in its direct and inverse implications, the protein folding and protein design problems. Then, in Section 1.3, we described the two methodological perspectives under which it is possible to tackle both problems, along with their strengths and shortcomings. Models inspired

by statistical physics, based on evolutionary data, provide interpretable parameters and generative capabilities but are limited in their expressiveness. Deep learning approaches, in contrast, achieve remarkable predictive accuracy by learning complex dependencies directly from data, but typically remain opaque and difficult to control. The tension between these two perspectives defines a remarkable gap in the field of computational biology. An ideal approach would combine the interpretability and controllability of statistical physics with the expressive power of modern machine learning. Yet, the state of the art suggests that such a hybrid model is far from our grasp. Aware of the complexity of such a quest, this thesis is but a humble contribution in a larger ocean of similar works aimed at laying some of the conceptual and methodological groundwork towards bridging the worlds of statistical inference and deep learning applied to protein analysis.

In this spirit, the objectives of the thesis are twofold. First, to explore whether architectural elements from deep learning, and in particular the attention mechanism that has proved groundbreaking in the last few years, can be integrated into statistical models in a way that provides insight into what such mechanisms actually learn when applied to protein sequence data. This motivates the development of AttentionDCA [42], an attention-based reformulation of classical statistical models, whose goal is not improved predictive accuracy but improved interpretability. Second, to advance generative modeling of protein sequences by extending statistical approaches with richer feature representations, leading to FeatureDCA [43]. Unlike AttentionDCA, this framework does yield performance improvements over standard generative DCA models, while also providing fine-grained and interpretable control over *in silico* protein generation.

Together, these two contributions exemplify different ways of addressing the gap between statistical physics and deep learning: one by probing the interpretability of deep learning mechanisms within a physics-inspired framework, the other by enhancing the generative capabilities of statistical models using ideas borrowed from machine learning. Neither approach provides a full solution, but both highlight directions in which interpretability, controllability, and accuracy can be pursued concurrently.

The remainder of this thesis develops the necessary background information and then presents these contributions. Chapter 2 introduces the statistical-physics framework for sequence-based inference. Chapter 3 presents the attention mechanism

and transformer architectures in the context of protein modeling. The following chapters describe in detail the AttentionDCA and FeatureDCA frameworks, and evaluate their implications for protein structure prediction and design.

Chapter 2

Statistical Inference Models for Protein Sequences

In this chapter, we review the state of the art in statistical methods developed over the past two decades for protein sequence modeling. We begin from the biological motivations rooted in evolutionary dynamics and the principles of statistical inference that led to the development of Direct Coupling Analysis, and proceed to its applications in computational biology tasks such as structure prediction, mutational landscape analysis, and protein design.

2.1 Evolutionary Constraints

2.1.1 Conservation and Co-evolution

Proteins, as biological macromolecules encoded in the genetic material of an organism, evolve under the constant action of natural selection [3]. Across evolutionary timescales, mutations accumulate in their amino acid sequences, but only those changes that preserve structural stability and functional activity are likely to survive. This interplay between variability and constraint leaves recognizable signatures in the ensemble of proteins observed to this day [44–46].

Two proteins are said to be homologous if they derive from a common ancestor and, as a result, exhibit a similar three-dimensional structure and function. Homology,

therefore, implies evolutionary relatedness, typically reflected in a conserved fold and often in shared biochemical roles. Collections of homologous proteins can be grouped into protein families, the natural units on which comparative and statistical analyses are based. A key concept in this classification is that of the protein domain. As defined in Section 1.1.3, domains are compact, structurally independent subunits of proteins that often fold autonomously and carry out specific biochemical functions. They can be thought of as the evolutionary building blocks of proteins: stable modules that are conserved across species and reused in different combinations to generate the vast diversity of protein architectures found in nature. Because domains maintain their structure and function even when embedded in different protein contexts, they provide a reliable framework to define protein families. Indeed, curated databases such as Pfam [47] and InterPro [48] classify proteins primarily at the domain level, grouping together sequences that share the same domain architecture and hence a common evolutionary origin.

Within a protein family, natural selection leaves two complementary evolutionary signatures [45]. On the one hand, some positions are highly conserved: substitutions at these sites tend to disrupt folding or function and are thus eliminated from the evolutionary record. Conserved residues often correspond to catalytic sites, metal-binding motifs, or structural anchors essential for stability. On the other hand, some positions may display a certain degree of variability limited to structural constraints. In such cases, a mutation at one site can be tolerated only if it is compensated by a corresponding mutation at another site. This phenomenon, known as co-evolution, reflects the fact that residues are subject to shared structural and functional constraints: a mutation at one position alters the local compatibility conditions, requiring compensatory substitutions at interacting sites

In summary, the evolutionary history of a protein family is inscribed in its sequences as a balance between conservation, marking residues of essential importance, and co-evolution, highlighting sets of residues that interact or cooperate. Both aspects are central to the statistical inference models that will be developed in the following sections.

2.1.2 Multiple Sequence Alignments

The evolutionary constraints described above become accessible to statistical analysis only once homologous sequences are brought into a common frame of reference. This is the purpose of a multiple sequence alignment (MSA), the central structure for statistical inference of protein data. An MSA arranges the amino acid sequences belonging to a common protein family so that equivalent positions, corresponding to homologous residues, are aligned across all sequences. In this representation, each row corresponds to a protein sequence and each column to a specific position along the chain.

From a formal point of view, an MSA can be seen as a matrix of size $M \times L$, where M is the number of sequences in the family (often referred to as the depth of the alignment) and L is the length of the aligned sequences. Each entry of the matrix belongs to an alphabet of size $q = 21$, consisting of the 20 natural amino acids plus a special gap symbol. Gaps are introduced during the alignment procedure to accommodate insertions and deletions that have occurred throughout the evolutionary history of the family, so that homologous residues result aligned.

The construction of reliable MSAs is a central task in computational biology. Progressive alignment algorithms, which iteratively align pairs of sequences according to a guide tree, were among the earliest approaches and remain effective when only a few sequences are available [49]. However, for large protein families containing thousands of diverse sequences, these methods tend to accumulate errors and become inconsistent. To overcome these limitations, modern pipelines rely on profile Hidden Markov Models (HMMs) [50–52].

In a profile HMM, the alignment is modeled as a probabilistic process defined by a chain of hidden states over observable states representing the amino-acid chain. Each hidden state can be of three types: match states corresponding to a consensus column of the alignment; insertion states, which allow for extra residues between consensus positions; and deletion states, which correspond to skipped consensus columns. Transition probabilities define the rates of moving from one hidden state to another, while emission probabilities define, for each site, a probability distribution over the amino acid alphabet that quantifies the likelihood of observing a given residue when in that state. Only match and insertion states emit residues, as deletion states are silent and only account for modeling gap symbols. Although the emission

distributions are site-specific and conditionally independent, correlations between different sites are introduced indirectly through the chain of transitions, which defines the most probable path through the model for a given sequence. In practice, profile HMMs encode conservation at each position while allowing insertions and deletions to occur with empirical probabilities, producing alignments that faithfully capture the evolutionary variability of the family. The parameters of the profile Hidden Markov Model are inferred from a small, manually curated seed alignment containing the defining sequence features of the protein family. Once the parameters are fixed, the HMM can be used to align new sequences so as to build deeper MSAs, which provide the statistical foundation for subsequent analyses such as DCA or deep learning models. Other than that, HMMs can be used to determine whether a specific protein sequence belongs to a given family or, more generally, to find a matching family among the ones available for analysis. This approach underlies large-scale resources such as Pfam [47] and InterPro [48], which distribute both the trained profile HMMs and the MSAs obtained from them.

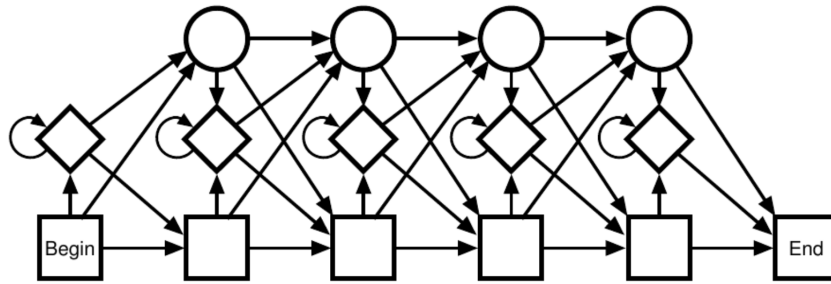


Fig. 2.1 Schematic representation of the architecture of a profile Hidden Markov Model (HMM). Direct graph topology known as “Plan7” architecture. Three kinds of hidden states are available: a match state (squares) that accounts for consensus positions, an insertion state (diamond) that accounts for extra residues between consensus columns, and a deletion state (circle) that represents skipped consensus positions. Arrows indicate the allowed transitions between states, including self-loops in insertion states for multiple residue insertions. The model starts at the *Begin* state and ends at the *End* state. Source [49]

Once an alignment is constructed, it can be treated as a collection of empirical data from which summary statistics can be extracted. The most fundamental observables are the single-site frequencies $f_i(a)$, measuring the probability of observing amino acid a at position i , and the pairwise frequencies $f_{ij}(a, b)$, measuring the joint

probability of observing residues a and b at positions i and j :

$$\begin{cases} f_i(a) = \frac{1}{M} \sum_{m=1}^M \delta(a, a_i^m) \\ f_{ij}(a, b) = \frac{1}{M} \sum_{m=1}^M \delta(a, a_i^m) \delta(b, a_j^m) \end{cases}, \quad (2.1)$$

where $\delta(a, a_i)$ is the Kronecker delta function. These quantities directly reflect conservation and co-evolution across the family. A simple measure such as the mutual information can already highlight positions that vary in a correlated way, although this signal is often blurred by indirect effects. In the following sections, we will see how statistical physics models can disentangle such effects to infer the direct interactions that underlie protein structure and function.

2.2 The Principle of Maximum Entropy

In this section, we introduce the Maximum Entropy Principle (MEP), one of the tools that lies at the foundation of Bayesian statistical inference and the core principle in Direct Coupling Analysis for biological sequence modeling.

In general, starting from an ensemble of empirical observations over a given dataset, the principle of maximum entropy provides a way to determine the most unbiased probability distribution that describes the given observations [20]. In particular, let us introduce a concrete example. Consider a discrete random variable x that takes values in $X = \{x_1, x_2, \dots, x_N\}$ with $N = |X|$, distributed according to the probability distribution $P(x)$, such that:

$$\begin{cases} P(x) \geq 0 & \forall x \in X \\ \sum_{x \in X} P(x) = 1 & . \end{cases} \quad (2.2)$$

The uncertainty associated with the distribution $P(x)$ is quantified by the Shannon entropy, defined as:

$$S(P) = - \sum_{x \in X} P(x) \log P(x). \quad (2.3)$$

This is a measure of the average unpredictability of the random variable: entropy vanishes when the outcome is certain, i.e. when all the probability mass is concentrated on a single state, and it is maximal when all outcomes are equally likely. In

this uniform case, where $P(x) = 1/|X|$ for all $x \in X$, one finds

$$S(P) = \log |X|. \quad (2.4)$$

Therefore, entropy interpolates between perfect knowledge of the event, a deterministic distribution with $S = 0$, and complete uncertainty, a uniform distribution with $S = \log |X|$.

Starting from this definition, the maximum entropy principle can be used to define probability distributions in situations where only partial knowledge is available. The idea is simple: given some constraints on the unknown distribution, we select the one with maximal entropy, i.e. maximum uncertainty, among the ones that fulfill the constraints. This way, the resulting model is consistent with the information at hand but does not introduce any further bias. For instance, suppose that the available information is represented by the expectation values of a set of observables $\{\phi_\mu(x)\}$, with $\mu = 1, \dots, K$. Then, we are interested in a probability distribution that satisfies

$$\sum_{x \in X} P(x) \phi_\mu(x) = \langle \phi_\mu \rangle \quad \forall \mu = 1, \dots, K, \quad (2.5)$$

together with the normalization condition

$$\sum_{x \in X} P(x) = 1. \quad (2.6)$$

Maximizing entropy under these constraints is a standard variational problem. To solve it, we introduce a functional

$$\mathcal{L}[P] = S(P) - \lambda \left(\sum_{y \in X} P(y) - 1 \right) - \sum_{\mu=1}^K \theta_\mu \left(\sum_{y \in X} P(y) \phi_\mu(y) - \langle \phi_\mu \rangle \right), \quad (2.7)$$

where λ and $\{\theta_\mu\}$ are Lagrange multipliers enforcing normalization and the expectation constraints. Taking the functional derivative with respect to $P(x)$ and imposing stationarity, we obtain

$$\frac{\delta \mathcal{L}}{\delta P(x)} = -\log P(x) - 1 - \lambda - \sum_{\mu=1}^K \theta_\mu \phi_\mu(x) = 0, \quad (2.8)$$

where we used the functional derivative identity $\delta P(y)/\delta P(x) = \delta(x, y)$. Rearranging terms, the distribution reads

$$P(x) = e^{-1-\lambda} e^{-\sum_{\mu=1}^K \theta_{\mu} \phi_{\mu}(x)} , \quad (2.9)$$

where the Lagrange multiplier enforcing the normalization condition λ can be determined by imposing

$$\sum_{x \in X} P(x) = e^{-1-\lambda} \sum_{x \in X} e^{-\sum_{\mu=1}^K \theta_{\mu} \phi_{\mu}(x)} = 1 . \quad (2.10)$$

This leads to the general solution

$$P(x) = \frac{e^{-\sum_{\mu=1}^K \theta_{\mu} \phi_{\mu}(x)}}{\sum_{y \in X} e^{-\sum_{\mu=1}^K \theta_{\mu} \phi_{\mu}(y)}} . \quad (2.11)$$

We can conclude that the maximum entropy principle yields a distribution that belongs to an exponential family, where the Lagrange multipliers $\{\theta_{\mu}\}$ enforce the constraints. This expression can be thought of as a Boltzmann distribution, where the observables $\{\phi_{\mu}(x)\}$ play the role of effective energy contributions and the multipliers $\{\theta_{\mu}\}$ act as generalized inverse temperatures controlling their relative weight.

In the following section we will see how this general result can be applied to the case of biological sequences such as proteins, where the observables correspond to empirical single-site and pairwise frequencies extracted from multiple sequence alignments.

2.3 The Potts Model for Protein Sequences

In the previous section we introduced the Maximum Entropy Principle (MEP) in a general setting. Now, let's apply it to the specific cause of protein sequence data.

Given a multiple sequence alignment, we can compute empirical single-site and pairwise frequencies that quantify conservation and co-evolution across a protein family. The goal is to construct a probability distribution, defined over the space of aligned sequences, that is consistent with these empirical statistics while introducing

no further bias. As shown in the following, the MEP construction leads to a generalized Potts model, which forms the basis of Direct Coupling Analysis (DCA), the ensemble of statistical inference methods for protein sequence modeling that will be one of the main characters of this thesis.

2.3.1 Maximum Entropy Construction

Let $\mathbf{a} = (a_1, \dots, a_L)$ denote an aligned sequence of L amino acids, where each a_i can take values among the 20 natural amino acids and the gap symbol used during alignment, for a total of $q = 21$ symbols. Using the definition in Equation 2.1, we can extract the empirical single-site and pairwise frequencies $f_i(a)$ and $f_{ij}(a, b)$ from the multiple sequence alignment. Thus, we can define a probability distribution $P(\mathbf{a})$ satisfying

$$P_i(a) = \sum_{\mathbf{a}} P(\mathbf{a}) \delta(a_i, a) = f_i(a) \quad \forall i, a, \quad (2.12)$$

$$P_{ij}(a, b) = \sum_{\mathbf{a}} P(\mathbf{a}) \delta(a_i, a) \delta(a_j, b) = f_{ij}(a, b) \quad \forall i < j, a, b, \quad (2.13)$$

together with the normalization condition

$$\sum_{\mathbf{a}} P(\mathbf{a}) = 1. \quad (2.14)$$

According to the Maximum Entropy Principle, the least biased distribution consistent with these constraints is obtained by maximizing the functional

$$\begin{aligned} \mathcal{L}[P] = & \sum_{\mathbf{s}} P(\mathbf{s}) \log P(\mathbf{s}) - \lambda \left(\sum_{\mathbf{s}} P(\mathbf{s}) - 1 \right) + \\ & - \sum_{i=1}^L h_i(s_i) \left(\sum_{\mathbf{s}} P(\mathbf{s}) \delta(s_i, a_i) - f_i(a_i) \right) + \\ & - \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j) \left(\sum_{\mathbf{s}} P(\mathbf{s}) \delta(s_i, a_i) \delta(s_j, a_j) - f_{ij}(a_i, a_j) \right), \end{aligned} \quad (2.15)$$

where the first term represents the Shannon entropy of the distribution and the constraint on the normalization condition, while the following two terms enforce the constraints on the single-site and pairwise frequencies. Following the general

derivation of Section 2.2, we find that the resulting distribution can be written as

$$P(\mathbf{a}) = \frac{1}{Z} \exp \left(\sum_{i=1}^L h_i(a_i) + \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j) \right) . \quad (2.16)$$

Here, $\{h_i(a)\}$ and $\{J_{ij}(a, b)\}$ are the Lagrange multipliers enforcing the empirical constraints, while Z is the normalization factor, defined as:

$$Z = \sum_{\mathbf{a}} \exp \left(\sum_{i=1}^L h_i(a_i) + \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j) \right) . \quad (2.17)$$

From a statistical physics point of view, this quantity is called the partition function and it can be used to determine all the relevant statistical properties of the system, such as the free energy or the expected value of any observable under the distribution $P(\mathbf{a})$. However, as it happens in most cases, its computation is unfeasible and approximation schemes are needed to determine it. Indeed, in the case of a protein family of length L , it would be necessary to sum over q^L possible sequences, a task that even for short sequences would be computationally impossible.

2.3.2 Statistical Physics Interpretation

The distribution in Equation 2.16 can be written in terms of an effective Hamiltonian, so that the probability distribution can be interpreted as a Boltzmann energy-based distribution:

$$H(\mathbf{a}) = - \sum_{i=1}^L h_i(a_i) - \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j) \quad (2.18)$$

$$P(\mathbf{a}) = \frac{1}{Z} e^{-H(\mathbf{a})} . \quad (2.19)$$

This construction highlights the statistical physics interpretation of the model. The parameters $h_i(a)$ play the role of effective local fields, quantifying the evolutionary pressure acting on site i . A higher local field denotes a higher level of conservation at a given position in the amino acid sequence. On the other hand, the couplings $J_{ij}(a, b)$ represent effective interactions between residues at positions i and j , showcasing co-evolution between the two. The partition function Z ensures normalization and encodes the contribution of all possible sequences. The resulting model is a q -state

Generalized Potts Model (GPM) (with $q = 21$), and it represents the minimal model consistent with the observed conservation and co-evolution in the protein family. It is worth noting that higher-order statistics, such as three-point frequency counts, could in principle be included in the Maximum Entropy construction to define a more detailed distribution. However, incorporating such constraints would then require a substantially larger amount of sequence data in order to infer the values of the corresponding parameters. In practice, the balance between data availability and inference reliability, together with the *a posteriori* observation that pairwise statistics are enough to capture the essential structural information in a protein family, has led to the choice of restricting the model to single-site and two-site frequencies, as defined in Equation 2.16

2.3.3 Gauge Freedom

Before moving on to the actual description of Direct Coupling Analysis, it should be emphasized that the parameters of the Generalized Potts Model (GPM), as defined in Equation 2.1, cannot be determined uniquely. Indeed, there is a fundamental redundancy in the choice of the parameters that reflects the fact that energy-based models in the form of Equation 2.19 are invariant under parameter transformations that shift the Hamiltonian by a constant amount. In particular, because such a probability distribution depends only on relative energies, global shift transformations leave $P(\mathbf{a})$ unchanged.

Alternatively to these energy-based considerations, the invariance of the choice of the parameters can be analyzed from an algebraic point of view by studying the number of constraints and parameters imposed by the MEP construction. In particular, the model defined in Equation 2.16 presents a total of

$$N_p = Lq + \frac{1}{2}L(L-1)q^2, \quad (2.20)$$

parameters in the form of local field $h_i(a)$ and couplings $J_{ij}(a,b)$. These are determined by imposing the constraints coming from the single-site and pairwise frequency counts. However, the number of independent constraints is lower than the number of parameters. In particular, only $L(q-1)$ of the one-point frequency counts are independent, since they must sum to one at each site. Similarly, for each pair (i,j) , only $(q-1)^2$ of the two-point frequencies are independent, since

marginalization over one index recovers the single-site frequency. The total number of independent constraints is therefore

$$N_c = L(q-1) + \frac{L(L-1)}{2}(q-1)^2 \quad . \quad (2.21)$$

Since $N_p > N_c$, the model is over-parameterized leading to a gauge freedom, namely, the existence of an infinite set of transformations of $\{h_i, J_{ij}\}$ that leave the distribution invariant. Explicitly, one possible realization of this gauge freedom is the transformation

$$\begin{cases} J_{ij}(a, b) \longrightarrow J_{ij}(a, b) + K_{ij}(a) + K_{ji}(b) \\ h_i(a) \longrightarrow h_i(a) + g_i - \sum_{j \neq i} (K_{ij}(a) + K_{ji}(a)) \end{cases} \quad , \quad (2.22)$$

where the functions $K_{ij}(a)$ and g_i are arbitrary. Different choices of these functions correspond to different gauges, i.e. different parameterizations of the same distribution. Common gauge choices include the lattice-gas gauge, in which

$$h_i(q) = 0, \quad J_{ij}(a, q) = J_{ij}(q, a) = 0 \quad \forall i, j, a,$$

so that all parameters are measured relative to a chosen reference state q , namely the gap state, and the zero-sum or (Ising) gauge, which enforces

$$\sum_a h_i(a) = 0, \quad \sum_a J_{ij}(a, b) = \sum_b J_{ij}(a, b) = 0 \quad \forall i, j, b.$$

Once the gauge is fixed, the number of effective parameters exactly matches the number of independent constraints N_c , ensuring that the inference problem is well-defined. The choice of gauge does not affect the model distribution $P(\mathbf{a})$, or any global, gauge-invariant observable. However, the numerical values of the fields and couplings depend on the chosen gauge. As it will be discussed in the following sections, some of the practical tasks of Direct Coupling Analysis, such as contact prediction, depend on the particular values of the couplings and the field. Therefore, it is common practice to fix a gauge on the inferred parameters *a posteriori*, so that different realizations of the same inference can be compared. Alternatively, it is possible to define gauge-invariant metrics such as Direct Information, in order to nullify the effect of the gauge on predictions.

2.4 Direct Coupling Analysis

The Potts model introduced in the previous section represents the minimal model capable of reproducing the empirical statistics defined as the frequency counts in Equation 2.1. In particular, given the model in Equation 2.16 and the observations in the form of a multiple sequence alignment, it is possible to infer the parameters $\{h_i(a), J_{ij}(a, b)\}$ such that

$$\langle \delta(a_i, a) \rangle_{P(\mathbf{a})} = f_i(a) \quad \forall i, a, \quad (2.23)$$

$$\langle \delta(a_i, a) \delta(a_j, b) \rangle_{P(\mathbf{a})} = f_{ij}(a, b) \quad \forall i < j, a, b. \quad (2.24)$$

From a statistical physics point of view, the task of inferring the parameters of a probabilistic model from direct observations of its realizations is known as an inverse problem. In particular, the possible inference strategies to the inverse Potts problem represent a class of techniques known as Direct Coupling Analysis (DCA), which, as already discussed throughout this Chapter and the previous one, represents the basis for statistical analysis of protein sequence data.

In this section, we explore the main strategies that have been implemented to solve Direct Coupling analysis since its first development in [22] to this day.

In principle, the inverse Potts problem can be solved by maximum likelihood inference. However, exact likelihood maximization requires evaluating the partition function of the model, a computation that scales exponentially with the length of the MSA. As a result, exact inference is computationally unfeasible for realistic protein families, and a variety of approximate methods have been developed to overcome this limitation. In the following, we first present the formal maximum likelihood framework and its computational bottleneck, and then describe the main approximations that underlie DCA.

2.4.1 Maximum Likelihood Inference

The first technique that we explore is the Maximum Likelihood Estimate (MLE), which, although exact in principle, proves to be computationally unfeasible due to the exponential complexity of computing the partition function of the model.

However, despite its impracticability in real scenarios, we present the case of MLE as it represents a pedagogical example of statistical inference.

Let $D = \{\mathbf{a}^1, \dots, \mathbf{a}^M\}$ denote a multiple sequence alignment of M sequences of length L , where each sequence $\mathbf{a}^m = (a_1^m, \dots, a_L^m)$ is composed of symbols from an alphabet of size $q = 21$. Assuming the Potts model distribution of Equation 2.16, the likelihood of the model on the dataset can be defined as

$$\mathcal{L}(\{h, J\} | D) = \prod_{m=1}^M P(\mathbf{a}^m) = \prod_{m=1}^M \frac{1}{Z} \exp \left(\sum_{i=1}^L h_i(a_i^m) + \sum_{i < j} J_{ij}(a_i^m, a_j^m) \right) . \quad (2.25)$$

For reasons of numerical stability, it is more convenient to work with the average log-likelihood,

$$\ell(\{h, J\}) = \frac{1}{M} \sum_{m=1}^M \log P(\mathbf{a}^m) , \quad (2.26)$$

which, in the case of a Potts model, becomes

$$\ell(\{h, J\}) = \sum_{i,a} f_i(a) h_i(a) + \sum_{i < j} \sum_{a,b} f_{ij}(a,b) J_{ij}(a,b) - \log Z . \quad (2.27)$$

Here, $f_i(a)$ and $f_{ij}(a,b)$ are the empirical one- and two-point frequency counts defined in Equation 2.1. The optimal parameters of the model are those that maximize the likelihood in Equation 2.27. In particular, from the stationary conditions

$$\frac{\partial \ell}{\partial h_i(a)} = f_i(a) - \langle \delta(a_i, a) \rangle_P = 0 , \quad (2.28)$$

$$\frac{\partial \ell}{\partial J_{ij}(a,b)} = f_{ij}(a,b) - \langle \delta(a_i, a) \delta(a_j, b) \rangle_P = 0 , \quad (2.29)$$

It can be shown that optimality is reached when the model reproduces the empirical frequency counts. The main obstacle in this reasoning arises in the evaluation of the model expectations $\langle \cdot \rangle_P$, which require knowledge of the full partition function

$$Z = \sum_{\mathbf{a}} \exp \left(\sum_{i=1}^L h_i(a_i) + \sum_{i < j} J_{ij}(a_i, a_j) \right) . \quad (2.30)$$

This sum extends over all possible sequences of length L , i.e. to q^L configurations. For typical proteins with $L \sim 100$ and $q = 21$, this corresponds to approximately 10^{130} terms, far beyond any computational capability available at the present time. As

a consequence, exact maximum likelihood inference of the Potts model parameters is practically impossible, and one must resort to approximations of different kinds. The level of approximation used to handle the intractable partition function and solve the inverse Potts problem defines other possible inference schemes for DCA. This will be the topic of the remainder of this Section.

2.4.2 Mean-Field Approximation

One of the first and most widely used approximation strategies in the early days of DCA is based on a mean-field treatment of the Potts model, leading to the so-called mean-field Direct Coupling Analysis (mfDCA) [21].

Within this approximation scheme, the central idea is to relate the couplings $J_{ij}(a, b)$ to the connected correlations observed in the data. In particular, let us define the empirical covariance matrix, which measures the correlation between observing amino acid a at site i and amino acid b at site j as

$$C_{ij}(a, b) = f_{ij}(a, b) - f_i(a)f_j(b) \quad , \quad (2.31)$$

Inspired by the Plefka expansion of the Ising model [53], the inverse of this covariance matrix provides an estimate of the direct couplings:

$$J_{ij}(a, b) \simeq -(C^{-1})_{ij}(a, b) \quad . \quad (2.32)$$

In practice, the covariance matrix computed from an MSA captures both direct and indirect correlations between residues. For instance, two positions i and j may appear correlated not only because they interact directly, but also because they both interact with a third site k . In mfDCA, one assumes a weakly interacting model in which correlations are small and approximately linear. This leads to approximation 2.32, which provides estimates of the coupling parameters that disentangle direct from indirect correlations: the inversion of the covariance matrix effectively suppresses transitive effects and isolates the direct interactions.

Despite its crude approximations, mfDCA was the first approach to demonstrate the power of direct coupling inference for contact prediction in protein families with length between 50 and 500 amino acids, far surpassing the earliest implementations of DCA with message passing algorithms [22]. However, the approximation comes

with significant limitations. By construction, it neglects higher-order corrections beyond the mean-field term, and its accuracy deteriorates in regimes of strong coupling or limited data. Moreover, the inversion of a noisy covariance matrix can lead to unstable estimates, making mfDCA particularly sensitive to the quality and depth of the alignment. As a consequence, while mfDCA is conceptually and historically important, it has been largely superseded by more accurate inference schemes such as Gaussian and pseudolikelihood methods, which will be discussed in the following sections.

2.4.3 Gaussian Approximation

An alternative route to approximate the inference is provided by the Gaussian Direct Coupling Analysis (GaussDCA), introduced in [54]. In this case, instead of treating the aligned sequences as realizations of a discrete Potts model, we relax their nature to continuous variable distributed as a multivariate Gaussian. The advantage of this approximation is that inference in Gaussian models is analytically tractable and direct couplings can be obtained in closed form as the inverse of the covariance matrix defining the Gaussian distribution.

Consider each sequence $\mathbf{a}^m$ in the alignment as a one-hot encoded binary vector of length Lq . Let $x_i^m(a)$ denote the indicator variable equal to one if residue a is present at site i in sequence m , and zero otherwise. The empirical covariance matrix of these indicator variables is defined as

$$\Sigma_{ij}(a, b) = \frac{1}{M} \sum_{m=1}^M (x_i^m(a) - f_i(a))(x_j^m(b) - f_j(b)), \quad (2.33)$$

where $f_i(a)$ are the empirical single-site frequencies. Within the Gaussian assumption, the inverse covariance matrix Σ^{-1} , also called the precision matrix, directly encodes the conditional dependencies between variables. In analogy with the Potts model, the couplings are estimated as

$$J_{ij}(a, b) \simeq -(\Sigma^{-1})_{ij}(a, b) \quad . \quad (2.34)$$

This approach has clear similarities with the mean-field approximation: both rely on matrix inversion to disentangle direct from indirect correlations. However, GaussDCA differs in that it abandons the discrete nature of the problem in favor of a

continuous Gaussian representation. This makes the method extremely fast and analytically transparent, but at the cost of accuracy, since the true distribution of protein sequences is far from Gaussian.

In practice, GaussDCA has been shown to improve on pure mean-field methods, but it is generally outperformed by more refined inference strategies such as pseudo-likelihood maximization. Its main value lies in its simplicity and efficiency, which make it a useful baseline and a conceptually appealing approximation within the family of DCA methods.

2.4.4 Boltzmann Machine Learning

Other than approximating the Potts Hamiltonian to simpler and tractable models, as in the cases of mfDCA and GaussDCA, a parallel approach is that of exploring different optimization techniques in place of the analytical scheme described in Section 2.4.1.

In particular, a conceptually straightforward way to solve the inverse Potts problem is to use stochastic optimization on the exact average log-likelihood defined in Equation 2.27. This approach, commonly referred to as Boltzmann Machine Direct Coupling Analysis (bmDCA) [24, 55], is directly inspired by Boltzmann Machines [33] in statistical physics and machine learning, i.e. energy-based models whose parameters are inferred by optimizing an objective function through some form of stochastic gradient ascent [56]. In the case of DCA, consider the gradient of the average log-likelihood derived in Section 2.4.1:

$$\begin{cases} \frac{\partial \ell}{\partial h_i(a)} &= f_i(a) - \langle \delta(a_i, a) \rangle_P \quad , \\ \frac{\partial \ell}{\partial J_{ij}(a,b)} &= f_{ij}(a,b) - \langle \delta(a_i, a) \delta(a_j, b) \rangle_P \quad . \end{cases} \quad (2.35)$$

Maximizing the log-likelihood corresponds to adjusting the parameters until the model expectations $\langle \cdot \rangle_P$ match the empirical frequencies $\{f_i, f_{ij}\}$. The objective function is concave in its parameters, so that the model admits a unique maximum.

The difficulty lies in the evaluation of the ensemble averages $\langle \cdot \rangle_P$, which require sampling from the Potts distribution. While direct computation is impossible due to the exponential complexity of the partition function, these expectations can be estimated by Markov Chain Monte Carlo (MCMC) simulations, which generate

approximate equilibrium samples from the model without explicitly evaluating Z , allowing for approximating actual ensemble averages with sample averages. This leads to a stochastic gradient ascent algorithm with parameter updates defined as

$$\begin{cases} h_i(a) & \leftarrow h_i(a) + \eta \left(f_i(a) - \langle \delta(a_i, a) \rangle_P \right) \\ J_{ij}(a, b) & \leftarrow J_{ij}(a, b) + \eta \left(f_{ij}(a, b) - \langle \delta(a_i, a) \delta(a_j, b) \rangle_P \right) \end{cases}, \quad (2.36)$$

where η is the learning rate controlling the magnitude of parameter updates at each iteration. This is typically chosen as a small positive constant to ensure stable convergence, with its optimal value depending on the size of the dataset and the variance of the Monte Carlo estimates; in practice, η is treated as a hyperparameter and can be tuned empirically, sometimes with a schedule that varies its value during training to refine convergence. In principle, iterating this update rule leads to maximum likelihood estimates of the Potts parameters. However, the method is computationally very demanding. At each iteration, long MCMC simulations are required to obtain reliable estimates of the ensemble averages. For protein families of realistic sizes, with hundreds of sites and thousands of sequences, convergence becomes prohibitively slow due to the Monte Carlo sampling at each update, which is computationally very costly in such high-dimensional systems. Despite these limitations, bmDCA plays a central role in the landscape of DCA inference methods. It provides the conceptually exact state-of-the-art solution to the inverse Potts problem, against which approximations such as mean-field, Gaussian, and pseudolikelihood methods can be compared.

A central advantage of bmDCA is that it defines a fully generative model of protein sequences. Once the parameters $\{h, J\}$ have been inferred, the Potts distribution assigns a probability to every possible sequence, and new sequences can be generated by Monte Carlo sampling. This way, one can sample *in silico* artificial sequences that are not explicitly present in the original alignment but are statistically consistent with it. This is due to the fact that the frequency constraints are fulfilled by construction: the stochastic gradient update rules explicitly drive the model averages towards the empirical frequencies, ensuring that the generated ensemble reproduces the conservation and co-evolutionary patterns observed in nature. This generative capability has been validated in practice, with bmDCA-sampled sequences shown to capture the essential statistics of natural protein families, and even to fold and

function in vivo [25]. The implications of this property for protein design will be further explored in the following chapters.

2.4.5 Pseudo-likelihood Maximisation

While Boltzmann Machine learning provides a conceptually exact solution to the inverse Potts problem, it is computationally impractical because ensemble averages must be estimated with costly Monte Carlo sampling. A compromise between efficiency and accuracy is represented by the maximum pseudo-likelihood estimate of DCA defined in [57] and known as PlmDCA. The idea behind PlmDCA is to replace the global joint probability distribution $P(\mathbf{a})$ with a product of univariate single-site probability distributions, thereby removing the dependence on the full partition function entirely. This approximation finds a natural justification in the statistical physics formalism through Callen's equations [58], which express the expected value of a spin variable in terms of its conditional probability given all the others. In the case of an Ising model, Callen's equation reads

$$\langle \sigma_i \rangle = \left\langle \tanh \left(\beta \sum_j J_{ij} \sigma_j + \beta h_i \right) \right\rangle . \quad (2.37)$$

In the case of a Potts model, for each site i , we can define the conditional probability of observing residue a_i given the configuration of the other sites $\mathbf{a}_{\setminus i}$ as

$$P(a_i | \mathbf{a}_{\setminus i}) = \frac{\exp \left(h_i(a_i) + \sum_{j \neq i} J_{ij}(a_i, a_j) \right)}{\sum_{b=1}^q \exp \left(h_i(b) + \sum_{j \neq i} J_{ij}(b, a_j) \right)} . \quad (2.38)$$

Unlike the full distribution $P(\mathbf{a})$, this expression involves only a local partition function, i.e. a sum over q states of site i , which can be evaluated exactly. The pseudo-likelihood of a dataset $D = \{\mathbf{a}^1, \dots, \mathbf{a}^M\}$ is then defined as

$$\mathcal{L}_{\text{PL}}(\{h, J\} | D) = \prod_{m=1}^M \prod_{i=1}^L P(a_i^m | \mathbf{a}_{\setminus i}^m) , \quad (2.39)$$

and the corresponding average log-pseudo-likelihood reads

$$\ell_{\text{PL}}(\{h, J\}) = \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^L \log P(a_i^m | \mathbf{a}_{\setminus i}^m) . \quad (2.40)$$

Although Callen's equations are exact identities, the full probability distribution cannot, in general, be exactly factorized into conditional distributions of single spins over all others. Therefore, the pseudo-likelihood defined in Equation 2.38 constitutes an approximation to the full likelihood of the model over the data. Nevertheless, this formulation provides a powerful and computationally efficient framework for inferring Potts model parameters while preserving the essential statistical dependencies among residues. In particular, a crucial feature of this objective function is that it factorizes over sites:

$$\ell_{\text{PL}} = \sum_{i=1}^L \ell_{\text{PL}}^{(i)} \quad ,$$

so that each $\ell_{\text{PL}}^{(i)}$ depends only on the parameters $\{h_i, J_{ij} : j \neq i\}$. This means that the optimization can be performed independently for each site, naturally allowing for parallelization of the learning problem. However, this site-wise optimization introduces an asymmetry in the model: the coupling $J_{ij}(a, b)$ estimated from $\ell_{\text{PL}}^{(i)}$ does not necessarily equal the value obtained from $\ell_{\text{PL}}^{(j)}$. In practice, symmetrisation schemes are applied to the inferred parameters, typically by averaging:

$$J_{ij}(a, b) \longleftarrow \frac{1}{2} (J_{ij}^{(i)}(a, b) + J_{ij}^{(j)}(a, b)),$$

so that the final parameters respect the symmetric nature of the Potts Hamiltonian. It is worth noting that, unlike bmDCA, PlmDCA does not enforce the empirical frequency constraints explicitly during learning. Nevertheless, it can be shown analytically that pseudo-likelihood maximization is a statistically consistent technique: in the large-data limit, the inferred parameters converge to the true ones, and the corresponding model reproduces the observed one- and two-point statistics, thereby ensuring compatibility with the maximum entropy construction [59].

In summary, PlmDCA achieves a balance between tractability and accuracy. It avoids the intractable partition function, admits efficient parallel optimization, and yields parameter estimates that are asymptotically consistent. In practice, PlmDCA has proven to be both fast and accurate, and remains the most widely used inference scheme in Direct Coupling Analysis. Among its applications, PlmDCA provides the best-performing predictions of residue-residue contacts, making it the de facto statistical learning (as opposed to deep learning) standard for structural inference from sequence data.

2.4.6 Autoregressive Inference

The main limitation of PlmDCA, when compared to bmDCA, is that it does not provide direct access to the exact full probability distribution in sequence space, or to quantities that depend on the global normalization partition function, such as single-sequence probabilities or entropy, which remain out of reach.

The autoregressive formulation of DCA (arDCA) presented in [26] addresses this specific limitation by introducing an exact single-site factorization of the full distribution. This particular factorization arises from the chain rule of probability:

$$P(\mathbf{a}) = \prod_{i=1}^L P(a_i | a_1, \dots, a_{i-1}) \quad . \quad (2.41)$$

This representation of the full distribution guarantees that the model is properly normalized and allows one to evaluate exact sequence likelihoods and related statistical quantities. In practice, the energy functions of the true conditionals would not be of a Potts-like form due to proliferation of interactions; however, arDCA approximates each of them with a Potts-like Hamiltonian of the form:

$$P(a_i | \mathbf{a}_{<i}) = \frac{\exp\left(h_i(a_i) + \sum_{j=1}^{i-1} J_{ij}(a_i, a_j)\right)}{\sum_{b=1}^q \exp\left(h_i(b) + \sum_{j=1}^{i-1} J_{ij}(b, a_j)\right)} \quad , \quad (2.42)$$

where $\mathbf{a}_{<i} = (a_1, a_2, \dots, a_{i-1})$. At this point, inference reduces to maximizing the average log-likelihood of the alignment under this autoregressive model. From a computational perspective, arDCA retains the tractability advantages of PlmDCA: inferring the model requires only local normalizations that can be performed in parallel and sampling is straightforward. In particular, sequences can be generated sequentially, one site at a time, by drawing the residue in each position from the corresponding conditional distribution given the previously generated residues. This makes arDCA efficient both for training and sampling.

It is important to emphasize that, compared to PlmDCA, the approximation in arDCA is of a different nature. On the one hand, pseudo-likelihood maximization approximates the joint probability by a product of single-site distributions conditioned on the full context $\mathbf{a}_{\setminus i}$, with the conditionals naturally taking a Potts-like form. On the other hand, arDCA factorizes the joint probability exactly but constrains each conditional to take a Potts parametrization, which naturally would be more complex.

The mathematical details on the differences between the two approximation schemes are given in Appendix Section A.4. A direct consequence of the autoregressive construction is that arDCA introduces an ordering of sites depending on the chosen factorization from Equation 2.42, thereby breaking permutation invariance. The couplings J_{ij} can no longer be interpreted as symmetric direct interactions, but only as directional coefficients that quantify how residues at earlier positions in the chosen ordering influence the conditional distribution of later sites. The lack of permutation invariance is the main theoretical drawback of arDCA. From a biological perspective, it is difficult to justify the existence of an intrinsically optimal autoregressive ordering, and computing a de facto optimal one is exponentially complex. Nevertheless, heuristic choices such as natural or entropic ordering, where the sequence order is determined by the empirical variability of sites, have been proposed. In practice, the performance of arDCA is relatively stable across different orderings, suggesting that this limitation is of minor practical importance even if it remains an open theoretical issue.

In summary, the autoregressive implementation of DCA provides a complementary approach to PlmDCA. It yields a fully normalized probability distribution, enables the computation of global statistical quantities, and allows for straightforward generative sampling. Its limitations lie in the dependence on sequence ordering and the approximate nature of the conditional parametrization, but in practice these effects appear to be small. At the same time, its generative performance has been found to be comparable to that of bmDCA, while requiring substantially lower computational cost, making arDCA a valuable extension of the DCA framework.

2.5 Regularization and Sequence Re-weighting

The inference of Potts models from multiple sequence alignments is challenged by two fundamental issues. First, the number of available sequences is typically small compared to the number of parameters of the model. For a protein family of length L and alphabet size q , the generalized Potts model has $\mathcal{O}(L^2 q^2)$ parameters, which usually exceeds by far the number of statistically independent sequences in the alignment. As a result, naive maximum likelihood estimation would overfit the sampling noise, leading to unstable parameter estimates and poor generalization. Second, the sequences in an alignment are not independent samples of evolutionary

trajectories. They are related by common ancestry and reflect strong phylogenetic correlations, as well as non-evolutionary biases introduced by sequencing efforts. Consequently, some organisms may be overrepresented, while others are scarcely sampled, distorting the empirical frequency counts.

In this section, we explore the two complementary strategies adopted in DCA to overcome these problems. There are sequence re-weighting, which reduces the impact of redundancy and phylogenetic bias, and regularization, which controls overfitting by penalizing large parameter values.

2.5.1 Sequence re-weighting

Due to bias induced at the level of sequencing biological data, MSAs often contain highly similar sequences from phylogenetically close organisms. If treated equally, these sequences would dominate the empirical statistics, inflating the weight of certain evolutionary branches. To mitigate this bias, a re-weighting scheme is applied: each sequence receives a weight inversely proportional to the number of similar sequences above a given identity threshold.

Formally, let $\text{Id}(\mathbf{a}^m, \mathbf{a}^n)$ denote the fraction of identical residues between two sequences $\mathbf{a}^m$ and $\mathbf{a}^n$, also known as the normalized Hamming distance. For a threshold $x \in [0, 1]$ (typically $x \simeq 0.8$), the weight of sequence m is defined as

$$w_m = |\{n \mid \text{Id}(\mathbf{a}^m, \mathbf{a}^n) \geq x\}|^{-1} . \quad (2.43)$$

The effective number of independent sequences, or effective depth of the MSA, is then

$$M_{\text{eff}} = \sum_{m=1}^M w_m . \quad (2.44)$$

This quantity provides a more realistic measure of the information content of the alignment. Weighted empirical frequencies are computed as

$$\begin{cases} f_i(a) = \frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m \delta(a, a_i^m) \\ f_{ij}(a, b) = \frac{1}{M_{\text{eff}}} \sum_{m=1}^M w_m \delta(a, a_i^m) \delta(b, a_j^m) \end{cases} . \quad (2.45)$$

By construction, redundant sequences contribute less to the statistics, while more diverse sequences are emphasized. This correction makes empirical frequencies

more representative of independently evolved sequences and reduces phylogenetic bias in subsequent inference. This re-weighting scheme plays an important role when defining log-likelihoods such as that defined in Equation 2.26, which becomes

$$\ell(\{h, J\}) = \frac{1}{M_{\text{eff}}} \sum_{m=1}^M w^m \log P(a^m) \quad (2.46)$$

2.5.2 Regularization

Even after re-weighting, the number of effective sequences is generally insufficient to constrain all parameters of the Potts model reliably. To avoid overfitting, it is therefore necessary to introduce a regularization, i.e. a bias towards small parameter values that stabilizes the inference. The most common choice in DCA is the L_2 -regularization, which penalizes large fields and couplings. In the case of pseudo-likelihood maximization, one maximizes the penalized objective

$$\tilde{\ell}(\{h, J\}) = \ell_{\text{PL}}(\{h, J\}) - \lambda_h \sum_{i,a} h_i(a)^2 - \lambda_J \sum_{i < j} \sum_{a,b} J_{ij}(a,b)^2, \quad (2.47)$$

where λ_h and λ_J are regularization strengths for fields and couplings, respectively. This procedure is equivalent to introducing Gaussian priors on the parameters and has the effect of shrinking them towards zero, thereby reducing variance at the cost of a small bias. Moreover, in the absence of a specific gauge imposed a posteriori, the choice of the regularization strength λ_h, λ_J effectively fixes the gauge, as shown in Appendix Section A.3. Another form of regularization widely used in DCA is the introduction of pseudo-counts in the empirical frequencies. This amounts to mixing the observed frequencies with a uniform background,

$$\begin{cases} \tilde{f}_i(a) = (1 - \lambda) f_i(a) + \lambda \frac{1}{q} \\ \tilde{f}_{ij}(a,b) = (1 - \lambda) f_{ij}(a,b) + \lambda \frac{1}{q^2} \end{cases}, \quad (2.48)$$

where $\lambda \in [0, 1]$ is the pseudo-count parameter [23]. This ensures that no frequency vanishes, improving numerical stability in covariance estimation and mean-field approximations. Regularization is thus indispensable to make Potts inference a well-posed inverse problem.

Together, regularization and reweighting allow the approximate inference schemes described in Section 2.4 to yield reliable estimates of Potts couplings, paving the way for their applications in structure prediction and mutational effect modeling, which will be discussed in the next Section.

2.6 Applications of Direct Coupling Analysis

The statistical inference schemes introduced in Section 2.4, complemented by the practical corrections discussed in Section 2.5, yield estimates of the fields and couplings of the Potts model that describe the sequence variability of a protein family. Beyond their conceptual interest, these models have found a wide range of applications in computational biology as they are built to capture the evolutionary constraints that define the structural and functional dependencies that proteins must satisfy in order to correctly fold and function within an organism.

Because of this, the earliest and most prominent application is the prediction of residue-residue contacts in protein structures, which provided a breakthrough in structural bioinformatics well before the advent of deep learning. More recently, Potts models have been used as generative models to design synthetic protein sequences and as predictors of mutational effects by comparing the statistical energies of wild-type and mutant proteins. In this section, we discuss the main applications of Direct Coupling Analysis: contact prediction, generative modelling, and mutational effect prediction.

2.6.1 Contact Prediction

The first and most influential application of Direct Coupling Analysis has been the prediction of residue-residue contacts in protein structures. The central idea is that structural proximity leaves an evolutionary signature in the form of correlated mutations across multiple sequence alignments. Before DCA was introduced, a natural first attempt to quantify such correlations was by using the mutual information (MI) between different columns of an alignment, defined as

$$\text{MI}_{ij} = \sum_{a,b} f_{ij}(a,b) \log \frac{f_{ij}(a,b)}{f_i(a)f_j(b)} \quad , \quad (2.49)$$

where $f_i(a)$ and $f_{ij}(a,b)$ are the empirical single-site and pairwise frequencies. Mutual information measures how much knowing the residue at site i reduces the uncertainty about the residue at site j . Although easy to compute and conceptually appealing, MI does not disentangle direct correlations from indirect ones. For instance, two sites may appear correlated simply because both are correlated with a third position.

Direct Coupling Analysis was introduced precisely to overcome this limitation, by inferring a global Potts model that explains the observed correlations while distinguishing direct from transitive effects. The original scoring function proposed in this context was the Direct Information (DI) [22], defined as the mutual information computed from the two-site distribution generated by the inferred couplings. Since this construction requires additional steps, we postpone its detailed derivation to the appendix. Although DI provides a principled, gauge-invariant measure of direct correlations, its computation requires solving an additional self-consistency problem for every site pair, which makes it computationally expensive; for this reason, later DCA pipelines have largely replaced it with simpler and faster scoring functions such as the Frobenius norm.

Given the couplings $J_{ij}(a,b)$ inferred from the model, the Frobenius norm of the coupling matrix provides a direct way to assign a scalar interaction score to each pair (i,j) :

$$F_{ij} = \sqrt{\sum_{a,b} J_{ij}(a,b)^2}. \quad (2.50)$$

Since the Frobenius norm depends on the chosen gauge, it is customary to fix the parameters in the zero-sum gauge before computing scores. This choice is not arbitrary: the zero-sum gauge is precisely the one that minimizes the Frobenius norm of each coupling matrix $J_{ij}(a,b)$, leaving only the irreducible pairwise interaction signal. As a result, scores obtained in this gauge are both comparable across site pairs and free of redundant contributions that could otherwise be absorbed into the fields.

While the Frobenius norm already provides a useful ranking of contacts, systematic background effects remain: highly conserved positions tend to accumulate large scores regardless of whether they are in spatial contact. To correct for this, the

Average Product Correction (APC) is applied:

$$F_{ij}^{\text{APC}} = F_{ij} - \frac{\left(\sum_k F_{ik}\right)\left(\sum_k F_{kj}\right)}{\sum_{k,\ell} F_{k\ell}}. \quad (2.51)$$

This correction effectively removes global biases and highlights specific pairwise signals.

In practice, the Frobenius norm combined with APC has become the standard DCA score for contact prediction, due to its robustness and ease of computation. Residue pairs with the largest corrected scores provide reliable contact maps that have been successfully used to guide protein structure prediction. The quality of a predicted contact map is typically quantified by the positive predictive value (PPV), defined as the fraction of correctly predicted contacts among the top N ranked pairs. In the following chapters, this measure will be used to compare different inference schemes and scoring functions. The alternative notations $\text{PPV}(N)$ and $\text{PPV}@N$ will be used interchangeably, following common practice in the literature.

2.6.2 Generative Modeling

A key feature of Potts models is that they define a full probability distribution over sequence space,

$$P(\mathbf{a}) = \frac{1}{Z} \exp\left(\sum_i h_i(a_i) + \sum_{i<j} J_{ij}(a_i, a_j)\right), \quad (2.52)$$

where each sequence $\mathbf{a} = (a_1, \dots, a_L)$ is assigned a statistical weight. In the cases in which the inference can be performed exactly, such as for bmDCA and arDCA, this makes the model inherently generative: new sequences can be sampled directly from the distribution $P(\mathbf{a})$.

The generative potential of coevolutionary models was already recognized in early work. In particular, the GREMLIN method was introduced in [60] as a Markov Random Field model for protein families that learns by pseudo-likelihood maximisation with L_1 regularisation. In this framework, novel sequences can be generated by Gibbs sampling: at each step, a site i is chosen and resampled from its conditional distribution given the rest of the sequence, $P(a_i \mid \mathbf{a}_{\setminus i})$. While GREMLIN

primarily used the generative distribution to assign likelihoods and discriminate natural from random or functional from non-functional sequences, it demonstrated that coevolutionary models capture the statistical constraints that shape protein families, and thus can, in principle, be used to generate artificial but biologically realistic sequences. More recent methods have expanded this idea. Among these, in bmDCA [55], a Monte Carlo sampling approach is used explicitly to generate ensembles of artificial sequences that reproduce the one- and two-point statistics of the natural alignment. This generative aspect has been validated experimentally: in particular, Russ et al. [25] demonstrated that bmDCA-generated sequences of the enzyme chorismate mutase can fold and function *in vivo*. As already described in Section 2.4, the autoregressive formulations of DCA (arDCA) [26] are also generative by construction, since they model the sequence probability as a product of tractable conditionals that can be sampled sequentially.

Regardless of their formulation, the quality of generative models can be assessed by comparing statistical properties of generated and natural sequences. Among these, the most informative quantity is the connected correlation

$$C_{ij}(a, b) = f_{ij}(a, b) - f_i(a)f_j(b) \quad , \quad (2.53)$$

which measures the statistical dependency between positions in the alignment beyond what is expected from their independent conservation. A reliable generative model should reproduce the connected correlations observed in the natural alignment, ensuring that the ensemble of artificial sequences captures the same pairwise structure as the data. Another common validation strategy is to project both natural and artificial sequences onto the leading principal components of sequence space and verify that their distributions overlap, indicating that the large-scale organization of the family has been faithfully recovered. These validation procedures, along with the autoregressive implementation of DCA that plays a central role in the second part of this thesis, will be revisited and explored in greater detail in later Chapters 4 and 6.

In summary, generative modeling shows that Potts models do not merely describe observed statistics, but also capture the underlying constraints that define viable protein sequences. Next, we will see how these very same biological constraints can be leveraged to evaluate mutations on existing proteins.

2.6.3 Prediction of Mutational Effects

Mutations are alterations in the amino acid sequence of a protein, typically arising from point substitutions in the underlying genetic code. At the molecular level, a single substitution can have different consequences: due to redundancy at the level of the codon-to-amino-acid mapping, it may be silent, leaving the protein structure and function essentially unchanged; it may be deleterious, destabilizing the fold or disrupting active sites; or it may even be beneficial, improving stability, activity, or conferring new functions. Due to the fact that proteins operate under tight structural and functional constraints, most mutations are expected to reduce fitness, and only a small fraction are tolerated or advantageous. Understanding and predicting these effects is central both to evolutionary biology and to applied fields such as protein engineering and variant interpretation.

As mentioned before, a well-inferred statistical model of biological sequences should capture the evolutionary constraints that shaped the variability observed within a protein family. These constraints can then be leveraged to distinguish sequences that are statistically compatible with the family from sequences that are unlikely under the inferred model, providing a natural framework to evaluate the effect of mutations. In the case of a Potts model, each sequence $\mathbf{a}$ is assigned a probability

$$P(\mathbf{a}) = \frac{1}{Z} e^{-H(\mathbf{a})}, \quad (2.54)$$

where $H(\mathbf{a})$ is usually referred to as the statistical energy of the sequence. This terminology should be understood in a precise but limited sense. The quantity $H(\mathbf{a})$ is the effective Hamiltonian of the maximum-entropy model inferred from homologous sequences, and therefore corresponds to a negative log-probability up to the additive constant $\log Z$. It should not be interpreted as a physical folding free energy, nor as a direct thermodynamic free energy associated with a microscopic molecular system. Rather, it measures how compatible a sequence is with the statistical constraints learned from the natural family. Within this interpretation, the effect of a mutation can be scored by comparing the statistical energy of the mutant sequence with that of the reference sequence:

$$\Delta E(\mathbf{a}^{\text{nat}} \rightarrow \mathbf{a}^{\text{mut}}) = H(\mathbf{a}^{\text{mut}}) - H(\mathbf{a}^{\text{wt}}) \quad . \quad (2.55)$$

A positive ΔE means that the mutant has lower probability under the model than the wild type, and is therefore less compatible with the evolutionary constraints encoded in the MSA. Conversely, a negative ΔE indicates that the mutant is assigned a higher probability by the model. The predictive power of this mutational score can be evaluated against experimental data from deep mutational scanning (DMS). In these high-throughput assays, libraries of single- or multiple-site mutations are expressed and assayed in parallel, yielding quantitative measurements of the functional impact of thousands of variants. The precise notion of functionality probed by DMS depends on the protein family under study: in enzymes it may correspond to catalytic activity, in binding domains to recognition specificity, or in structural proteins to stability. For example, in the case of the TEM-1 β -lactamase, DMS experiments measure the ability of variants to confer antibiotic resistance, while for viral surface proteins such as influenza hemagglutinin they quantify binding and immune escape, and for metabolic enzymes like chorismate mutase they assess catalytic efficiency [61–63].

DMS experiments thus provide an empirical map of the fitness landscape of a protein, against which computational predictions can be directly benchmarked. Potts models have been shown to achieve good agreement with DMS data across diverse protein families [26, 64, 65]. This success can be traced to their epistatic nature. Indeed, even just for single-site mutations $(a_1, \dots, a_i, \dots, a_L) \rightarrow (a_1, \dots, b_i, \dots, a_L)$, the mutational score depends on the entire background of the wild-type sequence, accounting for the fact that the effect of a mutation at one site depends on the state of other positions in the sequence:

$$\Delta E(a_i \rightarrow b_i) = h_i(a_i) - h_i(b_i) + \sum_{j \neq i} [J_{ij}(a_i, a_j) - J_{ij}(b_i, a_j)] \quad . \quad (2.56)$$

Simpler models that neglect epistasis, such as profile models that assume independent sites, are able to capture conservation but fail to reproduce coevolution. Consequently, profile-based mutational scores correlate much more weakly with experimental measurements.

The predictive accuracy of Potts models is often comparable to, though generally below, that of modern deep learning approaches. Methods such as DeepSequence, which parameterize more flexible latent-variable generative models, can reach state-of-the-art performance in DMS benchmarks. However, these models come at the cost of interpretability, as the relationship between their internal parameters and protein structure or function is less transparent than in Potts models.

Beyond DMS validation, mutational scoring based on Potts models has been applied in several related contexts. Examples include predicting changes in thermodynamic stability ($\Delta\Delta G$), distinguishing disease-associated variants from benign polymorphisms, identifying specificity-determining residues in protein families, and characterizing determinants of protein-protein interaction specificity. In addition to static comparisons with DMS, Potts-based mutational scoring has also been extended to model the outcome of directed evolution experiments, in which large mutant libraries undergo iterative rounds of mutation, selection, and amplification. By embedding statistical energies into a dynamical description of variant enrichment, these models link mutational scores not only to endpoint fitness measurements but also to the temporal trajectories of sequence populations under laboratory evolution.

In conclusion, Potts models provide a natural and interpretable framework to quantify mutational effects from sequence data by leveraging the very same evolutionary constraints used for structural modeling. Sequence variation can then be connected to protein fitness, bridging the gap between statistical inference and experimental measurements of function.

2.6.4 Other Applications

While contact prediction, generative modeling, and mutational effect prediction represent the principal applications of DCA specifically relevant to this thesis, a range of additional uses have been explored in recent years. These exploit the ability of Potts models to capture coevolutionary dependencies not only within single protein families but also across different biological macromolecules and functional contexts.

One important extension is the study of protein-protein interactions [23]. In particular, by building joint models over pairs of multiple sequence alignments, coevolutionary couplings can reveal which proteins interact and identify the residues in spatial proximity at their interfaces. Applications include bacterial two-component systems, ribosomal proteins, and viral complexes such as the HIV Env glycoprotein with CD4 receptors [22, 66–68]. Analogous strategies have been applied to protein-RNA and protein-DNA systems, enabling the prediction of binding specificity and interaction surfaces. The same principles have also been used for RNA families. Coevolutionary analysis of RNA alignments can identify base-pairing partners and

long-range contacts, aiding the inference of secondary and tertiary structures of ribozymes, riboswitches, and other non-coding RNAs [69]. Within proteins, Potts couplings have been used to detect networks of coevolving residues associated with allosteric communication and long-range functional coupling [45, 70, 71]. They also provide a means to identify specificity-determining residues that distinguish functional subfamilies or paralogous proteins.

Finally, DCA-inspired models have also been explored in the context of phylogenetic inference. Standard phylogenetic methods typically assume independent evolution at each site, an approximation that neglects the coevolutionary dependencies uncovered by Potts models. By incorporating correlated substitutions into evolutionary models, DCA-based approaches can, in principle, improve both the accuracy of sequence simulation and the reconstruction of phylogenetic trees. Recent studies have investigated how pairwise couplings can be integrated into substitution models to account for correlated mutations, leading to more realistic evolutionary dynamics [72]. Although still at an early stage compared to classical phylogenetic approaches, these developments illustrate how information from Potts models may be used not only to describe present-day sequence ensembles but also to refine our understanding of their evolutionary history.

In summary, the scope of DCA extends well beyond its original applications. By leveraging correlated variation across sequences, Potts models provide insights into biomolecular interactions, functional networks, and evolutionary dynamics, making them a versatile tool across many areas of computational biology.

Chapter 3

Protein Language Modeling and the Attention Mechanism

In this chapter, we trace the historical sequence of models and architectures that, starting from natural language processing, led to the development of the attention mechanism and its application to protein analysis, now commonly referred to as Protein Language Modeling. In particular, we focus on AlphaFold and the inner mechanism at its core that will be the focus of our investigation in the following chapter. We conclude by analyzing a recent work by Rende et al. [73] in which a formal mapping between the generalized Potts model and the attention mechanism is presented.

3.1 Natural Language Processing

Natural languages are complex systems that humans developed to communicate and express meaning. Although we are far from a complete understanding of the mechanisms at their origins, we can start with the assumption that they have emerged organically over time, shaped by cultural and social dynamics rather than by explicit design. As a result, they often exhibit irregularities, ambiguities, and exceptions, but also an extremely vast event space, or space of possibility, which makes them expressive enough to describe virtually any conceivable scenario or concept. This stands in contrast to artificial languages, such as programming languages or formal logic, which are deliberately constructed to be unambiguous and strictly rule-based.

The complexity and flexibility of natural languages make them powerful for human communication, but at the same time pose unique challenges when we try to model them computationally [74].

Natural Language Processing (NLP) is the field of machine learning that takes on this challenge. Its purpose is to design models that can interpret and generate human language in a meaningful way. This broad aim is manifested in many concrete tasks: predicting the continuation of a sentence, translating a text into another language, identifying sentiment in a review, or summarizing long passages into concise statements. All these tasks share a common requirement: the model must capture both the local structure of language, such as the grammatical relationships between neighboring words, and the global coherence that extends across longer contexts, such as the consistency of meaning throughout an entire paragraph. From a computational perspective, natural language can be described as a sequence of symbols chosen from a finite vocabulary. Each sentence is one such sequence, and the richness of language arises not from the symbols themselves but from the patterns of dependency that connect them. These dependencies can be short-range, as in subject–verb agreement, or long-range, as when the interpretation of a pronoun depends on a noun introduced several sentences earlier. The task of NLP is therefore to model probability distributions over sequences of words in such a way that coherent, grammatical, and meaningful sentences are assigned higher likelihood than incoherent ones [74, 75].

In the following subsections, we will review the first statistical and deep learning approaches to this problem. As we will see, these early models exploited the sequential nature of language and provided the foundations for modern NLP, but they also carried structural limitations that would later motivate the introduction of so-called attention mechanisms.

3.1.1 Early statistical and shallow models

The first attempts at computational modeling of natural language were grounded in relatively simple statistical approaches. Among the most influential of these were the so-called n -gram models, which represented language as a Markov process of fixed order [76]. In such models, the probability of observing a word was assumed to depend only on the previous $n - 1$ words. For example, a bigram model ($n = 2$)

would predict each word solely on the basis of its immediate predecessor, while a trigram model ($n = 3$) would extend the context to two preceding words. These models were conceptually straightforward and easy to estimate from data, which made them attractive in the early years of NLP. Yet the very simplicity of n -gram models quickly revealed their limits. Language is rich in long-range dependencies, such as subject-verb agreement across clauses or the use of pronouns that refer back to nouns introduced much earlier in a text. Restricting context to a handful of preceding words inevitably misses such relations. Furthermore, the number of possible n -grams grows exponentially with n , which means that even with very large collections of text, the majority of potential sequences will never occur in the data. Various smoothing techniques, such as Laplace smoothing or the more sophisticated Kneser-Ney method [76, 77], were introduced to alleviate this problem, but they could not fundamentally overcome the locality assumption inherent in the n -gram framework.

In parallel, other shallow representations were explored for tasks in which full sequence modeling was not essential. A common approach was the bag-of-words representation [78], in which a sentence or a document is described simply by the set of words it contains, often encoded as frequency counts or binary indicators. While effective for coarse tasks such as topic classification or sentiment analysis, this representation discards word order entirely and is therefore incapable of capturing syntactic or semantic structure.

Attempts were also made to move beyond shallow statistics by embedding words into vector spaces that captured patterns of co-occurrence. Methods such as Latent Semantic Analysis (LSA) [79] represented words and documents in a lower-dimensional space derived from matrix factorization, providing the first glimpses of semantic similarity in distributed representations. However, these approaches remained fundamentally limited: they could capture general associations between words, but not the sequential and hierarchical properties of natural language.

Taken together, these early statistical and shallow models established important foundations for NLP. They demonstrated that useful patterns could be extracted from large texts and that probabilistic reasoning could be applied to human language. At the same time, their limitations, i.e. locality, sparsity, and lack of true compositional structure, made it clear that more expressive models would be required to capture the full complexity of language. This realization set the stage for the introduction

of neural sequence models, which attempted to overcome these shortcomings by modeling sentences as dynamic processes unfolding over time.

3.1.2 Neural sequence models before attention

The exploration of neural architectures provided more flexible representations of sequential data. In particular, a first natural step was to represent sentences as ordered lists of basic units, called tokens. Depending on the application, a token can be a single character, a whole word, or a smaller subword fragment. Regardless of the chosen level of granularity, tokens are drawn from a finite vocabulary, and a sentence is described as a sequence of such tokens arranged in a particular order. Before being processed by a neural network, tokens must be represented in a numerical form. A straightforward option is one-hot encoding, where each token corresponds to a basis vector of length equal to the vocabulary size. However, this representation is both sparse and uninformative: all tokens are equidistant and no notion of similarity is preserved. A more effective approach, adopted in most modern models, is to map tokens into dense real vectors known as embeddings, whose dimensionality is usually lower than the original vocabulary size. These embeddings are trainable parameters that can be learned jointly with the model or pre-trained, and they capture semantic relationships between tokens: words or symbols that occur in similar contexts are mapped to vectors that lie close to each other in the embedding space. A notable example is *word2vec* by Mikolov et al. [80], which introduced the idea of learning embeddings by predicting the surrounding context of words in large text corpora. This method showed that statistical regularities in raw text could be translated into geometric structure in the embedding space, and it quickly became a cornerstone of NLP. In this way, embeddings provide a continuous and compact representation of discrete input symbols, serving as the natural starting point for sequence models.

Once language is represented in this form, the challenge is to design a model that can process tokens one after the other while retaining information about what has already been seen. This idea gave rise to Recurrent Neural Networks (RNNs) [81], which became the first widely used neural sequence models in NLP. In an RNN, each token is read sequentially, while the model produces an internal hidden state that is updated at every iterative step. The hidden state serves as a memory of the past, carrying forward contextual information as the sequence unfolds. Formally, the

update can be written as

$$h_t = f(Wx_t + Uh_{t-1} + b) \quad , \quad (3.1)$$

where x_t is the embedded token at position t in the sequence, h_{t-1} is the previous hidden state, f is a non-linear activation function, and W, U, b are trainable parameters. This recurrence makes it possible, at least in principle, to carry on information arbitrarily far in the past. In practice, however, learning long-range dependencies with RNNs proved extremely difficult due to the well-known problems of vanishing and exploding gradients during training. To address these difficulties, more sophisticated recurrent units were introduced. The most influential were the Long Short-Term Memory (LSTM) network and its simplified variant, the Gated Recurrent Unit (GRU) [36, 82]. These architectures augment the basic recurrence with gating mechanisms that regulate the flow of information, enabling the network to retain relevant signals over longer spans while discarding noise. LSTMs in particular became the standard choice in many NLP tasks throughout the 2000s and early 2010s, and significantly extended the effective memory of recurrent models.

A further step forward came with the encoder-decoder framework, also known as sequence-to-sequence (seq2seq) modeling [84]. In its original implementation, which would later be enhanced with an early version of the attention mechanism [41], a recurrent encoder reads an input sequence and compresses it into a fixed-length vector representation. This latent representation is then decoded by a second recurrent network into an output sequence, whose nature depends on the specific task. Figure 3.1 presents a schematic overview of the seq2seq architecture. The encoder-decoder paradigm proved highly successful in tasks such as machine translation, where an input sentence in one language must be transformed into its equivalent in another. Despite these advances, recurrent models retained several structural weaknesses. Because tokens are processed sequentially, training and inference are inherently slow and difficult to parallelize. Even with gating, capturing very long-range dependencies remained problematic, as the influence of distant tokens decayed with time. Finally, the encoder-decoder setup concentrated all information about the input sequence into a single hidden vector, which often proved to be a bottleneck, particularly for longer inputs.

These limitations made it clear that, while recurrent models represented an important step beyond shallow statistical methods, they still fell short of fully capturing the

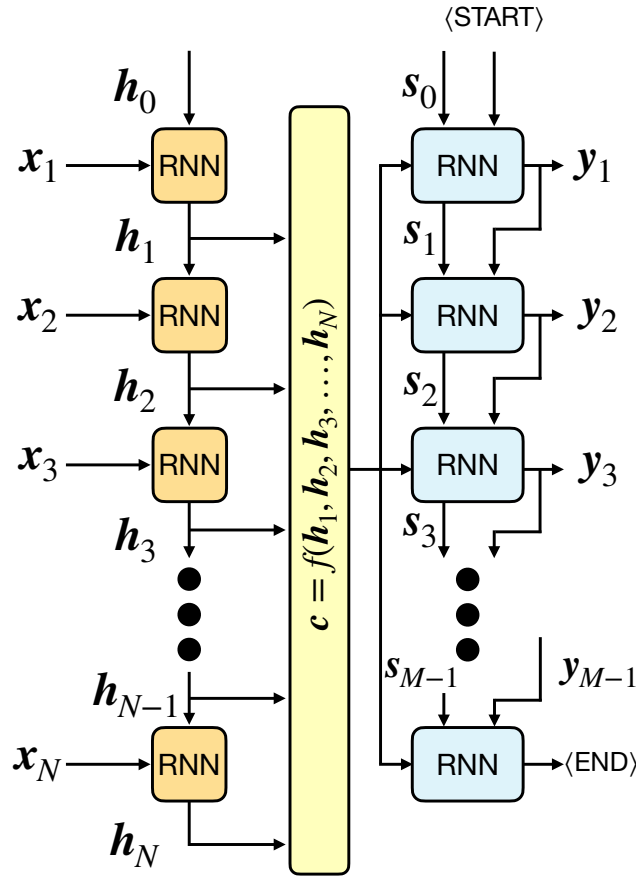


Fig. 3.1 **Encoder-decoder architecture for sequence-to-sequence modeling.** Schematic representation of the encoder-decoder seq2seq architecture used for sequence modeling. An input sequence $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ is processed by a recurrent encoder that maps it into a fixed-length context vector $\mathbf{c} = f(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N)$. The decoder, initialized with this context, generates the output sequence $(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M)$ autoregressively, predicting each token based on the previously generated ones. The $\langle \text{START} \rangle$ and $\langle \text{END} \rangle$ symbols indicate the beginning and end of the generated sequence. Source [83].

complexity of natural languages. The search for architectures that could overcome these barriers eventually led to the introduction of the attention mechanism, which

allowed models to directly access all parts of a sequence in parallel and without the constraints of sequential recurrence. This development will be the subject of the next subsection.

3.1.3 Attention

The notion of attention in machine learning was inspired by biological cognitive attention in animals, i.e. the ability to concentrate on relevant stimuli while filtering out those that are not necessary to produce an appropriate response. In the same spirit, the attention mechanism in a neural network allows a model to focus selectively on specific parts of its input while assigning less weight to other parts. Thus, rather than treating all input positions as equally important, the model learns to distribute its focus on different tokens depending on their relevance for the task at hand. In the context of Natural Language Processing, such mechanisms make it possible for syntactic and semantic relations to emerge more clearly, leading to richer representations of sentence meaning [85].

Attention can be implemented in several ways. To fix ideas, it is useful to first consider the case of self-attention, which is not historically the first formulation but provides the clearest illustration of how attention works practically and will later play a central role in the definition of the Transformer architecture. Self-attention relates different positions of the same sequence in order to produce a new representation that highlights the relative importance of each element with respect to the others. Consider an input sequence of N tokens each embedded in a d -dimensional vector space

$$X = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d,N}, \quad \mathbf{x}_i \in \mathbb{R}^d \quad .$$

A self-attention layer computes a row-stochastic matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, whose entry A_{ij} expresses how much the i -th element attends to the j -th one, i.e. how much attention token $\mathbf{x}_i$ should pay to token $\mathbf{x}_j$. Using this matrix, the input sequence is mapped to a new representation that encodes the self-attention information:

$$\mathbf{y}_i = \sum_{j=1}^N A_{ij} \mathbf{x}_j \quad .$$

The key step is the computation of the attention weights A_{ij} , which requires a measure of similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. In simple self-attention, this is achieved through a

(scaled) dot product,

$$A = \text{softmax}(X W X^\top) \quad ,$$

where W is a trainable parameter matrix learned by backpropagation, and the softmax function is a non-linear activation function of the form

$$\text{softmax}(M)_{ij} = \frac{e^{M_{ij}}}{\sum_k e^{M_{ik}}} \quad . \quad (3.2)$$

As mentioned before, from a historical point of view, the first use of attention in sequence models did not take the form of self-attention. It appeared within recurrent encoder-decoder architectures, most notably in the additive attention mechanism proposed by Bahdanau et al. [41], followed shortly afterwards by the multiplicative (dot-product) variant. In these models, an attention layer was placed between the encoder and decoder to provide a different context vector at every decoding step. This is in contrast to what happens in the original seq2seq model, where the context vector is fixed after the encoding phase and passed statically to the decoder. Figures 3.1 and 3.2 show the differences between the two approaches, the original seq2seq architecture and the attention-based version by Bahdanau et al. [41]. To better explain this, consider the hidden states of the encoder and decoder RNNs $\mathbf{h} = [\mathbf{h}_1, \dots, \mathbf{h}_N]$ and $\mathbf{s} = [\mathbf{s}_1, \dots, \mathbf{s}_M]$, as defined in Section 3.1.2. In the most general case, vectors $\mathbf{h}_i$ and $\mathbf{s}_j$ live in different vector spaces $\mathbb{R}^{d_e}$ and $\mathbb{R}^{d_d}$ respectively. Moreover, the number of sequential encoding and decoding steps, N and M , depends on the lengths of the input and output sequences, for which, generally, $N \neq M$ as in the case of text summarization or sentiment analysis. In this framework, at each decoding step i , the decoder produces a context vector that depends on the previous decoder hidden state $\mathbf{s}_{i-1}$ and on the full set of encoder hidden states. Formally, the context vector is defined as

$$\mathbf{c}_i = \sum_{j=1}^N \alpha_{ij} \mathbf{h}_j \quad ,$$

where the weights α_{ij} are the entries of an attention matrix $\alpha \in \mathbb{R}^{M \times N}$, obtained by normalizing a set of alignment scores:

$$\alpha_{ij} = \text{softmax}(f(\mathbf{s}_{i-1}, \mathbf{h}_j)) \quad . \quad (3.3)$$

Here f is a small neural function (typically a feed-forward network) that scores the compatibility between the previous decoder hidden state $\mathbf{s}_{i-1}$ and each encoder state

$\mathbf{h}_j$. The particular form of the neural function f distinguishes between additive and multiplicative attention: in the first case, encoder and decoder states are combined through additive interactions, whereas in the second case, the similarity is measured directly through a dot product between hidden states. Despite their differences, both mechanisms share the same principle: the decoder forms a context vector by weighting the encoder states according to their relevance at each step of generation, so that α_{ij} represents how much $\mathbf{s}_{i-1}$ attends to $\mathbf{h}_j$ when predicting the next token.

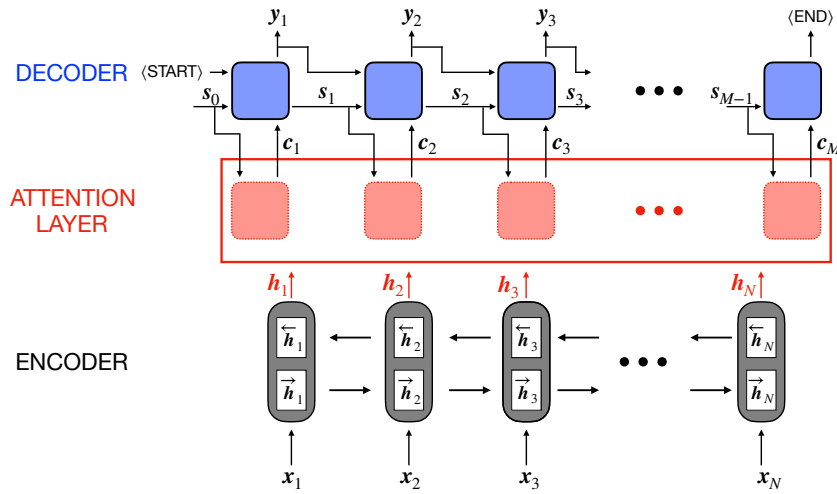


Fig. 3.2 **Additive attention in encoder-decoder models.** Schematic representation of the encoder-decoder architecture with additive attention, as introduced by Bahdanau *et al.* [41]. The encoder processes the input sequence $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ into hidden states $(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N)$. At each decoding step t , the attention mechanism computes alignment scores between the current decoder state $\mathbf{s}_{t-1}$ and each encoder state $\mathbf{h}_j$, producing normalized weights α_{tj} as described in Equation 3.3. These weights define a context vector $\mathbf{c}_t = \sum_j \alpha_{tj} \mathbf{h}_j$, which summarizes the relevant information from the input for predicting the next token $\mathbf{y}_t$. This dynamic weighting allows the decoder to focus on different parts of the input sequence during generation. Source [83].

These early formulations of attention, whether additive or multiplicative, transformed the way encoder-decoder models processed sequences. By allowing the decoder to access the entire set of encoder states dynamically, they overcame the limitations of a single fixed-length representation and significantly improved performance in tasks such as machine translation. At the same time, the alignment weights introduced by attention provided a degree of interpretability, since they could be visualized as soft mappings between input and output tokens. The success of these

mechanisms suggested that attention could serve not merely as an auxiliary component to recurrent models, but as the central operation of a new architecture. This idea would soon materialize in the Transformer, which dispensed with recurrence altogether and placed attention at the core of natural language modeling.

3.1.4 Attention as a soft dictionary

A unifying way to understand different formulations of attention is to think of them as a soft dictionary retrieval system. This analogy becomes clear by comparing the difference between hard and soft dictionaries. In programming languages, a hard dictionary is an associative array composed of keys, each paired with a unique value. To retrieve a value, a query is compared against the keys until a match is found, and the corresponding value is returned. In contrast, a soft dictionary produces a weighted combination of all values, with the weights determined by a measure of compatibility between the query and each key. Formally, given a set of key–value pairs $\{(\mathbf{k}_j, \mathbf{v}_j)\}_{j=1}^N$ and a query $\mathbf{q}$, attention returns

$$\mathbf{y} = \sum_{j=1}^N f_{\text{score}}(\mathbf{q}, \mathbf{k}_j) \mathbf{v}_j \quad , \quad (3.4)$$

where f_{score} is a similarity function followed by a normalization step, typically a softmax that ensures the weights sum to one.

Although this interpretation was implicit in earlier attention mechanisms, the query-key-value formalism was explicitly introduced by Vaswani et al. in the influential 2017 paper *Attention Is All You Need* [40]. In this work, the authors presented the Transformer, an encoder-decoder architecture based entirely on attention and self-attention mechanisms, which enabled efficient in-block processing of word sequences. By dispensing with recurrence, the Transformer effectively solved the major limitation of previous encoder-decoder models: the long training times imposed by sequential computation. Unlike RNN-based approaches, the Transformer can be parallelized across all sequence positions, making training substantially faster. In their formulation, attention for a single query takes the form

$$\text{Attention}(\mathbf{q}, K, V) = \text{softmax}\left(\frac{\mathbf{q}K^\top}{\sqrt{d}}\right)V \quad , \quad (3.5)$$

where K and V collect the keys and values in matrix form, and d is the dimensionality of the key vectors. For an entire set of queries Q , the operation can be computed compactly by means of matrix multiplications as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V . \quad (3.6)$$

This scaled dot-product attention uses a similarity score normalized by a softmax function, which converts the weights into a probability distribution and typically produces sharp focus on the most relevant positions. The matrix formulation allows for efficient implementation with linear algebra libraries and provides a high degree of parallelization that resolves the training speed issues of earlier recurrent models.

The specific nature of queries, keys, and values depends on the task and model. In the case of self-attention, all three are derived from the same input sequence $\mathbf{X} \in \mathbb{R}^{d_x, N}$, through learned linear projections:

$$\begin{aligned} Q &= W_Q^\top X , \\ K &= W_K^\top X , \\ V &= W_V^\top X , \end{aligned} \quad (3.7)$$

where $W_Q \in \mathbb{R}^{d_x, d}$, $W_K \in \mathbb{R}^{d_x, d}$ and $W_V \in \mathbb{R}^{d_x, d_v}$ are trainable parameter matrices. In this way, the model can learn to highlight specific dependencies within the input sequence. In natural language, however, multiple types of dependencies may coexist. For instance, elements of a sentence are in relationship to each other through a wide range of grammatical, syntactical, and semantic rules. To capture such heterogeneous relations, Vaswani et al. proposed multi-head attention. In this design, input X is projected into Q , K , and V multiple times through independent parameter sets (W_Q^h, W_K^h, W_V^h) , one for each attention head $h = 1, \dots, H$. Each head produces its own attention map, and the results are concatenated and projected through a final linear layer to produce the output. This parallel mechanism enables the model to capture different types of dependencies simultaneously, yielding richer and more expressive representations.

In the following subsection, we will complete the introduction of attention in NLP by presenting the implementation details of the Transformer architecture.

3.1.5 Transformer

In the previous subsection, we introduced attention in the query-key-value formalism and its extension to multi-head attention. The Transformer architecture, introduced by Vaswani et al. in 2017 [40], is built entirely around these principles. By eliminating recurrence and convolution, it relies exclusively on attention to model dependencies within and across sequences, while still adopting the encoder-decoder framework familiar from earlier sequence-to-sequence models. Crucially, the Transformer operates block-wise: entire sequences are processed in parallel rather than token by token, marking a fundamental departure from recurrent architectures and significantly lowering the computational cost of processing a single sequence.

In a Transformer architecture, each encoder layer consists of two main components: a multi-head self-attention mechanism and a feed-forward network, both followed by residual connections and layer normalizations. Stacking multiple layers of this type produces contextualized representations in which each token attends to all other tokens, integrating both local and long-range dependencies in parallel. The decoder is structurally similar but includes two key modifications. First, its self-attention is masked, so that each position can only attend to earlier tokens in the output sequence. This is essential during training, when the full target sequence is available in parallel: without the mask, the model would have access to “future” tokens and could simply copy them, defeating the purpose of autoregressive (or masked language) modeling. The mask enforces causality, ensuring that the prediction at step t depends only on positions $< t$. Second, each decoder layer includes an additional multi-head cross-attention block. In this module, the queries come from the decoder states while the keys and values are provided by the encoder outputs, enabling the decoder to condition its predictions directly on the input sequence.

Before entering the first encoder layer, input tokens are mapped into continuous vectors in $X \in \mathbb{R}^{N,d}$ by an embedding layer, as introduced in Section 3.1.2. Due to the block-wise nature of the query-key-value implementation of the encoder-decoder architecture, a Transformer has no inherent notion of the order of the input sequence. In particular, since all operations are performed by means of matrix multiplications of the initial embedded input matrix, swapping its columns does not affect the final output. Because of this, specific positional encodings must be added to the embedded tokens. In the original formulation, positional encodings were deterministic sinusoidal functions of the token index, designed to provide each

position with a unique representation while allowing the model to infer relative positions. Concretely, for feature $i \in \{1, \dots, d\}$ of the positional encoding and position t in the input sequence, one defines $P \in \mathbb{R}^{N,d}$ such that

$$p_{i,t} = \begin{cases} \sin\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is even} \\ \cos\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is odd} \end{cases}, \quad (3.8)$$

The sum of token embeddings and positional encodings $\tilde{X} = X + P$ forms the input to the model. In addition to attention, each Transformer layer contains a simple feed-forward network applied independently to each position, typically consisting of two linear layers with a ReLU activation in between. Despite its simplicity, this component expands the representational capacity of the model. Residual skip connections around each sub-layer, combined with layer normalization, improve gradient flow and stabilize optimization, enabling deep stacking of encoder and decoder layers.

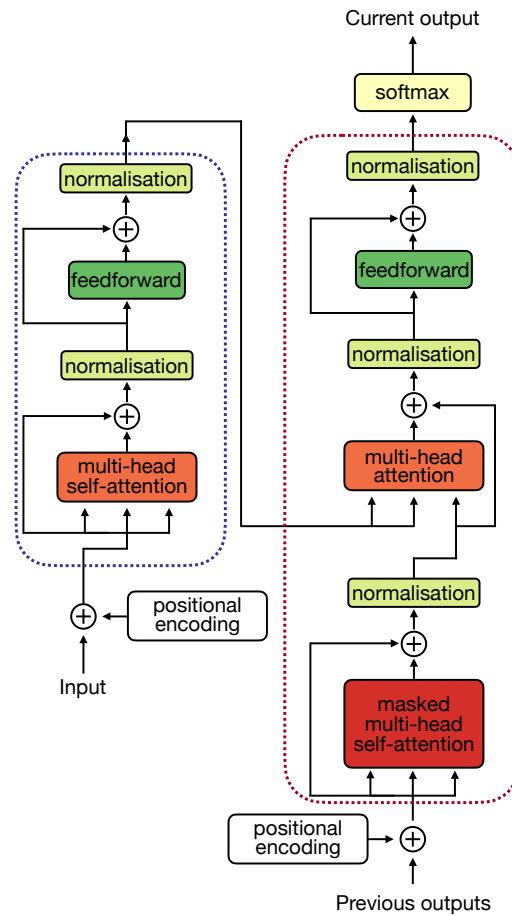


Fig. 3.3 **Schematic overview of the Transformer architecture.** The Transformer model, introduced by Vaswani *et al.* [40], is composed of stacked encoder and decoder blocks that rely entirely on attention mechanisms rather than recurrence or convolution. The encoder consists of a multi-head self-attention module followed by a feed-forward network, with residual connections and normalization layers. The decoder mirrors this structure, adding an additional attention layer that allows the decoder to condition its predictions on the encoder outputs. Positional encodings are added to the input embeddings to provide information about token order. Source [83].

The training procedure follows the autoregressive principle. During training, the entire target sequence is shifted by one position and fed into the decoder as input, a technique known as teacher forcing. The mask ensures that each position is predicted using only past context, despite the model having access to the full sequence in parallel. At inference time, however, tokens must be generated sequentially: the decoder produces one token at a time, and each prediction is fed back as input for the

next step. A representation of the Transformer architecture is shown in Figure 3.3, while Tables 3.1 and 3.2 contain an overview of the operations and parameters inside the encoder and decoder layers of the Transformer.

This design brought several decisive advantages. By eliminating recurrence, the Transformer achieves full parallelization during training, drastically reducing training time compared to RNN-based models. Relying solely on attention, it provides direct pathways for modeling long-range dependencies, unconstrained by sequential processing. Finally, the explicit embedding and positional encoding scheme gives the model flexibility to scale across tasks and vocabularies. The impact of the Transformer was immediate and transformative: it set new standards in machine translation, and in a few years became the foundation of virtually all modern NLP systems, including large pretrained models such as BERT and GPT. More importantly, it established attention as the central mechanism in deep learning. In the next section, we will see how the same ideas can be adapted to biological sequences, where attention provides a natural way to model co-evolutionary signals in proteins.

Table 3.1 Layer-wise overview of the Transformer encoder. The encoder block processes an input sequence through multi-head self-attention followed by a position-wise feed-forward network, with residual connections and layer normalization after each sublayer. In the original Transformer architecture [40], the model dimension satisfies $d_o = d$, $d_k = d_v = d/H$, the softmax is performed on the second index of its argument, and the encoder is composed of $L = 6$ identical layers stacked sequentially.

Stage	Operation	Parameters
Input	Text $[t_1, t_2, \dots, t_N]$ with t_i from vocabulary $\mathcal{V}$	–
Tokenization	$t_i \rightarrow s_i \in [1, \dots, \mathcal{V}]$	–
Embedding lookup	$\mathbf{x}_i = E_{enc}[s_i] \in \mathbb{R}^d$	$E_{enc} \in \mathbb{R}^{ \mathcal{V} \times d}$
Positional encoding	$\mathbf{z}_i = \mathbf{x}_i + \mathbf{p}_i \in \mathbb{R}^d$	$\mathbf{P} \in \mathbb{R}^{N,d}$ (fixed)
Attention input	$\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_N] \in \mathbb{R}^{N,d}$	–
Query map	$\mathbf{Q}^h = \mathbf{Z}\mathbf{W}_Q^h \in \mathbb{R}^{N,d_k}$	$\mathbf{W}_Q^h \in \mathbb{R}^{d,d_k}$
Key map	$\mathbf{K}^h = \mathbf{Z}\mathbf{W}_K^h \in \mathbb{R}^{N,d_k}$	$\mathbf{W}_K^h \in \mathbb{R}^{d,d_k}$
Value map	$\mathbf{V}^h = \mathbf{Z}\mathbf{W}_V^h \in \mathbb{R}^{N,d_v}$	$\mathbf{W}_V^h \in \mathbb{R}^{d,d_v}$
Self-attention	$\mathbf{A}^h = \text{softmax}\left(\frac{\mathbf{Q}^h \mathbf{K}^{h\top}}{\sqrt{d_k}}\right) \mathbf{V}^h \in \mathbb{R}^{N,d_v}$	–
Heads concatenation	$\mathbf{C} = [\mathbf{A}^1, \dots, \mathbf{A}^H] \in \mathbb{R}^{N, Hd_v}$	–
Multi-head attention	$\mathbf{MHA} = \mathbf{C}\mathbf{W}_O \in \mathbb{R}^{N,d_o}$	$\mathbf{W}_O \in \mathbb{R}^{Hd_v, d_o}$
Add & Norm	$\mathbf{Y} = \text{LayerNorm}(\mathbf{MHA} + \mathbf{Z})$	$\gamma_1, \beta_1 \in \mathbb{R}^{d_o}$
Feed-forward	$\text{FF}(\mathbf{Y}) = \text{ReLU}(\mathbf{Y}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2$	$\mathbf{W}_1 \in \mathbb{R}^{d_o, d_{ff}}, \mathbf{W}_2 \in \mathbb{R}^{d_{ff}, d_o}$
Add & Norm	$\tilde{\mathbf{Y}}_{enc} = \text{LayerNorm}(\mathbf{Y} + \text{FF}(\mathbf{Y}))$	$\gamma_2, \beta_2 \in \mathbb{R}^{d_o}$

Table 3.2 Layer-wise overview of the Transformer decoder. The decoder block consists of masked multi-head self-attention, encoder–decoder cross-attention, and a position-wise feed-forward network, each followed by residual connections and layer normalization. As in the original Transformer architecture [40], the model dimension satisfies $d_o = d$, and the decoder is composed of $L = 6$ identical layers stacked sequentially. During training, the output distribution at each position is obtained by projecting the final decoder representation through a linear map whose weight matrix is tied to the decoder embedding matrix, i.e. $W_{lm} = E_{dec}^\top$.

Stage	Operation	Parameters
Input	Target words $[\tilde{t}_1, \dots, \tilde{t}_M]$ with $\tilde{t}_i$ from vocabulary $\mathcal{V}$	–
Shift	$[\tilde{t}_1, \dots, \tilde{t}_M] \rightarrow [\langle \text{Start} \rangle, \tilde{t}_1, \dots, \tilde{t}_{M-1}]$	–
Tokenization	$\tilde{t}_i \rightarrow s_i \in [1, \dots, \mathcal{V}]$	–
Embedding lookup	$\mathbf{x}_i = E_{dec}[s_i] \in \mathbb{R}^d$	$E_{dec} \in \mathbb{R}^{ \mathcal{V} \times d}$
Positional encoding	$\mathbf{z}_i = \mathbf{x}_i + \mathbf{p}_i \in \mathbb{R}^d$	$\mathbf{P} \in \mathbb{R}^{M, d}$ (fixed)
Attention input	$Z = [\mathbf{z}_1, \dots, \mathbf{z}_N] \in \mathbb{R}^{N, d}$	–
Query map	$Q^h = ZW_Q^{h, dec} \in \mathbb{R}^{M, d_k}$	$W_Q^{h, dec} \in \mathbb{R}^{d, d_k}$
Key map	$K^h = ZW_K^{h, dec} \in \mathbb{R}^{M, d_k}$	$W_K^{h, dec} \in \mathbb{R}^{d, d_k}$
Value map	$V^h = ZW_V^{h, dec} \in \mathbb{R}^{M, d_v}$	$W_V^{h, dec} \in \mathbb{R}^{d, d_v}$
Masked self-attention	$MA^h = \text{softmax}(\frac{Q^h K^{h\top}}{\sqrt{d_k}} + M_{\text{mask}}) V^h$	$\{W_Q^{h, dec}, W_K^{h, dec}, W_V^{h, dec}\}$
Multi-head attention	$MHA = [MA^1, \dots, MA^H] W_O^{dec}$	$W_O^{dec} \in \mathbb{R}^{Hd_v, d_o}$
Add & Norm	$Y = \text{LayerNorm}(Z + MHA)$	$\gamma_1, \beta_1 \in \mathbb{R}^{d_o}$
Cross-attention	$MHA_{cross} = MHA(Y, \tilde{Y}_{enc})$ with Q from decoder, K, V from encoder outputs $\tilde{Y}_{enc}$	$\{W_Q^{h, cross}, W_K^{h, enc}, W_V^{h, enc}\}$
Add & Norm	$\tilde{Y} = \text{LayerNorm}(Y + MHA_{cross})$	$\gamma_2, \beta_2 \in \mathbb{R}^{d_o}$
Feed-forward	$FF(\tilde{Y}) = \text{ReLU}(\tilde{Y}W_1 + b_1)W_2 + b_2$	$W_1 \in \mathbb{R}^{d_o, d_{ff}}, W_2 \in \mathbb{R}^{d_{ff}, d_o}$
Add & Norm	$\tilde{Y}_{dec} = \text{LayerNorm}(\tilde{Y} + FF(\tilde{Y}))$	$\gamma_3, \beta_3 \in \mathbb{R}^{d_o}$
Output	$\text{softmax}(\tilde{Y}_{dec}W_{lm} + b_{lm})$	$W_{lm} \in \mathbb{R}^{d_o, \mathcal{V} }, b_{lm} \in \mathbb{R}^{ \mathcal{V} }$

3.2 Protein Language Modeling

The success of attention-based architectures across a wide range of natural language processing tasks inspired a paradigm shift in computational biology: the treatment of protein sequences as a language. In this section, we explore the analogies and mappings between NLP and protein sequence analysis that motivated the development of Protein Language Models (PLMs), setting the stage for breakthroughs such as AlphaFold in protein structure prediction.

3.2.1 From languages to proteins

The idea that proteins can be treated as a language has become increasingly influential in recent years. This analogy is motivated by the observation that protein sequences share many of the properties that make natural languages amenable to modeling with machine learning methods. Both are expressed as discrete symbols drawn from a finite alphabet, both exhibit dependencies that span different ranges of context, and both obey hidden rules that constrain what constitutes a valid sequence. Just as words acquire meaning from their placement within a sentence, amino acids acquire structural and functional roles from their context within a protein [35].

At the most basic level of this analogy, amino acids correspond to the words or characters of a language. Protein sequences are then equivalent to sentences: ordered lists of tokens that together convey meaning. A collection of related proteins, such as a protein family, can be seen as a language, with its own characteristic “grammar” defined by evolutionary and biochemical constraints. Within these sequences, short-range dependencies reflect local biochemical interactions, just as in language, short-range dependencies often encode grammatical rules. In proteins, recurrent patterns of such local interactions give rise to secondary structures, analogous to recurring phrases or idioms in natural language. Long-range dependencies, by contrast, connect distant residues that fold together in three dimensions, much as semantic relations in language can connect words that are far apart in a sentence. At a higher level, the global meaning of a sentence corresponds to the overall structure and function of a protein. In this mapping, grammar corresponds to the biochemical interactions that constrain residue combinations, syntax reflects evolutionary constraints that shape sequence variability, and semantics corresponds to the biological role and structural realization of the protein [86, 87].

The analogy extends beyond representation to the level of tasks. Many of the classical applications of NLP have natural counterparts in protein modeling. Machine translation, which maps a sentence in one language to another, is analogous to modeling interactions between proteins of different families, where information must be exchanged between sequences with distinct “grammars.” Summarization, which compresses long texts into concise statements, corresponds to extracting recurring motifs or domains that capture the essence of a family. Classification tasks, in which sentences or documents are assigned to categories, correspond to the functional labeling of proteins, such as assigning enzyme classes or predicting subcellular localization [86]. These mappings are not perfect, since proteins are not designed for communication but are shaped by evolution and physics, yet they provide a powerful conceptual framework for transferring modeling strategies from NLP to biology.

The usefulness of this analogy has been supported by recent advances in protein language models. Models trained solely on raw sequence data, without structural supervision, have been shown to capture information about secondary and tertiary structure, residue contacts, and even functional properties. This demonstrates that the linguistic analogy is more than metaphorical: it reflects genuine statistical regularities in sequence data that can be exploited for predictive modeling. In the following subsections, we will examine how this perspective has led from early sequence embeddings to large-scale Transformer models, and ultimately to AlphaFold, which combines evolutionary information with attention mechanisms to achieve near-experimental accuracy in structure prediction.

3.2.2 Early protein language models

The development of protein language models began with simple adaptations of methods from natural language processing and gradually advanced toward more sophisticated architectures. This progression illustrates how increasingly expressive models were able to capture the complex dependencies within protein sequences, moving from local statistical representations to large-scale pretraining strategies that revealed structural and functional signals directly from raw data.

An early contribution was ProtVec, proposed by Asgari and Mofrad in 2015 [88], in which protein sequences were decomposed into overlapping k -mers, typically

with $k = 3$, and each fragment was treated as an elementary unit analogous to a word. Using the word2vec algorithm described in Section 3.1, embeddings were learned so that k -mers appearing in similar sequence contexts were placed close to one another in the vector space. By averaging across all k -mers in a sequence, a global representation of the protein could be obtained, which proved effective in tasks such as family classification, enzyme function prediction, and subcellular localization [88]. ProtVec established that proteins could be embedded into continuous spaces much like words in natural language, but its representations were static and local: every k -mer was mapped to the same vector regardless of context, and sequence-level embeddings were derived by simple aggregation. As a consequence, order information and long-range dependencies were largely lost, highlighting the need for models capable of producing contextualized representations of entire sequences.

To overcome these constraints, recurrent neural networks were introduced as a natural extension. By processing amino-acid sequences sequentially, they were able to integrate contextual information over variable lengths. Bepler and Berger (2019) presented a bidirectional LSTM framework trained with structural supervision, in which the sequence encoder was optimized to predict structural similarity between proteins [89]. The resulting embeddings captured information about secondary and tertiary structure and proved useful across a range of downstream tasks, including remote homology detection, contact prediction, and functional annotation. This work illustrated the potential of contextualized protein representations and introduced the idea of transfer learning within computational biology. Yet, as in NLP, recurrent architectures remained limited by slow training, poor scalability to very long sequences, and vanishing gradients that hindered the modeling of distant dependencies.

In parallel, generative approaches to protein modeling were developed in order to estimate probability distributions over sequence families, merging deep learning and statistical inference techniques. A prominent example is DeepSequence, introduced by Riesselman et al. (2018), which applied variational autoencoders to multiple sequence alignments. By learning latent representations that captured the evolutionary constraints of a family, the model could assign likelihoods to variants, and these likelihoods correlated with mutational fitness data obtained experimentally [32]. This connection established a statistical link between unsupervised modeling of sequence variation and the functional consequences of mutations. Related autoregressive and probabilistic frameworks extended this principle, reinforcing the idea that generative modeling provides direct access to evolutionary information.

The introduction of standardized evaluation benchmarks marked an important step in consolidating progress. Rao et al. (2019) released the TAPE benchmark [37], a collection of tasks designed to assess protein models across a variety of problems, including secondary structure prediction, contact prediction, remote homology, and fluorescence estimation. By providing a common reference, TAPE enabled systematic comparison across embedding methods, recurrent architectures, and the first Transformer-based models. Results showed that pretraining on large sequence databases yielded representations that generalized well across tasks, even when training objectives were as simple as masked language modeling.

These developments prepared the ground for the adoption of Transformers in protein modeling. Elnaggar et al. (2020) introduced the ProtTrans suite, which included BERT- and GPT-inspired architectures trained on billions of sequences from UniRef and BFD [90]. ProtBERT became a widely used resource, showing that masked language modeling at scale could be applied directly to proteins. In parallel, Rao et al. (2020) presented the ESM series, trained on hundreds of millions of sequences [91]. These models revealed that structural features such as secondary structure, residue contacts, and even aspects of tertiary organization emerged naturally from sequence-only training. Interpretability analyses added further insight: Vig et al. (2021) demonstrated that attention heads aligned with residue contacts, binding sites, and post-translational modifications, while deeper layers encoded more global structural and functional properties [92].

In summary, early protein language models evolved from shallow embeddings-based models to recurrent and generative architectures, and finally to large-scale Transformers. Each stage contributed to the growing realization that evolutionary and structural information is encoded in raw sequence data and can be extracted through appropriate models. This trajectory culminated in the integration of attention with co-evolutionary analysis in AlphaFold, which transformed protein structure prediction, effectively solving the protein folding problem from a computational point of view.

3.2.3 AlphaFold

As previously described in the introductory Section 1.2, the problem of predicting the three-dimensional structure of a protein from its amino-acid sequence, known as

the protein folding problem, has been a central challenge in computational biology for decades and, as such, has seen a wide variety of approaches and techniques to solve it. Among these, classical approaches included homology modeling, which builds structures by analogy to known templates; threading, which aligns sequences to existing folds; and physics-based simulations grounded in molecular dynamics. While each of these strategies achieved partial success, their accuracy and scalability remained limited, particularly for proteins without close structural homologues. A fundamental improvement came with the incorporation of evolutionary information, as seen in Section 2.4. By analyzing correlated mutations across multiple sequence alignments, co-evolutionary approaches such as Direct Coupling Analysis were able to infer residue–residue contacts, which in turn guided structure prediction. This statistical foundation provided the key signal that would later be integrated into deep learning frameworks.

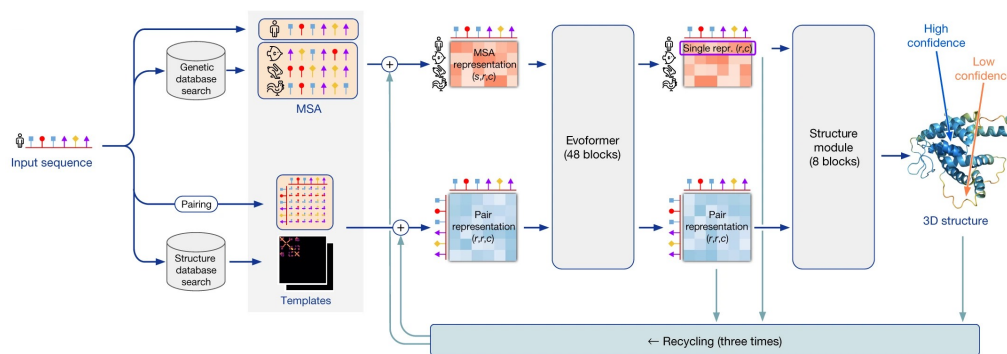


Fig. 3.4 Schematic overview of the AlphaFold architecture. The model takes as input a target sequence and a multiple sequence alignment (MSA) of homologous proteins. These inputs are processed through the Evoformer blocks, which iteratively exchange information between sequence and pairwise representations. The refined pair representation is then passed to the structure module, which predicts the three-dimensional atomic coordinates of the protein. The architecture integrates evolutionary information, geometric reasoning, and attention mechanisms into a unified end-to-end framework for protein structure prediction. Source [19].

The first breakthrough in this direction was AlphaFold1, presented by DeepMind at CASP13 (Critical Assessment of Structure Prediction) in 2018 [18]. The model combined deep residual convolutional networks with co-evolutionary features derived from MSAs, predicting distance distributions between residues and assembling them

into three-dimensional models. AlphaFold1 substantially outperformed competing methods, marking the entry of deep learning into protein structure prediction. Yet, the choice of using convolutional architectures left room for further innovation. That innovation arrived two years later with AlphaFold2, unveiled at CASP14 in 2020 [19]. Shown in its schematic representation in Figure 3.4, AlphaFold2 represented a complete redesign of the original architecture, with attention mechanisms at its core. The central innovation lies in the Evoformer block, a module that processes two parallel and interdependent representations. The first is the so-called MSA representation, which encodes information across homologous sequences in a multiple sequence alignment. To update this representation, the Evoformer applies attention operations along both axes of the MSA: row-wise attention integrates information across different homologous sequences at the same residue position, capturing how variation at a site is distributed in evolution; column-wise attention integrates information along the length of each sequence, allowing residues at different positions to attend to each other. In this way, the MSA representation combines two complementary views: the vertical dimension, which reflects the evolutionary variability of each site, and the horizontal dimension, which reflects dependencies between positions in the sequence. Column attention is particularly significant, since it directly models correlations between positions within a protein, analogous to the couplings extracted by co-evolutionary analysis. It is precisely this mechanism that will form the focus of the following chapter, where we connect it to Potts models in the formulation of AttentionDCA. The second track of the Evoformer is the pair representation, which is defined over all residue–residue pairs in the target sequence and can be seen as an $L \times L$ matrix for a protein of length L . Each entry in this matrix stores an embedding that encodes the current model’s belief about the relationship between a pair of residues, including whether they are likely to be in spatial proximity. The pair representation is updated through specialized operations that reflect the geometric constraints of protein structures. Triangle updates and triangle attention layers ensure that the interactions between residues i and j are informed by their relationships with a third residue k , thus modeling the transitive consistency of residue triplets. Information flows continuously between the MSA and pair representations: correlations detected across homologous sequences are projected into pairwise residue constraints, while pair-level updates feed back into the MSA embeddings. After a series of Evoformer blocks, the refined pair representation is passed to a geometry module that predicts three-dimensional coordinates, producing highly accurate struc-

tural models. Figure 3.5 represents a schematic overview of the Evoformer block inside AlphaFold 2.

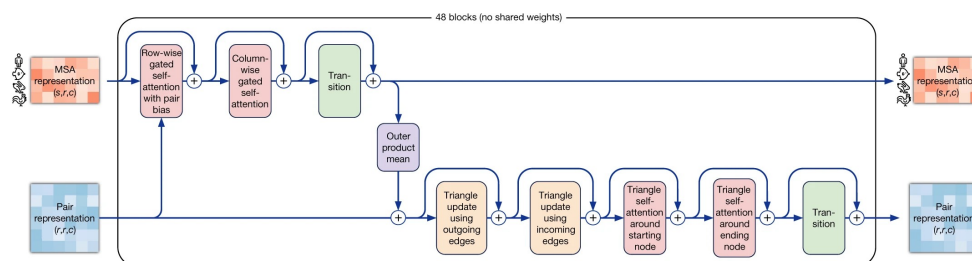


Fig. 3.5 The Evoformer block within the AlphaFold 2 architecture. The Evoformer operates on two coupled representations: the multiple sequence alignment (MSA) representation and the pair representation. Row-wise and column-wise attention update the MSA embeddings by integrating evolutionary and positional information, while the pair representation is refined through triangle updates and attention layers that encode geometric relationships between residues. Iterative communication between the two representations enables the model to infer residue–residue couplings and build consistent structural constraints prior to 3D coordinate prediction by the structure module. Source [19]

The impact of AlphaFold2 was transformative. At CASP14, it achieved near-experimental accuracy for the majority of targets, effectively solving the long-standing protein folding problem. The model has since become a widely used tool in structural biology, enabling new discoveries across a range of applications. Subsequent work extended AlphaFold beyond monomeric folding. AlphaFold-Multimer adapted the architecture for protein–protein complexes, while specialized variants incorporated ligands, RNA, or designed proteins [93]. The most recent iteration, AlphaFold3, further generalized the framework, combining the Evoformer core with diffusion-based structure refinement and extending prediction to protein–ligand, protein–nucleic acid, and multi-component assemblies [94]. These developments indicate that AlphaFold is evolving from a folding engine into a general platform for molecular modeling.

In summary, AlphaFold represents the culmination of decades of progress in protein structure prediction, integrating evolutionary information, attention mechanisms, and geometric reasoning into a unified framework. At its core lies the Evoformer, a Transformer-inspired block that iteratively processes multiple sequence alignments and residue–residue representations, capturing evolutionary variation and structural couplings in a coherent way. Within the Evoformer, column-wise

attention is of particular importance: by modeling dependencies between positions in a sequence, it serves as a modern reformulation of the statistical couplings extracted by co-evolutionary analysis. This connection is central to the present thesis. In the following chapter, we will analyze column attention through the lens of statistical physics, showing how it relates to Potts models and how this analogy motivates the formulation of AttentionDCA.

3.3 Attention and the Generalized Potts Model

The aim of this section is to establish a formal link between attention mechanisms and statistical physics models, a connection that will provide the theoretical foundation for the computational work of Chapter 4. In particular, we focus on the factored form of self-attention, where the explicit dependencies on sequence positions and residue identities are factorized. This variant of attention was first introduced by Bhattacharya et al. (2020), who demonstrated that even a single factored attention layer trained on protein multiple sequence alignments can recover residue–residue contacts with remarkable accuracy [95]. The work presented in Chapter 4 can be regarded as a direct extension of this idea. There, we analyze a single-layer factored attention more systematically, compare the couplings it extracts to those obtained from Direct Coupling Analysis, and further expand the model into an autoregressive generative implementation. Almost concurrently, Rende et al. (2024) developed the theoretical counterpart of this approach [73]. Their study showed that a single layer of factored self-attention trained with site-wise cross-entropy, mathematically equivalent to the pseudo-likelihood maximization used in Potts inference, can be written exactly as a generalized Potts model with low-rank couplings. Beyond establishing this equivalence, they employed replica methods to analyze the sample complexity of attention layers, characterizing the conditions under which the learned couplings recover the underlying statistical structure of the data.

In what follows, we review the main elements of this theoretical framework. Our goal is not to reproduce the technical derivations in full, but to extract the key insights that will allow us to connect factored attention to Potts models and to motivate the formulation of AttentionDCA in the next chapter.

3.3.1 Generalized Potts Model and Self-Attention

We begin by recalling the generalized Potts model formulation used in [73], which is slightly different from the one used in Chapter 2. Consider a sequence of spins $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_L)$ of length L , where each spin $\mathbf{s}_i \in \mathbb{R}^q$ is encoded as a one-hot vector taking values in an alphabet of size q (for proteins, the 20 amino acids plus a possible gap). The generalized Potts Hamiltonian is defined as

$$H(\mathbf{s}) = -\frac{1}{2} \sum_{i,j=1}^L J_{ij} \mathbf{s}_i^T U \mathbf{s}_j, \quad (3.9)$$

where $J \in \mathbb{R}^{L \times L}$ encodes the coupling between positions i and j , while $U \in \mathbb{R}^{q \times q}$ is a similarity matrix that encodes interaction between colors. Without loss of generality, we can drop the field term and neglect self-interactions by setting $J_{ii} = 0$. The probability distribution over sequences is then given by the Boltzmann measure $P(\mathbf{s}) \propto \exp[-\beta H(\mathbf{s})]$, where β is the inverse temperature. Sampling from this distribution generates sequences that reflect the pairwise constraints encoded in the couplings J_{ij} . As discussed in Chapter 2, this is the statistical physics foundation of Direct Coupling Analysis: one can infer the effective evolutionary constraints that shape a protein family by fitting J_{ij} to reproduce the statistics of a multiple sequence alignment.

In Section 2.4, when dealing with Direct Coupling Analysis, we showed that among the computationally feasible techniques to infer the parameters of the generalized Potts model given a set of observed sequences, the pseudo-likelihood maximization (PLM) approach is one of the best in terms of computational efficiency and accuracy. At its core, PLM amounts to learning the distribution of the i -th spin $\mathbf{s}_i$ conditioned to the remaining elements of the sequences $\mathbf{s}_{\setminus i}$. In the context of natural language modeling, this strategy is usually referred to as Masked Language Modeling (MLM). In a Transformer architecture, or even in a simple self-attention layer, training by means of MLM implies masking one or more tokens and letting the model predict the missing elements. Thus, what in statistical learning is known as pseudo-likelihood inference corresponds exactly to masked language modeling in machine learning, the two being different perspectives on the same training principle.

Let us now consider a simple self-attention mechanism, characterized by learnable query, key, and value mappings $Q, K, V \in \mathbb{R}^{d,d}$. The output of a single layer is a

distribution given by:

$$\begin{aligned}\tilde{\mathbf{p}}_i(\mathbf{s}_{\setminus i}) &= \sum_{j=1}^L \frac{\exp[(E\mathbf{t} + \mathbf{p}_i)^\top Q^\top K(\alpha E\mathbf{s}_j + \mathbf{p}_j)]}{\sum_l \exp[(E\mathbf{t} + \mathbf{p}_i)^\top Q^\top K(\alpha E\mathbf{s}_l + \mathbf{p}_l)]} V(E\mathbf{s}_j + \alpha \mathbf{p}_j) \\ &= \sum_{j=1}^L \frac{\exp[\mathbf{q}_i^\top \mathbf{k}_j]}{\sum_l \exp[\mathbf{q}_i^\top \mathbf{k}_l]} \mathbf{v}_j \quad .\end{aligned}\tag{3.10}$$

In this notation, each spin $\mathbf{s}_j$ is embedded onto a more informative vector space through the linear mapping $E \in \mathbb{R}^{d,q}$ and added to a positional encoding $\alpha \mathbf{p}_j$, where α is a parameter that regulates the relative importance of embedding and positional encoding. Spin $\mathbf{s}_i$ is masked by token $\mathbf{t}$, so that the model can predict it without knowing it beforehand. The output of the self-attention layer $\tilde{\mathbf{p}}_i$ can be mapped to a probability distribution over the possible colors of input $\mathbf{s}_i$ by choosing $d = q$.

Rende et al. tested this setup by training a one-layer Transformer on sequences sampled from a generalized Potts model, with a binary symmetric positional interaction matrix $J_{ij} \in \{0, 1\}$ and a color similarity matrix with i.i.d. Gaussian entries $U_{a,b} \sim \mathcal{N}(0, 1)$. Training was performed by stochastic gradient ascent on the cross-entropy loss

$$\mathcal{L} = -\frac{1}{L} \sum_{i=1}^L \sum_{c=1}^q s_{ic} \log \tilde{p}_{ic}(\mathbf{s}) \quad .\tag{3.11}$$

They showed that the resulting attention maps recover the original interaction matrix J , albeit with some blurring. This recovery improves when three consecutive layers are used, in which case the average of the attention maps of the final two layers aligns more closely with the true interactions.

These results highlight both the potential and the limitations of vanilla self-attention when applied to data generated by a generalized Potts model. A single layer can already recover non-trivial information about the interaction structure, and stacking a few layers improves the correspondence with the ground truth couplings. However, the recovery remains approximate, and the learned attention maps tend to be blurred representations of the underlying matrix J . This raises the question of whether a more constrained parametrization of attention could capture the Potts interactions more faithfully. In the next subsection, we turn to the *factored attention* formulation, where positional and color dependencies are explicitly separated, and where the connection to Potts models becomes exact.

3.3.2 Factored Attention

As mentioned earlier, the factored self-attention variant was first introduced by Bhattacharya et al. (2020) as a way to analyse the inner workings of attention layers inside Transformers and Evoformer blocks. In their work they showed that even a single layer of factored attention, when trained on protein families, could recover contact maps comparable to those obtained from Potts models in the Direct Coupling Analysis framework. This provided the first concrete evidence that attention mechanisms and Potts couplings are closely related.

Rende et al. later formalized this connection by proving that a factored attention layer can be written exactly as a Potts model. Starting from Equation 3.10, one obtains the factored form by using one-hot encodings for positions, setting the embedding matrix to the identity $E = I_q$, fixing the masking token to $\mathbf{t} = 0$, and taking $\alpha = 0$. Under these assumptions, the probability distribution simplifies to

$$\tilde{p}_{ia}(\mathbf{s}_{\setminus i}) = \sum_{j=1}^L \frac{\exp[(Q^T K)_{ij}]}{\sum_l \exp[(Q^T K)_{il}]} (V \mathbf{s}_j)_a = \sum_{j=1}^L A_{ij} (V \mathbf{s}_j)_a, \quad (3.12)$$

where A is the attention matrix and V acts on residue identities. The crucial point is that setting $\alpha = 0$ decouples positional and color degrees of freedom: positional interactions are encoded in A , while residue compatibilities are encoded in V .

This new form of attention coincides with the conditional distribution of the pseudo-likelihood approximation to the generalized Potts model,

$$p(s_{ia} = 1 \mid \mathbf{s}_{\setminus i}) = \frac{\exp\left(\beta \sum_{j=1}^L J_{ij} (U \mathbf{s}_j)_a\right)}{\sum_c \exp\left(\beta \sum_{j=1}^L J_{ij} (U \mathbf{s}_j)_c\right)}, \quad (3.13)$$

by identifying $U = V$ and $\beta J = A$. Under the same assumptions as before, they showed that this implementation of factored attention recovers the interaction matrix almost perfectly. Moreover, since the masked-language-modeling objective coincides with Potts pseudo-likelihood, all statistical properties of the latter are inherited by the former. Because of this fact, factored attention enjoys statistical consistency, meaning that in the limit of infinitely many samples, the inferred parameters converge to their true values. Rende et al. went further in their analysis by computing the sample complexity of self-attention using replica methods and validating the

predictions against numerical experiments. They showed that the model displays the characteristic *double descent* curve of generalization error in deep neural networks, highlighting the overfitting regime of the model.

In recent years, factored attention has proved useful well beyond protein modeling, finding applications in computer vision and in the modeling of quantum many-body systems. This breadth further underlines its role as a versatile and principled architecture that bridges machine learning and statistical physics.

This closes the historical and logical series of ideas introduced in this chapter: starting from the analogy between natural languages and proteins, through the development of attention mechanisms and Transformers, and culminating in their formal connection with generalized Potts models. In the next chapter, we build on this theoretical groundwork and present a computational study of factored attention in practice. In particular, we introduce *AttentionDCA*, an extension of classical direct coupling analysis designed as a tool to analyze the behavior of attention layers in the context of protein structure prediction.

Chapter 4

AttentionDCA: Interpretable Attention-based Direct Coupling Analysis

The previous chapters established the statistical foundation of Direct Coupling Analysis and the role of the attention mechanism in protein language models. To exploit the interpretability of the former and the predictive power of the latter, a natural development is to map simplified single attention layers onto DCA formulations. Among these, the factored attention model introduced by Bhattacharya et al. represents a particularly transparent case [95].

In this chapter, we present our own implementation of factored attention, which we refer to as AttentionDCA, to emphasize its connection with the statistical physics tradition of DCA methods. In particular, our contribution lies in systematically analyzing its performance and extending the modalities of contact prediction so as to explicitly link the attention mechanism to protein structure prediction. Moreover, we generalized AttentionDCA to the first instance of DCA multi-family inference, and adapt it into an autoregressive generative model.

The results discussed in this chapter highlight AttentionDCA as a minimal but informative architecture. Its reduced complexity, interpretability through attention matrices, and capacity to integrate information across families make it a useful bridge between classical DCA approaches and the transformer-based deep learning models that dominate current protein structure prediction.

The content, figures and tables presented in this chapter are based on the material from the paper “*Direct Coupling Analysis and the Attention Mechanism*” [42], adapted and expanded for the purposes of this thesis.

4.1 Implementation

In this section, we present the different implementations of AttentionDCA, starting from the reformulation of the standard factored attention introduced by Bhattacharya *et al.* and moving to the novel multi-family and generative versions. Throughout the section, we will define the computational tools and benchmarking metrics that will be used in the Results section to analyze the model.

4.1.1 Standard version

In line with the DCA formulation introduced in Chapter 2, sequences of a protein family are represented by a Multiple Sequence Alignment (MSA) and modeled as independent samples from a Boltzmann distribution,

$$P(\mathbf{a}) = \frac{1}{Z} e^{-H(\mathbf{a})}, \quad (4.1)$$

where $\mathbf{a} = (a_1, \dots, a_L)$ denotes a sequence of length L , with each position a_i taking values in an alphabet of size $q = 21$ (the 20 natural amino acids plus the gap symbol), and Z is the partition function enforcing normalization. The inverse temperature β is set equal to one, implying its dependence inside the parameters of the Hamiltonian. The latter is defined as

$$H(\mathbf{a}) = - \sum_{i < j} J_{ij}(a_i, a_j) - \sum_i h_i(a_i), \quad (4.2)$$

where $J \in \mathbb{R}^{L \times L \times q \times q}$ is the coupling tensor and $h \in \mathbb{R}^{L \times q}$ are local fields. As a modeling choice, motivated by the empirical focus on contact prediction, we omit explicit local fields and concentrate on the coevolutionary signal encoded in pairwise interactions. In a fully general Potts model, local fields and couplings are not uniquely defined, and part of the single-site contribution can be redistributed into the pairwise tensor by exploiting gauge freedom. In the factored attention model,

however, the pairwise tensor is constrained by the attention parametrization, so this redistribution is not guaranteed to preserve the same functional form. The absence of explicit fields should therefore be understood as a simplifying assumption of the model, rather than as a completely general gauge choice.

In standard DCA, the interaction tensor encodes both positional and amino acid information, capturing the evolutionary constraints of a protein family. After inference, the amino-acid degrees of freedom are integrated out through the Frobenius norm of the interaction tensor, leaving purely positional information that acts as a proxy for the direct coevolutionary signal and for the physical contacts in the three-dimensional structure. Experimentally, the number of relevant contacts scales linearly with the length of the sequence, whereas the interaction tensor grows as $\mathcal{O}(L^2)$. This suggests that the effective coevolutionary signal may admit a low-rank representation. The factored attention mechanism introduced by Bhattacharya *et al.* provides a natural realization of such a decomposition. In addition, the separation between positional and amino-acid degrees of freedom allows for a more direct proxy for contact prediction, as we describe below.

To implement the factored attention, the coupling tensor is parameterized as

$$J_{ij}(a,b) = \sum_{h=1}^H \text{softmax}\left(Q^h K^{hT}\right)_{ij} V_{ab}^h, \quad (4.3)$$

where $Q^h, K^h \in \mathbb{R}^{L \times d}$ are the *query* and *key* matrices, and $V^h \in \mathbb{R}^{q \times q}$ is the *value* matrix, in one of the H attention heads that collectively build the tensor J . Here Q^h and K^h encode positional degrees of freedom that are family-specific, while V^h captures amino acid interaction features. The softmax operation is performed row-wise, yielding the attention matrix

$$A^h = \text{softmax}\left(Q^h K^{hT}\right), \quad (4.4)$$

whose entries represent the self-attention between residues along the sequence, and satisfy $\sum_j A_{ij}^h = 1$ for each position i . The form of Equation 4.3 can be interpreted as a low-rank factorization of the Potts coupling tensor, where attention heads define a structured decomposition of positional and amino-acid interactions. This mirrors the general multi-head self-attention mechanism of Transformer architectures, and in

particular the column-attention layer of the Evoformer block in AlphaFold, which explicitly models coevolutionary constraints in an MSA.

Inference of the parameters is carried out by maximizing the pseudo-likelihood of the model, equivalent to a Masked Language Modeling scheme in NLP (see Sections 2.4.5 and 3.3.2). For an MSA $D = \{\mathbf{a}^m\}_{m=1}^M$ of depth M , with a set of reweighting coefficients $\{w^m\}_{m=1}^M$ such that $\sum_m w^m = M_{\text{eff}}$ the pseudo-likelihood reads

$$\begin{aligned}\mathcal{L}(J|D) &= \frac{1}{M_{\text{eff}}} \sum_{m=1}^M \sum_{i=1}^L w^m \log P(a_i^m | \mathbf{a}_{\setminus i}^m, J) \\ &= \frac{1}{M_{\text{eff}}} \sum_{m=1}^M \sum_{i=1}^L w^m \left\{ \sum_{j \neq i} J_{ij}(a_i^m, a_j^m) - \log \left[\sum_{a=1}^q \exp \left(\sum_{j \neq i} J_{ij}(a, a_j^m) \right) \right] \right\} .\end{aligned}\quad (4.5)$$

To prevent overfitting and aid the optimization, we add an L2 regularization term,

$$\tilde{\mathcal{L}}(J|D) = \mathcal{L}(J|D) - \lambda \sum_{i,j,a,b} J_{ij}^2(a,b). \quad (4.6)$$

The nonlinear decomposition of the interaction tensor resulting from Equation 4.3 makes the optimization landscape highly non-convex. Because of this, the maximization procedure is therefore much more challenging than that of PlmDCA or arDCA. We use stochastic gradient ascent with an ADAM optimizer on mini-batches of fixed size in order to better navigate the likelihood landscape.

Once the parameters have been inferred, residue-residue contacts are extracted from the model. In standard DCA methods, the interaction strength between sites i and j is quantified by the Frobenius norm of the symmetrized coupling tensor

$$F_{ij} = \sqrt{\sum_{a,b} \tilde{J}_{ij}^2(a,b)} , \quad \tilde{J}_{ij}(a,b) = \frac{1}{2} (J_{ij}(a,b) + J_{ji}(b,a)) \quad (4.7)$$

To correct for spurious correlations due to background effects, an Average Product Correction (APC) is applied to the Frobenius scores. Ranking all site pairs (i, j) by their corrected F_{ij} produces a list of predicted contacts. This contact extraction scheme was used in Bhattacharya et al. [95] to compare a factored attention layer with Direct Coupling Analysis. However, in the context of AttentionDCA, an alternative scoring scheme can be defined directly from the attention matrices. By

averaging the attention matrices over the H heads and symmetrizing,

$$A_{ij} = \sum_{h=1}^H \text{symm}(A_{ij}^h) \quad , \quad (4.8)$$

we obtain the attention score (AS), which can be used in place of the Frobenius score (FS). This definition is both simpler and more directly connected to the underlying mechanism of the model, as it bypasses the construction of the full $\mathbb{R}^{L,L,q,q}$ coupling tensor. In Section 4.2.2, we compare the predictive accuracy of these two approaches.

The quality of contact prediction is assessed against experimental structures deposited in the Protein Data Bank (PDB). Following established practice, two residues are considered in contact if the distance between any heavy atoms is less than 8 Å, provided that $|i - j| \geq 6$ so as to exclude trivial local contacts along the sequence. The accuracy of the predicted contacts is quantified by the Positive Predictive Value (PPV), defined as the fraction of true contacts among the top n predictions: $\text{PPV}(n) = \text{TP}(n)/n$. Throughout this thesis, we use the standard notation $\text{PPV}@n$ to denote $\text{PPV}(n)$.

Due to the non-convex nature of the optimization and the use of stochastic gradient ascent, individual inference runs produce slightly different coupling tensors and attention matrices. This variability introduces noise in the contact predictions. To mitigate this effect, we aggregate multiple runs. Specifically, for Frobenius scores, we define the final contact score for a pair (i, j) as the maximum score obtained across all realizations. Averaging over roughly twenty runs is sufficient to obtain robust and reproducible predictions. As shown in Section 4.2.2, this procedure yields stable PPV curves and reveals a consistent set of core contacts that are reliably predicted across runs.

4.1.2 Multi-family version

A long-standing goal in protein modeling is that of designing inference schemes that can capture signals from multiple protein families simultaneously, in order to generate sequences that reproduce selected structural and functional traits. In the context of shallow statistical inference techniques, a first step in this direction would be introducing a DCA model capable of multi-family learning. However, standard DCA formulations are not suited to this task. Since both the coupling tensor J and

the local fields h depend explicitly on the family, there is no natural way to identify a common set of parameters across unrelated multiple sequence alignments of different lengths. Each protein family must therefore be analyzed independently, even though many of the biochemical constraints shaping amino acid interactions can be thought of as universal, only depending on the underlying physicochemical interactions.

The factored attention implementation of DCA offers a natural solution. In particular, in Eq. 4.3 the value matrix $V \in \mathbb{R}^{q,q}$ does not depend on the length of the family and, in principle, it can be shared across different families. In Bhattacharya et al. it is shown that a set of value matrices learned in a protein family can be *frozen* and used during the learning of Query and Key matrices for another protein family, resulting in comparable accuracy in tasks such as contact prediction. Expanding their work, we introduce a novel multi-family model in which the value matrices are directly learned across multiple families simultaneously.

Formally, given N_F protein families, each corresponding to an MSA $D_f \in \mathbb{R}^{L_f, M_f}$, we define a multi-family pseudo-likelihood as the sum of the single-family likelihoods:

$$\mathcal{L}_{\text{MF}} = \sum_{f=1}^{N_F} \mathcal{L}_f(\{Q_f^h, K_f^h, V^h\}, D_f), \quad (4.9)$$

where each single-family likelihood is given by Equation 4.6. Here, the set of parameters $\{Q_f^h, K_f^h\}$ is family-dependent and can be used to extract the contact score for each MSA, while $\{V^h\}$ are shared across all families and act as the parameters coupling together each single-family pseudo-likelihoods. To allow parameter sharing, each family must use the same number of attention heads H , so that the dimensions of $\{V^h\}$ are compatible across the dataset. Inference is performed by maximizing Eq. 4.9 using the same stochastic optimization scheme as in the standard case, with ADAM updates applied simultaneously to the shared $\{V^h\}$ and to the family-specific $\{Q_f^h, K_f^h\}$.

The advantage of the multi-family setup is that the V^h matrices are learned from a broader set of MSAs, thereby capturing amino acid interaction patterns that are not tied to a single family. In principle, this allows families with limited sequence depth to benefit from information learned from larger or related families, while still retaining family-specific structural signals in their Q^h and K^h matrices.

4.1.3 Autoregressive version

As discussed in Chapter 2, the autoregressive formulation of DCA (arDCA) was introduced to overcome one of the main limitations of inference through pseudo-likelihood, namely the lack of direct access to the full probability distribution and the consequent difficulties in using the model generatively. In arDCA this issue is addressed by the exact factorization of the full probability $P(\mathbf{a})$ into a product of conditionals, enabling both efficient evaluation and straightforward sampling of novel sequences.

A similar autoregressive strategy can be adopted within the AttentionDCA framework, extending the implementation described in Section 4.1.1 into a generative version. The starting point is the autoregressive decomposition

$$P(\mathbf{a}) = P(a_1) \prod_{i=2}^L P(a_i | a_1, \dots, a_{i-1}), \quad (4.10)$$

which expresses the sequence probability as a product of single-site conditionals. To enforce this causal structure in the Hamiltonian in Equation 4.2, we apply a lower-triangular mask to the interaction tensor, such that a position i can only interact with positions $j < i$. Formally,

$$J_{ij}(a, b) = \begin{cases} \sum_{h=1}^H \text{softmax}(Q^h K^{hT})_{ij} V_{ab}^h, & i > j, \\ 0, & i \leq j. \end{cases} \quad (4.11)$$

This modification makes sequential sampling from left to right possible, in close analogy with the masked self-attention mechanism introduced in Chapter 3 in the context of the Transformer architecture.

In section 2.4.6, we mentioned that a direct consequence of the conditional factorization in Equation 4.10 is that the inferred couplings no longer represent symmetric direct interactions, as in standard DCA. Instead, they encode the dependencies between a site and those that come before it along the amino-acid sequence. For this reason, contact prediction cannot rely on the Frobenius score of the coupling tensor. Instead, we follow the approach introduced in Trinquier *et al.* and define the epistatic score (ES), which quantifies the non-additive contribution of simultaneous mutations in two different sites. For positions i, j in a wild-type sequence $\mathbf{a}$, mutating $a_i \rightarrow b_i$

and $a_j \rightarrow b_j$ yields

$$\Delta\Delta E_{ij}(b_i, b_j) = \Delta E(a_i \rightarrow b_i, a_j \rightarrow b_j) - \Delta E(a_i \rightarrow b_i) - \Delta E(a_j \rightarrow b_j) \quad , \quad (4.12)$$

where ΔE denotes the energy change associated with a mutation:

$$\begin{aligned} \Delta E(a_i \rightarrow b_i) &= E(a_1, \dots, b_i, \dots, a_L) - E(a_1, \dots, a_i, \dots, a_L) = \\ &= -\log \frac{P(a_1, \dots, b_i, \dots, a_L)}{P(a_1, \dots, a_i, \dots, a_L)} \end{aligned} \quad (4.13)$$

The resulting quantity $\Delta\Delta E_{ij}$ replaces J_{ij} in the construction of the epistatic score by means of its average-product-corrected Frobenius norm, thus providing a contact score for the autoregressive model. Importantly, this definition is robust with respect to the choice of the reference sequence, as already observed for arDCA.

Beyond contact prediction, the autoregressive formulation turns AttentionDCA into a fully generative model. Sampling proceeds sequentially from Eq. 4.10, drawing one residue at a time from the learned univariate conditional distributions $P(a_i | \mathbf{a}_{<i})$. The resulting artificial MSAs can be directly compared to natural ones by examining their statistical properties, such as single-site frequencies, two-site connected correlations, or principal component projections. These benchmarks, already introduced in Chapter 2 for arDCA, will be revisited in Section 4.2.4 to assess the generative performance of the autoregressive AttentionDCA.

4.2 Results

AttentionDCA, in its standard, multi-family, and autoregressive versions introduced in the previous section, was evaluated on a set of protein families whose structural data is available in the Protein Data Bank (PDB). Each family is represented by a multiple sequence alignment (MSA) of variable length and depth, spanning a wide range of effective sequence diversity. A summary of the datasets along with the results of various AttentionDCA implementations can be found in Table 4.1.

The analysis is organized around two main tasks. First, we assess the accuracy of contact prediction, comparing the Frobenius norm, the Attention Score, and the Epistatic Score introduced for the autoregressive version. Second, we investigate the generative capabilities of the autoregressive formulation by comparing the statistical

properties of the generated MSAs to those of their natural counterparts. Both tasks are evaluated against established benchmarks in literature: experimental contact maps from crystallographic structures for the first, frequency counts and principal component statistics for the second.

The results are presented in four parts. We begin with an analysis of the parameter space and the role of hyperparameters, emphasizing the relation between compression ratio and contact prediction performance, in Section 4.2.1. Then, we compare the contact prediction accuracy of AttentionDCA with standard DCA methods, focusing on both the Frobenius and Attention scores, in Section 4.2.2. The structural signal encoded in attention heads is analyzed in Section 4.2.3, highlighting the sparsity of the information carried by individual heads. Finally, in Section 4.2.4 we evaluate the generative performance of the autoregressive version, assessing both contact prediction via the Epistatic Score and the statistical similarity between natural and artificial MSAs.

4.2.1 Parameter reduction and hyper-parameters

The training of AttentionDCA requires setting both learning- and model-related hyperparameters. The former include the learning rate, the mini-batch size, and the number of training epochs. In our implementation, we adopt a learning rate of $\eta = 5 \times 10^{-3}$ and a batch size of $n_b = 1000$. The number of epochs depends on the depth of the MSA, as reported in Table 4.1. These values were chosen empirically to ensure stable optimization across protein families.

More relevant to the model’s expressive power are the structural hyperparameters that define the low-rank factorization of the interaction tensor, namely the number of attention heads H and the inner dimension d of the query and key matrices in Equation 4.3. In principle, both H and d can be tuned independently, but their effect is better understood in terms of the total number of parameters they define. For AttentionDCA, this is given by

$$N_{\text{AttentionDCA}} = 2HLd + Hq^2 \quad , \quad (4.14)$$

where $q = 21$ is the alphabet size. For comparison, the number of parameters in PlmDCA is

$$N_{\text{PlmDCA}} = \frac{L(L-1)}{2}q^2 + Lq \quad , \quad (4.15)$$

which scales quadratically with sequence length L . We can define a compression ratio relative to PlmDCA as a convenient measure of parameter reduction.

$$c_r = \frac{N_{\text{AttentionDCA}}}{N_{\text{PlmDCA}}} = \frac{2HdL + Hq^2}{L(L-1)q^2/2 + Lq} \quad , \quad (4.16)$$

We observe that contact prediction accuracy depends primarily on the compression ratio rather than on the individual values of H and d . At fixed c_r , the performance of the model remains essentially constant across different choices of H and d , provided H is sufficiently large to stabilize training. Figure 4.1 illustrates this behavior for three representative families (PF00014, PF00076, PF00763), where $\text{PPV}@L$, $\text{PPV}@3L/2$, and $\text{PPV}@2L$ remain stable across different (H, d) combinations yielding the same compression ratio. This indicates that the number of effective parameters, rather than their particular arrangement, governs the quality of inference.

At very small numbers of heads ($H \lesssim 10$), the performance deteriorates, as optimization tends to stall at suboptimal stationary points. Conversely, increasing the total number of parameters indefinitely does not lead to progressively better predictions: when AttentionDCA approaches the parameter count of PlmDCA ($c_r \rightarrow 1$), its accuracy decreases. Likely, this is a consequence of the non-convex nature of the likelihood landscape, which makes the optimization increasingly difficult as the number of parameters grows. Overall, we identify an optimal compression regime in the range $0.05 \lesssim c_r \lesssim 0.20$, where AttentionDCA achieves stable and accurate contact prediction with a substantial reduction in parameter count.

4.2.2 Standard and multi-family contact prediction

In standard DCA, the consolidated method for contact prediction is to compute the Frobenius norm of the inferred and symmetrized interaction tensor. To improve the accuracy of the prediction, the parameters are initially mapped into a zero-sum gauge, which minimizes the Frobenius norm, and successively average-product corrected, so as to smooth out global biases in the MSA. Pairs of residues (i, j) are then ranked based on increasing Frobenius score (FS), resulting in a list of

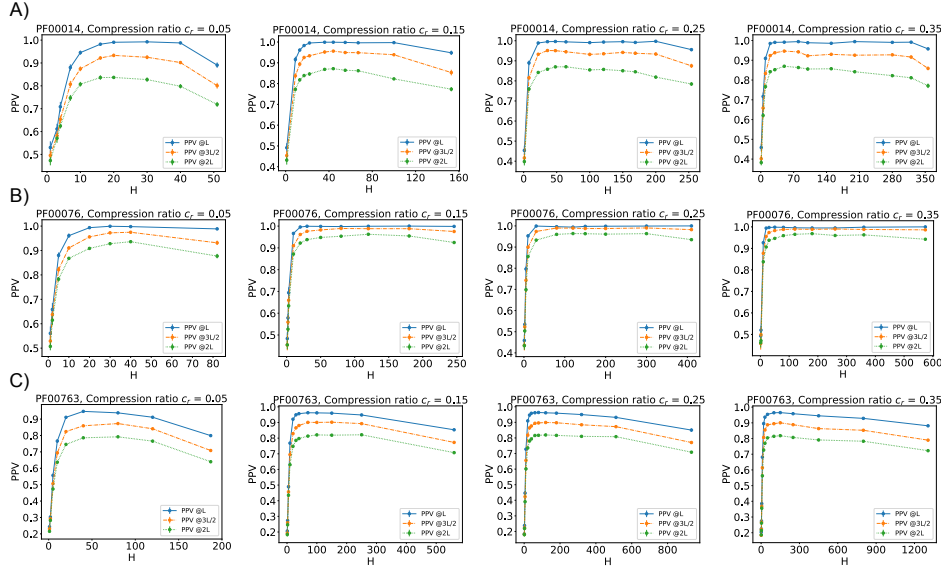


Fig. 4.1 PPV@L, 3L/2 and 2L (blue, orange and green curve respectively) for different values of the compression ratio r for families PF00014, PF00076 and PF00763 respectively in panels A), B) and C). In each plot, the compression ratio is held constant while the Positive Predicted Value is calculated across different inference runs with varying numbers of heads H . Since the compression ratio is fixed, for each value of H there is a corresponding value of d given by inverting Equation 4.16 at fixed c_r . Source [42]

increasingly likely contacts. In order to evaluate the accuracy of the prediction, predicted contacts are compared to experimentally known structures of the protein family under study, producing a Positive Predicted Values (PPV) as a function of the number of predictions. Alternatively to this classic way of doing contact prediction, the factored attention implementation of Equation 4.3 suggests another way of determining the contacts of a protein family by averaging the positional-dependent attention heads and symmetrizing the result, as shown in Equation 4.8. After the average-product correction, the attention score (AS) thus defined can be used as a contact score and compared to the standard Frobenius score.

As mentioned in Section 4.1.1, the non-convexity of the optimization landscape produces an intrinsic noise during learning, which can be smoothed out simply through different realizations of the same inference. In particular, in the case of the FS contact prediction, for each position pair we defined a merged score as the maximum score through all realizations as in $F_{ij} = \max\{F_{ij}^1, F_{ij}^2, \dots, F_{ij}^m\}$. In the case of the AS contact prediction, the average of the attention matrices from

each realization is sufficient to produce a robust score. Varying the number of the realizations m , it can be seen that a threshold of $m = 20$ is enough to return a consistent contact prediction.

Figure 4.2 compares the PPV curves obtained with AttentionDCA to those from PlmDCA (black curve), taken as a benchmark, for three representative families (PF00014, PF00072, PF00763). Gray curves correspond to an ideal predictor that identifies all and only true contacts, while the red and blue curves represent the FS PPV from single and merged realizations of AttentionDCA, respectively. The compression ratio c_r between AttentionDCA and PlmDCA was fixed at 0.2. For all families, the agreement with PlmDCA is strong, with only minor fluctuations visible in the single-run curves. These results must be understood in light of the considerable parameter reduction: while PlmDCA is a state-of-the-art method for contact prediction among shallow statistical models, reproducing its accuracy with only 20% of the parameters is a significant result in itself, independently of the additional advantages that will be discussed below.

Concerning the multi-family learning, its results are consistent with those obtained in the single-family setting. This indicates that the shared Value matrices capture essential biochemical signals across different families without compromising the accuracy of family-specific contact prediction. Complemented by the analysis of Bhattacharya *et al.*, this supports the view that amino-acid interaction patterns can be considered largely universal and shared across different domains, whereas the family-specific features are encoded separately in Query and Key matrices.

This setup not only preserves the accuracy of single-family models, but also highlights the flexibility of AttentionDCA in integrating diverse protein family data, offering an efficient alternative for multi-family protein analysis. The dark and light green curves in Fig. 4.2 represent the PPVs obtained from multi-family training, performed in parallel across all families using a fixed number of heads $H = 128$ and adjusting the inner dimension d to match the 20% compression ratio used in the single-family simulations. The shown curves are obtained through the Frobenius score computed on the J_{ij} tensor which is given by Equation 4.3 and parameters $\{Q_f^h, K_f^h, V^h\}_{h=1, \dots, H}$ for $f = \{\text{PF00014}, \text{PF00072}, \text{PF00763}\}$.

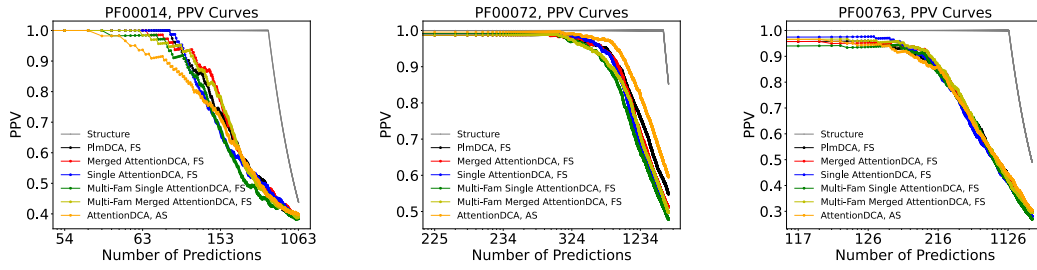


Fig. 4.2 Positive Predicted Value Curves for families PF00014, PF00072, PF00763 respectively. Each plot compares PPVs computed through the Frobenius Score (FS) and the Attention Score (AS), in particular: PlmDCA with FS (black curve); merged- and single-realization of AttentionDCA with FS (red and blue curves respectively); Multi-Family AttentionDCA with merged- and single-realization (green and yellow curves respectively); AttentionDCA with AS (orange curve). The gray curve represents the PPV of an ideal model that predicts all and only positive contacts, hence it depicts the prediction of the full structure of the family. Source [42]

4.2.3 Attention heads

In Chapter 3, we discussed the rise of the attention mechanism in Natural Language Processing as a tool to learn functional relationships between elements of a dataset and to produce encoded representations. In Protein Language Modeling, deep learning architectures such as MSA Transformer or AlphaFold exploit variants of the attention mechanism to disentangle the physical and evolutionary constraints among homologous sequences. The formal analogy between factored self-attention and classical Potts models, described in Section 3.3.2, and the results shown in Fig. 4.2 suggest that the attention heads inside the interaction tensor J_{ij} should capture structural information that can be compared to the contact map obtained from the average-product corrected Frobenius norm of the full interaction tensor.

By analogy with NLP, one might expect each attention head to specialize in a particular secondary or tertiary structure, much like Transformer heads specialize in specific semantic relations in a sentence. In practice, however, this expectation is difficult to verify in the case of proteins. The lack of an obvious one-to-one correspondence between the variety of possible structural motifs and the number of heads, and the fact that heads derived from Eq. 4.3 are intrinsically asymmetric, complicates their direct interpretation. Rather than analyzing each head separately *a*

posteriori, it is therefore more practical to average all heads into a single attention matrix, as in Eq. 4.8.

The averaged attention matrix can be used to extract structural information either in the form of the Attention Score (AS) or by direct visual comparison with the experimental contact map of the family under study. Figure 4.3 illustrates this comparison for families PF00014, PF00072, and PF00763. In the contact maps, blue and red dots represent positive and negative predictions, while gray dots show the actual crystallographic structure. Conversely, the heat maps of the averaged attention matrices are displayed in grayscale, with the experimental contacts indicated in red. The maxima of the attention matrix (darker dots) align with the known contacts, demonstrating that AttentionDCA effectively focuses on the structural determinants of the protein family. Although the specialization of individual attention heads is difficult to interpret, an important observable property is the sparsification of structural information within each head. In practice, only a small fraction of the largest entries in each attention matrix is sufficient to recover contact prediction accuracy comparable to that obtained from the full matrices. This is illustrated in panel C of Figure 4.3, which shows a sparse attention matrix constructed by retaining only the $k = L$ highest elements from each head. The resulting heatmap still highlights the relevant structural contacts, indicating that the essential information captured by AttentionDCA is concentrated in a sparse subset of attention weights.

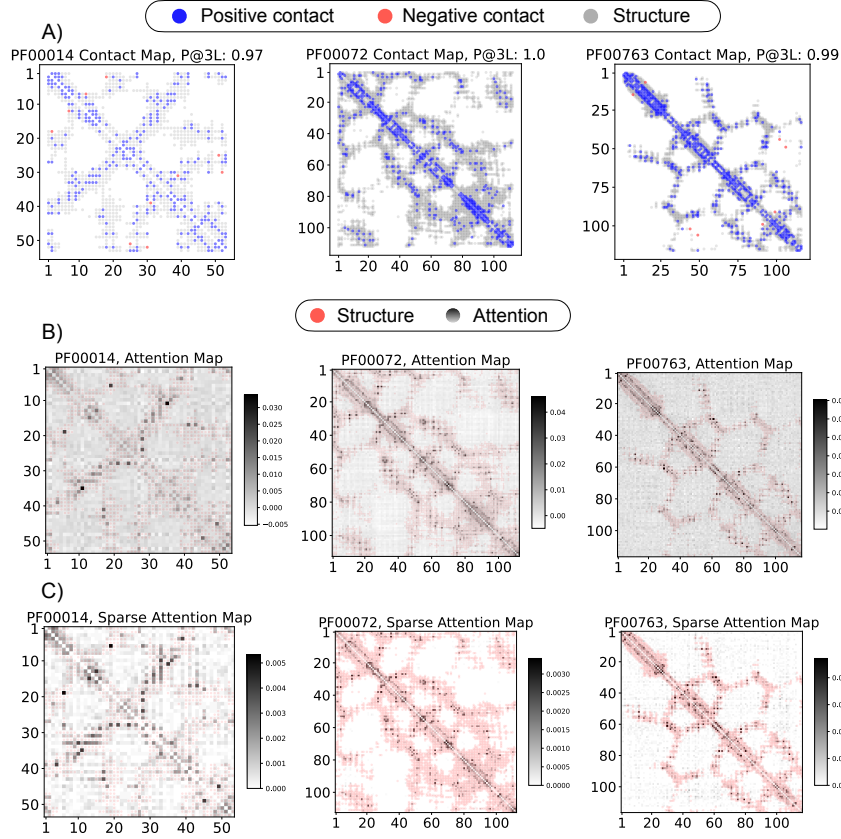


Fig. 4.3 Various contact maps for families PF00014, PF00072, PF00763. A) Contact map built from the Frobenius Score applied to the interaction tensor inferred by AttentionDCA. Blue and red dots represent positive and negative predictions respectively, while gray dots reproduce the structure of the family. B) Grayscale heat map of the averaged attention matrix built using Equation 4.8, from the $\{Q^h, K^h\}_{h=1, \dots, H}$ matrices by AttentionDCA. C) Grayscale heat map for the sparse attention matrix constructed by maximizing over the k highest elements from each head, using $k = L$, cf. Supplementary Material Sec. G. Both in B) and C) structural information is depicted in red. Source [42]

4.2.4 Generativity

The autoregressive formulation of AttentionDCA introduced in Section 4.1.3 enables the model to be used as a generative architecture, capable of sampling artificial MSAs that reproduce the statistical properties of natural protein families. In this section, we assess its performance both in terms of contact prediction, using the

epistatic score defined in Eq. 4.12, and in terms of generativity, by comparing the statistics of natural and generated alignments.

Panel A of Figure 4.4 shows PPV curves for three representative families (PF00014, PF00072, PF00763). The black and red curves correspond to PlmDCA (FS) and arDCA (ES), which serve as benchmarks, while the blue and green curves show the single- and merged-run implementations of autoregressive AttentionDCA with the epistatic score. The yellow curve represents an alternative strategy in which artificial MSAs generated by AttentionDCA are fed back into a non-generative model (PlmDCA or standard AttentionDCA) to compute a Frobenius Score. This indirect approach yields among the most accurate predictions within the generative framework, and also serves as a test of the model’s sampling quality.

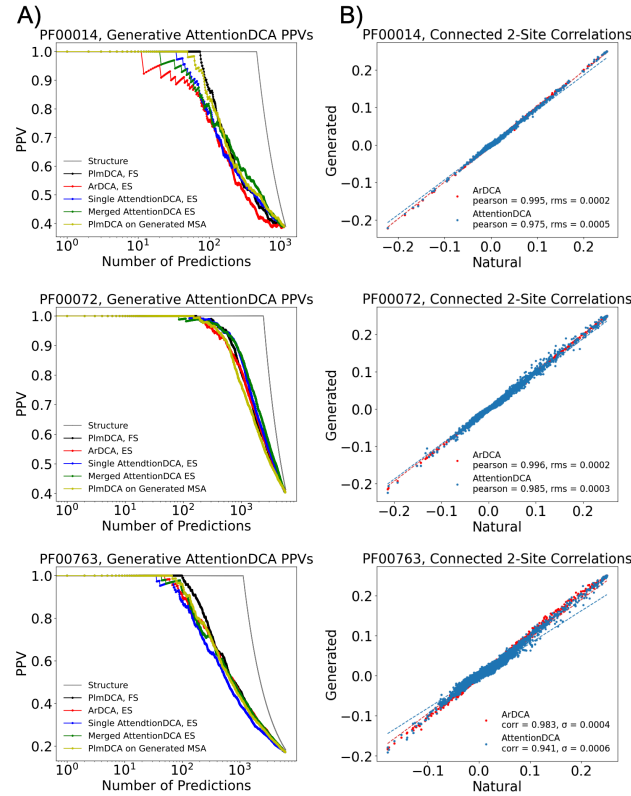


Fig. 4.4 Results from Autoregressive AttentionDCA applied to families PF00014, PF00072 and PF00763. For each family, the panels show the PPV curves and the connected two-site correlations. A) The plots on the left compare the PPV curves from PlmDCA (FS, black curve), arDCA (ES, red curve), with the single- and merged-realizations of AttentionDCA with ES (blue and green curves respectively) and the PPV from the FS of PlmDCA applied to a Generated MSA sampled from Autoregressive AttentionDCA. The gray curve represents the PPV of an ideal model that predicts all and only positive contacts, hence it depicts the prediction of the full structure of the family. B) The plots on the right show a comparison between the two-site connected correlations of the natural and artificial MSAs generated by Autoregressive AttentionDCA and arDCA (blue and red dots respectively). Source [42]

The generative capabilities of autoregressive AttentionDCA are further evaluated by comparing summary statistics between natural and artificial MSAs. Panel B in Figure 4.4 shows the two-site connected correlations, with a Pearson correlation coefficient close to one, confirming that the model reproduces pairwise statistics with high accuracy. A complementary perspective is provided by Principal Compo-

ment Analysis (PCA): Figure 4.5 compares the leading components of natural and generated alignments, showing a close overlap for both arDCA and AttentionDCA.

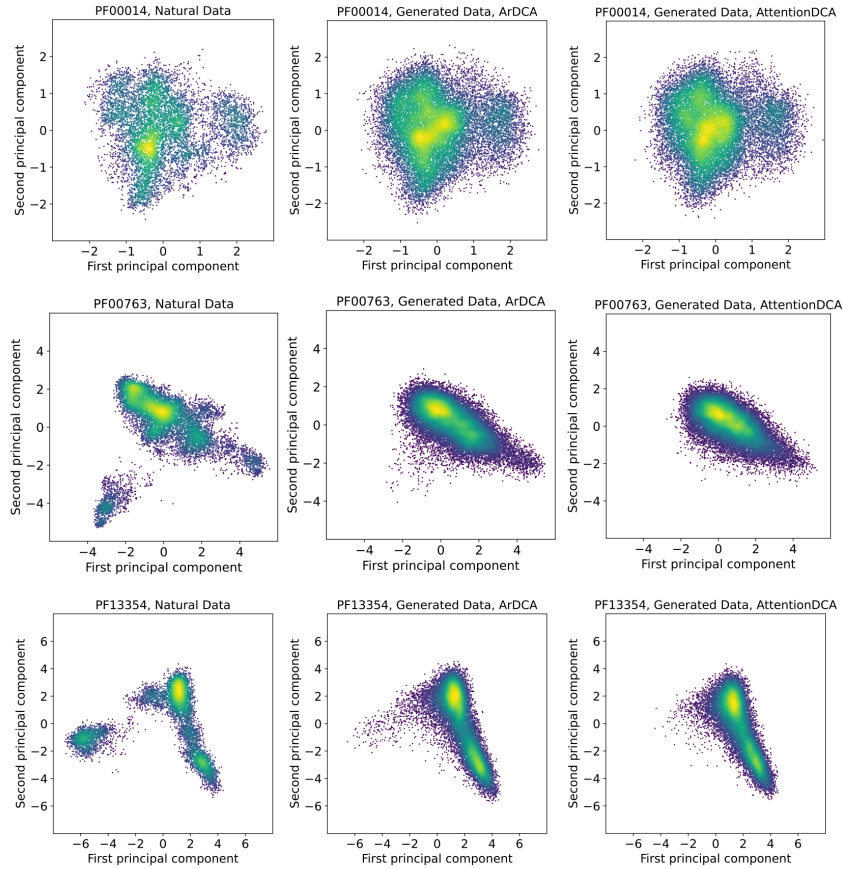


Fig. 4.5 Principal Component Analysis on protein families PF00014, PF00763 and PF13354. Comparison between natural data and artificial data generated by using arDCA and AttentionDCA. Source [42]

The analyses of correlations and principal components confirm that, provided enough data are available, the autoregressive AttentionDCA is capable of generating realistic protein-like sequences. However, for many of the families listed in Table 4.1, the contact prediction obtained from the epistatic score falls significantly below that of the Frobenius score from the standard version or of PlmDCA. This discrepancy can be traced back to the fact that autoregressive inference requires a sufficiently large number of independent sequences. In general, when the effective depth M_{eff} of an MSA is reduced below a certain threshold, the statistical signal becomes too weak and the contact prediction accuracy of DCA models decreases. The exact

threshold is model dependent, with different DCA variants displaying different levels of robustness to M_{eff} degradation.

Figure 4.6 illustrates how contact prediction accuracy decays as a function of M_{eff} for families of different lengths. PlmDCA shows the highest robustness, maintaining reliable predictions even for $M_{\text{eff}} < 1000$. By contrast, the autoregressive version of AttentionDCA deteriorates rapidly when $M_{\text{eff}} \lesssim 2000$. As a result, its accuracy on shallow families such as PF00595, PF13354, or PF00035 is intrinsically worse than that of PlmDCA or of the standard AttentionDCA. This observation highlights a key limitation of the generative extension: while it offers interpretability and the ability to sample artificial sequences, its applicability is restricted to sufficiently deep MSAs.

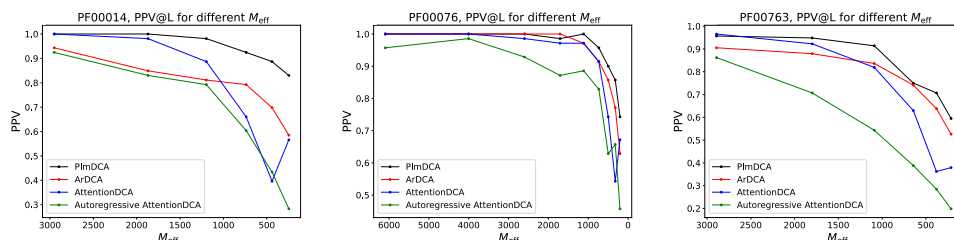


Fig. 4.6 **PPV@L for different M_{eff}** Computed using PlmDCA (FS), arDCA (ES), AttentionDCA (FS) and Autoregressive AttentionDCA (ES). Families PF00014, PF00076, and PF00763 are shown. Source [42]

4.3 Conclusion and Perspectives

In this chapter, we introduced AttentionDCA, our in-house DCA implementation of factored attention, and explored its performance across a variety of protein families. The model was analyzed in its standard formulation, in a multi-family generalization where the value matrices are shared across alignments, and in an autoregressive extension that enables generative sampling. The results obtained from these three implementations provide a comprehensive picture of the strengths and limitations of attention-based statistical models for protein sequences.

The first key finding is that the standard version of AttentionDCA achieves contact prediction accuracies comparable to PlmDCA, despite using only about 5 – 20% of its parameters. The Frobenius and Attention Scores produce similar results, with the latter offering a more direct and interpretable link between the attention

mechanism and structural contacts. The multi-family version extends this framework further by demonstrating that value matrices can be shared across unrelated families without loss of accuracy. This observation supports the interpretation of the V matrices as encoding biochemical signals that are largely universal, while the Q and K matrices remain responsible for capturing the family-specific evolutionary constraints.

The autoregressive version of AttentionDCA provides a bridge to generative modeling. By enforcing a causal masking of the couplings, the model can sample novel sequences and reproduce the main statistical features of natural alignments, including amino acid frequencies, connected correlations, and principal component distributions. Contact prediction in this framework is based on the epistatic score, which, although less robust than the Frobenius or Attention Scores, provides a meaningful way to extract pairwise interactions from an autoregressive model. The comparison with arDCA confirms that attention-based architectures can be successfully adapted as a generative architecture.

Alongside these strengths, the analysis has also revealed important limitations. Optimization in AttentionDCA is non-trivial due to the non-convexity of the likelihood, requiring multiple runs and aggregation strategies to obtain stable predictions. Moreover, the autoregressive formulation shows a strong dependence on the effective depth of the MSA. While PlmDCA retains reasonable accuracy for alignments with $M_{\text{eff}} < 1000$, the autoregressive AttentionDCA deteriorates rapidly when $M_{\text{eff}} \lesssim 2000$, resulting in poor performance on shallow families. This sensitivity restricts the applicability of the generative extension to protein families with sufficiently deep sequence coverage.

Taken together, these results highlight the strength and the challenges of attention-inspired statistical models for proteins. On the one hand, they offer parameter efficiency, interpretable attention maps that can be directly related to structural contacts, and the possibility of sharing interaction priors across families. On the other hand, they face optimization difficulties and exhibit limitations in low-data regimes. These trade-offs suggest natural directions for future work. A first perspective is to better understand and interpret the universal information contained in the value matrices, relating it to the biochemical properties of amino acids. A second is to enrich the architecture with additional layers that mimic the logic of AlphaFold and Transformer-based architectures while remaining fully interpretable. An example

of this would be including embedding layers before the factored attention module. In this sense, AttentionDCA should be seen not as a step towards larger black-box models, but rather as a methodological contribution that extends the statistical physics tradition of protein modeling while retaining transparency and interpretability at its core.

Table 4.1 **Summary of the protein families used throughout the analysis and main results.** Name, Description, Length, Depth, Effective Depth of the Protein Family. Number of parameters for PlmDCA and AttentionDCA. Two-site connected correlation Pearson coefficient ρ for arDCA and AttentionDCA. FS Positive Predicted Values at L for PlmDCA and AttentionDCA. Source [42]

Name	Description	L	M	M_{eff}	N_{PlmDCA}	$N_{\text{AttentionDCA}}$	ρ_{arDCA}	$\rho_{\text{AttentionDCA}}$	PPV@L PlmDCA	PPV@L AttentionDCA
PF00014	Kunitz Domain	53	8871	2950	608,811	124,288	0.995	0.975	1.0	1.0
PF00035	DSRM	67	18462	2176	976,458	193,664	0.986	0.963	0.880	0.895
PF00072	RR	112	574565	59290	2,743,608	543,872	0.996	0.985	1.0	1.0
PF00076	RRM	70	79366	6093	1,066,485	217,728	0.997	0.986	1.0	0.985
PF00169	PH Domain	105	55196	4115	2,410,065	486,528	0.985	0.958	0.990	0.990
PF00595	PDZ Domain	82	15299	1489	1,466,283	287,360	0.973	0.846	0.939	0.878
PF00677	Lumazine BD	87	16192	2853	1,651,608	323,712	0.994	0.982	0.908	0.896
PF00763	Catalytic Domain	116	10886	2892	2,943,906	590,976	0.983	0.941	0.956	0.956
PF13354	Beta-Lactamase	202	7515	1559	8,956,983	1,814,656	0.942	0.910	0.965	0.935

Chapter 5

Applications and Extensions of Attention-Based Direct Coupling Analysis

In this chapter, we explore possible applications of AttentionDCA beyond its use as an interpretability tool. In particular, we focus on how the structure of its parametrization, and the resulting flexibility with respect to standard DCA models, can be exploited in settings where different sources of information can be decoupled, such as in protein-protein interaction analysis or paralog matching.

In the second part of the chapter, we consider a concrete example in which AttentionDCA is used as an interaction score within a paralog matching framework, following recent collaborative work with the Laboratory of Statistical Biophysics at EPFL [96].

5.1 Decoupling and Flexibility in AttentionDCA

In Chapter 4, AttentionDCA has been introduced and analyzed as an interpretable parametrization of the attention mechanism in terms of Direct Coupling Analysis, following previous theoretical and experimental work [73, 95]. Beyond its interpretability, the structure of the parametrization induces a form of decoupling that

provides additional flexibility from a modeling perspective, enabling a range of extensions and applications.

As discussed previously, the DCA coupling tensor can be expressed as a sum over attention heads,

$$J_{ij}(a, b) = \sum_{h=1}^H J_{ij}^{(h)}(a, b) = \sum_{h=1}^H A_{ij}^{(h)} V_{ab}^{(h)}, \quad (5.1)$$

where $A_{ij}^{(h)}$ encodes positional dependencies and $V_{ab}^{(h)}$ describes amino-acid interactions. This factorization separates positional and compositional contributions, inducing a structured decoupling of the interaction tensor. In contrast to standard Potts models [23], amino-acid interaction patterns are shared across positions, while positional specificity is encoded independently through the attention maps, resulting in a low-rank and structured parametrization of the couplings [73, 83, 95].

A further level of structure is provided by the decomposition over attention heads, defining additive components J_{ij}^h to the full interaction tensor, similarly to what has been explored in Hopfield-Potts models [97, 98]. Although this decomposition does not, in general, lead to an automatic specialization of different heads, it provides a natural set of latent channels through which different interaction patterns can be represented and, if needed, controlled.

This structure becomes particularly relevant in systems where the interaction tensor itself can be decomposed into distinct contributions. Consider a sequence composed of two interacting segments, denoted α and β , for which the interaction tensor can be written as

$$J = J^{\alpha\alpha} + J^{\beta\beta} + J^{\alpha\beta} + J^{\beta\alpha}, \quad (5.2)$$

corresponding to intra- and inter-segment interactions. While this decomposition is not explicitly enforced in the standard AttentionDCA formulation, the head-based parametrization makes it possible to align different components of the model with these different interaction terms, for instance by associating subsets of heads with specific parts of the interaction tensor.

5.1.1 Subdomain Decomposition and Controlled Decoupling

A concrete realization of controlled decoupling in AttentionDCA has been explored in the unpublished Master’s thesis of A. D’Urso (EPFL), where the model is applied to paired protein sequences composed of two interacting domains α and β . The system under study consists of histidine kinase-response regulator (HK-RR) pairs [99], a prototypical example of interacting protein families in bacterial two-component signaling. The dataset consists of multiple sequence alignments of concatenated sequences $\mathbf{a} = (\mathbf{a}^\alpha, \mathbf{a}^\beta) = (a_1^\alpha, \dots, a_{L_\alpha}^\alpha, a_{L_\alpha+1}^\beta, \dots, a_{L_\alpha+L_\beta}^\beta)$, constructed so as to preserve the pairing between interacting partners across species. In this setting, the statistical signal encodes both intra-domain constraints within each protein and inter-domain coevolution between the two partners.

For such systems, the interaction tensor can be decomposed as in Eq. 5.2, where $J^{\alpha\alpha}$ and $J^{\beta\beta}$ correspond to intra-domain contributions, while $J^{\alpha\beta} = J^{\beta\alpha}$ to inter-domain ones. Although standard AttentionDCA does not explicitly enforce this decomposition, it can be induced by partitioning the set of attention heads into disjoint subsets associated with different interaction components. Denoting by $\mathcal{H}_{\alpha\alpha}$, $\mathcal{H}_{\beta\beta}$ and $\mathcal{H}_{\alpha\beta}$ the subsets of heads assigned to each contribution, one can consider a parametrization of the form

$$J_{ij}(a, b) = \sum_{h \in \mathcal{H}_{\alpha\alpha}} J_{ij}^{(h)}(a, b) + \sum_{h \in \mathcal{H}_{\beta\beta}} J_{ij}^{(h)}(a, b) + \sum_{h \in \mathcal{H}_{\alpha\beta}} J_{ij}^{(h)}(a, b). \quad (5.3)$$

In practice, this separation is enforced through two complementary mechanisms. First, the attention maps $A_{ij}^{(h)}$ are constrained to have support only within predefined blocks of the sequence, so that each subset of heads operates exclusively on $\alpha\alpha$, $\beta\beta$, or $\alpha\beta$ interactions. Second, the total likelihood function is decomposed as

$$\mathcal{L} = \mathcal{L}_{\alpha\alpha} + \mathcal{L}_{\beta\beta} + \mathcal{L}_{\alpha\beta} + \mathcal{L}_{\beta\alpha}, \quad (5.4)$$

and the different contributions can be selectively optimized or weighted, aligning the structure of the model with that of the interaction tensor.

An example of this construction is shown in Fig. 5.1, where inter-domain contact predictions obtained from the standard AttentionDCA model and the domain-decoupled version are compared. In the standard implementation, the model is trained on the full concatenated system with $H = 100$ heads, shared across both intra-

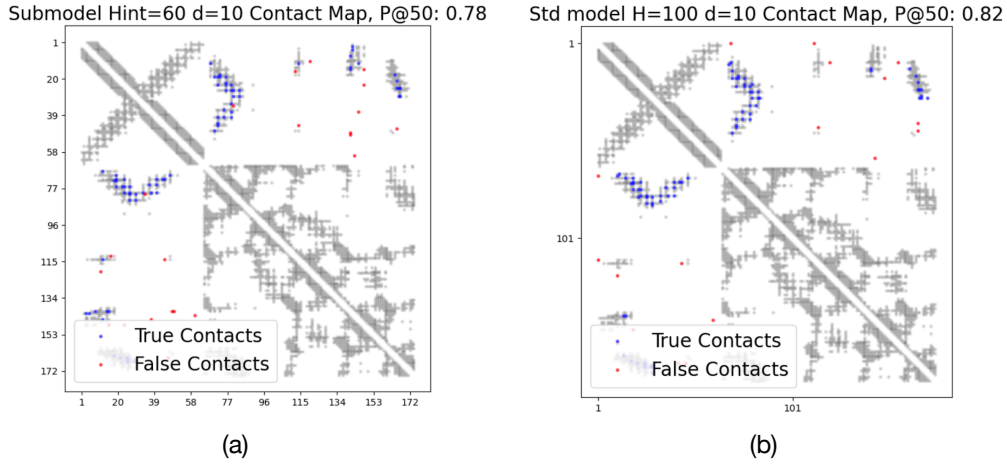


Fig. 5.1 Contact prediction for interacting domains in the histidine kinase-response regulator (HK-RR) system. (a) Domain-decoupled AttentionDCA model with $H_{\text{int}} = 60$ heads dedicated to inter-domain interactions. (b) Standard AttentionDCA model with $H = 100$ heads shared across all interaction components. In both cases, only inter-domain contacts are shown and used to compute the performance metric. Gray points denote true contacts, blue points correctly predicted contacts, and red points false positives. The decoupled model achieves comparable performance ($\text{PPV@50} \approx 0.78$ vs 0.82) while producing a representation that is aligned with the block structure of the interaction tensor. Adapted from Fig. 4.13 of the unpublished Master's thesis of A. D'Urso.

and inter-domain interactions. As a result, the model is effectively required to extract the inter-domain signal from a representation in which all interaction components are mixed. In contrast, the decoupled model employs $H_{\text{int}} = 60$ heads explicitly assigned to inter-domain interactions. Despite the reduced number of parameters dedicated to the interaction term, the decoupled model achieves a comparable quality of contact prediction ($\text{PPV@50} \approx 0.78$ vs 0.82), while producing a representation that is more directly focused on the relevant interaction block. It is important to note that, for the evaluation of the standard version shown in Fig. 5.1 (b), only inter-domain contacts are retained and used to compute the Positive Predictive Values (PPV) metric.

This comparison suggests that part of the capacity of the standard model is redundantly distributed across interaction components and must be implicitly reallocated during inference. By introducing explicit structural constraints, this capacity can be more efficiently concentrated on the relevant part of the interaction tensor.

These results show that, while AttentionDCA does not automatically disentangle different components of the interaction tensor, its parametrization provides the necessary structure to enforce such a decomposition when appropriate constraints are introduced. More generally, this illustrates how the modularity of the model can be exploited to impose explicit structural assumptions on the interaction space, enabling a controlled factorization of the coupling tensor in systems with multiple interacting components.

5.2 AttentionDCA in CLAPP: Class Pooling for Paralog Matching

In this section, we consider a concrete application of AttentionDCA to the problem of paralog matching, arising from a collaboration with the Laboratory of Statistical Biophysics at EPFL. We first introduce the CLAPP (Class Pooling for Paralog Matching) framework, which formulates paralog matching as a class-level inference problem. We then describe how AttentionDCA can be used as an interaction score within this setting. Finally, we discuss how the structure of the attention-based parametrization naturally suggests extensions in which class-level information could be incorporated directly into the model. The content and figures presented in this section are based on the material from the paper "*Partner determination from protein sequences using class information*" [96], adapted for the purposes of this thesis.

5.2.1 Paralog Matching and Coevolutionary Inference

Protein-protein interactions play a central role in most cellular processes, making their systematic identification a key problem in molecular biology. While experimental approaches provide high-quality interaction data, they remain costly and incomplete, motivating the development of computational methods capable of inferring interaction partners directly from sequence information.

In many protein families, the presence of multiple paralogs within the same organism introduces an additional layer of complexity. Gene duplication events give rise to sets of homologous sequences that can evolve distinct functional roles, often leading to specific pairing preferences between members of interacting families. As

a result, the identification of interacting partners requires resolving a combinatorial matching problem, in which multiple candidate pairs coexist within each species [100–102].

A classical strategy to address this problem exploits genomic proximity, based on the observation that interacting proteins are often encoded by nearby genes, particularly in prokaryotes [103]. While effective in some cases, this approach is limited to specific organisms and does not capture the full diversity of interaction networks. More generally, sequence-based methods provide a broader framework by leveraging the coevolutionary signal encoded in protein sequences. Interacting proteins evolve under shared selective constraints, and compensatory mutations at their interfaces lead to statistical dependencies that can be detected from multiple sequence alignments.

Within this framework, coevolutionary models such as Direct Coupling Analysis (DCA) have been widely used to define interaction scores between candidate pairs [22, 104, 105]. Given a concatenated alignment of sequences from two families, a statistical model is inferred and used to assign a score quantifying the compatibility of two sequences under the model. Paralog matching can then be formulated as an assignment problem, in which, for each species, one seeks the optimal one-to-one pairing between members of the two families.

Despite their success, these approaches present important limitations. The interaction scores are often noisy, especially in regimes with limited data or weak coevolutionary signal, and must be used to resolve a combinatorial matching problem within each species. In the absence of additional structure, the assignment relies entirely on these pairwise scores, making it sensitive to ambiguities between competing matches. This suggests that incorporating additional sources of information, when available, could help constrain the inference problem. In particular, if interaction specificity is organized at a higher level, for instance through class-dependent interactions, this structure can be exploited to reduce the effective complexity of the matching problem.

5.2.2 Class Pooling in CLAPP

CLAPP is based on the hypothesis that protein families can be partitioned into classes such that interactions are largely determined at the class level. More precisely,

sequences belonging to a given class in one family preferentially interact with sequences from specific classes in the partner family. When such a structure is present, the matching problem can be reformulated in terms of interactions between classes rather than between individual sequences. The method does not rely on a specific *a priori* definition of classes; rather, the class structure is defined depending on the system under study. In particular, it can reflect biological organization, such as functional or structural subclasses, or be obtained through statistical procedures, as in the case of clustering in principal component space.

Once the definition of the classes is determined for the given system, the matching problem can be reformulated by introducing a matrix S of dimensions $C_A \times C_B$, where C_A and C_B denote the number of classes in the two families. Instead of determining a permutation between individual sequences, the model infers a class-level assignment, thereby reducing the dimensionality of the problem.

For a dataset comprised of candidate paired sequences $\mathbf{a} = (\mathbf{a}^A, \mathbf{a}^B)$ from families A and B , we define the cost matrix $C \in \mathbb{R}^{C_A \times C_B}$ as

$$C_{ab} = \frac{\sum_{m=1}^M \delta(\sigma_m = a) \delta(\tau_m = b) I(\mathbf{a}^m) P(\mathbf{a}^m)}{\sum_m \delta(\sigma_m = a) \delta(\tau_m = b)}, \quad (5.5)$$

with $I(\mathbf{a}^m)$ the interaction score of paired sequence $\mathbf{a}^m$ and $P(\mathbf{a}^m) = 1/n_{s(m)}$ a within-organism prior, where $n_{s(m)}$ is the number of paralogs in organism $s(m)$. This construction corresponds to pooling interaction scores across all pairs of sequences assigned to the same pair of classes, effectively aggregating information across organisms.

The matching problem is then formulated as the minimization of

$$\sum_{a=1}^{C_A} \sum_{b=1}^{C_B} C_{ab} S_{ab}, \quad (5.6)$$

under the constraints $\sum_a S_{ab} = B_b$ and $\sum_b S_{ab} = A_a$. In the case $C_A = C_B$, choosing $A_a = 1/C_A$ and $B_b = 1/C_B$ leads to a scaled permutation matrix at the class level, which can be efficiently obtained using linear programming.

Once the class-level assignment is determined, it provides a global constraint that can be used to guide the matching at the sequence level. In this way, CLAPP replaces a collection of independent matching problems with a single structured

inference problem in a reduced space, whose dimensionality is determined by the number of classes rather than by the number of sequences.

The effectiveness of this approach relies on the validity of the underlying hypothesis: when interaction specificity is organized at the class level, pooling interaction scores amplifies coherent signals across related sequences and reduces the impact of noise in individual pairwise interactions.

5.2.3 AttentionDCA as an Interaction Score

The CLAPP framework requires two main ingredients: an interaction score $I(\mathbf{a})$ defined on candidate pairs of sequences, and a definition of classes for the two protein families. While the latter depends on the specific system under study and is not addressed here, we focus on the choice of the interaction score and consider AttentionDCA for this purpose.

As introduced in Chapter 4, AttentionDCA defines an energy-based statistical model in which the Potts coupling tensor $J_{ij}(a, b)$ is parametrized through the factored-attention form of Eq. 4.3. Given two interacting protein families A and B, the model is trained on a concatenated multiple sequence alignment of pairs $\mathbf{a} = (\mathbf{a}^A, \mathbf{a}^B)$. Since the correct matching between paralogs is not known *a priori*, the training alignment is constructed from all possible within-organism A-B concatenations. This dataset therefore contains both correct and incorrect pairs, but the correct pairs are included among the candidates and are expected to contribute coherently to the inter-family coevolutionary signal.

After training, the interaction score of a candidate pair is computed from the inter-family contribution to the Potts energy,

$$I(\mathbf{a}) = -E_{\text{int}}(\mathbf{a}) = \sum_{i=1}^{L_A} \sum_{j=1}^{L_B} J_{ij}(a_i, a_j), \quad (5.7)$$

where i and j run over the aligned positions of families A and B, respectively. Therefore, the score depends only on couplings across the two protein families, while intra-family terms are not used to rank candidate interactions. Pairs with stronger inter-family couplings receive higher interaction scores.

An example of the performance obtained using AttentionDCA within the CLAPP framework is shown in Fig. 5.2, for three representative bacterial paralog-matching benchmarks: histidine kinase-response regulator (HK-RR) systems [99], Hsp70-J-domain pairs, and the ModB-ModC components of the molybdate/tungstate ABC importer [106]. In these systems, reference interacting pairs are defined at the sequence level using genomic proximity, a common proxy for physical interaction in bacterial protein families. For HK-RR and ModBC, the pairings are taken from previously curated benchmarks, while the Hsp70-J-domain dataset is constructed by selecting bacterial homologs of the two families and retaining high-confidence candidate interactions based on genomic neighborhood. In the results shown here, AttentionDCA is trained with $H = 150$ attention heads and internal dimension $d = 23$, with the number of training epochs selected by early stopping. Class labels are obtained by applying k-means clustering in the principal-component space of one-hot encoded sequences. This provides a sequence-based partition of each family into subclasses, which are then used by CLAPP to pool interaction scores across organisms at the class level.

We compare standard within-species matching based on AttentionDCA interaction scores with the class-pooled inference provided by CLAPP, together with additional baselines and reference methods. While the AttentionDCA score alone already captures a non-trivial interaction signal, the introduction of class pooling consistently leads to a substantial improvement in matching accuracy across all systems, indicating that part of the information required to resolve paralog matching is more effectively exploited at the class level.

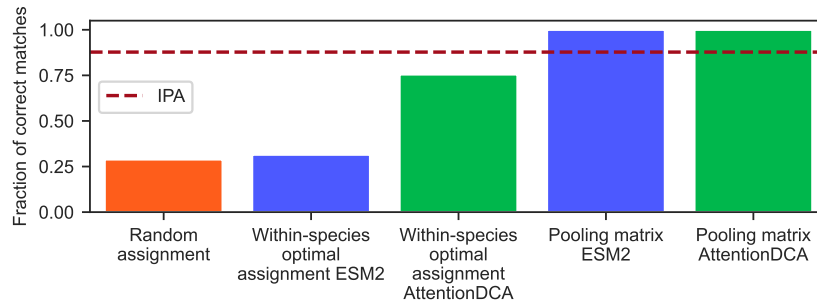
From a modeling perspective, CLAPP introduces class-level structure through the explicit construction of the pooled cost matrix C_{ab} , obtained by aggregating pairwise interaction scores. The empirical success of this approach suggests that interaction specificity is, at least in part, organized at the class level, and that such information can be effectively exploited to constrain the matching problem. The attention-based parametrization of AttentionDCA suggests a complementary perspective, in which this structure could be incorporated directly within the model. In particular, the factorized form of the coupling tensor naturally separates positional and compositional contributions, and could be extended to include additional variables, such as class labels, alongside sequence positions and amino-acid identities.

In this view, the explicit aggregation step used to construct C_{ab} can be interpreted as a first approximation to a more integrated, end-to-end formulation, in which class-level information is directly encoded in the parametrization of the interaction model. While this direction is not explored here, it highlights how the flexibility of the AttentionDCA parametrization may enable the integration of different sources of information within a unified framework.

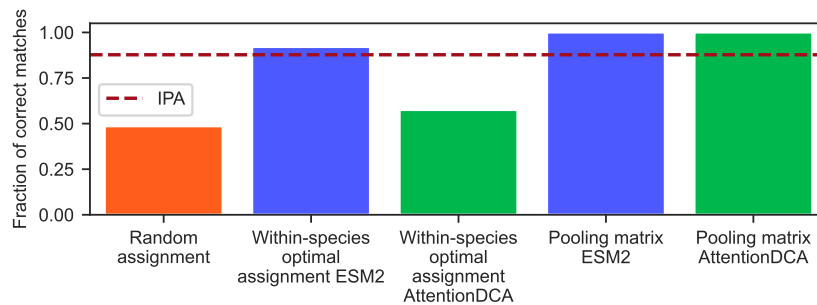
5.3 Conclusion

The results presented in this chapter suggest that AttentionDCA should not be viewed solely as a tool for interpreting black-box models, but rather as a flexible and computationally efficient statistical model in its own right. Its parametrization makes it naturally suited to problems in which additional structure can be exploited, such as in the presence of subdomains or class-dependent interactions.

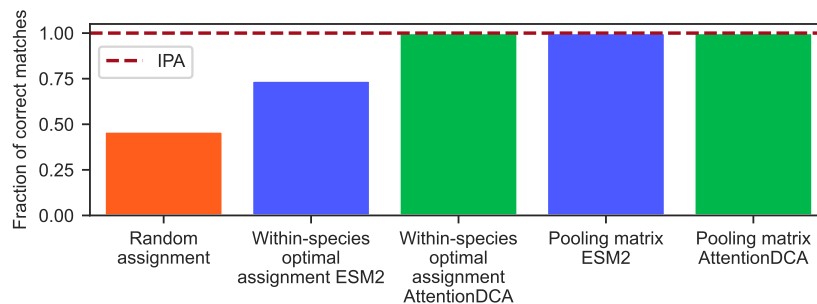
In this sense, AttentionDCA provides a simple yet structured framework that can be adapted to specific inference tasks, going beyond standard coevolutionary modeling while retaining a level of interpretability and control that is difficult to achieve with more complex models.



(a) Histidine kinase-response regulator (HK-RR)



(b) Hsp70-J-domain system



(c) ModBC (2ONK) complex

Fig. 5.2 Paralog matching performance for different protein systems using AttentionDCA interaction scores, with and without class pooling. Results are shown for (a) histidine kinase-response regulator (HK-RR), (b) bacterial Hsp70-J-domain pairs, and (c) the ModBC (2ONK) complex. In all cases, matching is performed either within species or using the CLAPP pooling framework. Ground truth interactions are inferred from genomic proximity, and clustering is performed using k-means to define class labels. Performance is compared to the iterative pairing algorithm (IPA). Adapted from [96].

Chapter 6

FeatureDCA: Controllable Generative Modeling of Protein Families

In this chapter we present FeatureDCA, an extension of autoregressive DCA [26] designed to introduce controllability in protein sequence generation. While standard generative DCA models accurately reproduce the statistics of multiple sequence alignments, they cannot steer the sampling toward specific regions of sequence space, and alternative generative approaches that allow such control are generally less interpretable. FeatureDCA addresses this gap by embedding biologically meaningful low-dimensional features into the autoregressive Potts framework, preserving interpretability while enabling targeted and flexible generative modeling. Here, we describe the theoretical formulation of the model, its implementation, and its validation on global and conditional generative tasks.

The content, figures, and tables presented in this chapter are based on the material from the paper “*Controllable Protein Design via Autoregressive Direct Coupling Analysis Conditioned on Principal Components*” [43], adapted and expanded for the purposes of this thesis.

6.1 Introduction and Motivation

The guiding motivation of this thesis, as outlined in Chapter 1, is to develop and analyze shallow statistical learning models for computational biology, with proteins

as the central case study. The purpose is twofold: on the one hand, to propose interpretable alternatives to the black-box methods that dominate modern protein analysis and design; on the other, to use transparent statistical frameworks as tools to shed light on the mechanisms that underlie these black boxes. In Chapters 2 and 3 we presented these two complementary perspectives. Chapter 2 reviewed Potts-based approaches derived from Direct Coupling Analysis, which yield data-efficient and interpretable generative models of protein families. Chapter 3 described the rise of deep learning approaches, from early protein language models to large-scale transformer architectures, which, although flexible and powerful, typically conceal the principles they exploit inside non-transparent high-dimensional embeddings.

The development of AttentionDCA in Chapter 4 exemplified how these two lines of thought can be combined. By explicitly mapping the attention mechanism to Potts-like couplings, AttentionDCA served both as a model for contact prediction and as a framework to interpret the internal representations of architectures such as AlphaFold or MSA Transformer. Its autoregressive extension further highlighted the generative potential of such shallow models. Nevertheless, like all generative DCA models, including bmDCA and ArDCA, AttentionDCA remains essentially unconditional: once trained, it samples sequences that reflect the statistics of the underlying alignment. Still, it cannot bias the generative process toward a desired structural or functional class. This absence of controllability is a crucial limitation. In protein design, one often wants to generate sequences that not only resemble natural families but also occupy specific regions of sequence space associated with different subtypes, phenotypes, or environmental adaptations. Without a mechanism to steer sampling, Potts-based models cannot address this need.

On the opposite side, several deep learning frameworks have been developed in recent years for protein sequence and structure generation. Variational autoencoders (VAEs), exemplified by DeepSequence [32], introduced a probabilistic latent-variable formulation in which protein sequences are encoded into a low-dimensional latent space and decoded back to sequences. While DeepSequence was primarily applied to mutational effect prediction, it is generative by construction and has subsequently inspired extensions that explore sampling from its latent space to generate novel sequences. In particular, generative adversarial networks (GANs) [107] extended this idea by training discriminators and generators that can be conditioned on labels, although their application to proteins has remained limited. More recently, large-scale transformer-based protein language models have become the dominant

paradigm. Models such as ProGen [108] demonstrated that conditioning generation on textual prompts allows one to target broad protein classes or functions. The ESM family of models [38], trained on hundreds of millions of sequences, showed that large embeddings can be leveraged not only for structure prediction but also for mutational effect estimation and conditional generation. Diffusion-based approaches, inspired by advances in generative modeling in computer vision, have also been applied to proteins [109, 110], allowing for iterative refinement of sequences starting from random noise and offering additional control through the manipulation of intermediate diffusion states.

These models convincingly demonstrate that controllability is possible, but they achieve it through mechanisms that are difficult to interpret. Conditioning is implemented via high-dimensional latent spaces, prompt tuning, or the manipulation of embeddings that are not directly connected to the evolutionary constraints encoded in MSAs. As a consequence, while one can bias a model such as ProGen or ESM toward a specific protein family or function, it is not clear how this control relates to the physical or evolutionary forces shaping sequence variability.

The motivation for the present chapter is therefore to close this gap. The aim is to retain the efficiency and interpretability of Potts-based inference while endowing the model with a controllable component, able to bias generativity toward user-defined directions in sequence space. This is precisely the idea behind FeatureDCA. Rather than relying on latent spaces or prompt engineering, the model introduces principled conditioning in a transparent and shallow way, linking it explicitly to low-dimensional biologically meaningful features. In this manner, FeatureDCA provides a framework for controllable generation that complements the interpretability of statistical physics models with a capacity that, until now, has largely been confined to their deep learning counterparts. In particular, in this analysis, we focus on principal components of the MSA as conditioning variables, since they provide a compact representation of sequence variability, but the same approach can be naturally extended to other experimentally or computationally derived features.

In the remainder of this chapter, we show that FeatureDCA attains generative performance comparable to, and in some cases exceeding, established models such as bmDCA and ArDCA, while at the same time allowing controllable, feature-conditioned sampling. In particular, we demonstrate that the model reproduces both the pairwise correlations and the low-dimensional projections of natural MSAs,

all while maintaining substantial sequence diversity. This balance between fidelity to evolutionary constraints and exploration of novel regions of sequence space is essential for practical protein design.

Beyond statistical validation, we assess the structural plausibility of generated sequences using structure prediction tools such as AlphaFold and ESMFold. A detailed case study on the Response Regulator family (PF00072) illustrates how conditioning on PCA-derived features enables the generation of sequences that recover distinct structural subclasses within the same protein domain. Finally, we benchmark the model on deep mutational scanning data, showing that FeatureDCA retains predictive power for the effects of point mutations. Together, these results position FeatureDCA as a flexible and interpretable framework for controllable protein sequence generation, bridging the strengths of Potts-based inference with the steering capabilities that, until now, were mostly confined to deep learning approaches

6.2 Model

In this section, we introduce the mathematical formulation of FeatureDCA. The goal is to extend the autoregressive Potts models described in Chapter 2 so that sequence generation can be explicitly conditioned on low-dimensional biologically meaningful features. To this end, we first describe how the joint distribution of sequences and feature vectors can be represented within an autoregressive framework. We then motivate the functional form of the conditional probabilities by means of a Gaussian approximation, which naturally leads to an additional term in the Hamiltonian coupling sequences and features. Finally, we define the inference procedure based on maximum likelihood estimation. This formulation provides the foundation on which the generative and predictive properties of FeatureDCA, presented in the following sections, are built.

6.2.1 Autoregressive modeling and Gaussian approximation

From a theoretical point of view, the starting point of FeatureDCA is the idea of extending the autoregressive implementation of the Potts model so as to describe not only protein sequences but also a set of associated low-dimensional features. Given a

multiple sequence alignment of M homologous proteins of length L , we denote each sequence by $\mathbf{a}^m = (a_1^m, \dots, a_L^m)$ and associate to it a feature vector $\mathbf{y}^m \in \mathbb{R}^d$, where the components of $\mathbf{y}^m$ capture biologically meaningful information such as principal components of the alignment or other experimental or computational annotations. The dataset therefore consists of paired observations

$$D = \{(\mathbf{a}^m, \mathbf{y}^m)\}_{m=1}^M. \quad (6.1)$$

The objective is to learn the joint distribution $P(\mathbf{a}, \mathbf{y})$, which in turn allows us to generate new protein sequences conditioned on fixed feature vectors. Following the autoregressive decomposition introduced in Chapter 2, we factorize the conditional distribution of sequences given features as

$$P(\mathbf{a}|\mathbf{y}) = \prod_{k=1}^L P(a_k | a_{<k}, \mathbf{y}), \quad (6.2)$$

where $a_{<k} = (a_1, \dots, a_{k-1})$, with the convention that $a_{<1}$ is the empty set. This decomposition is exact and shifts the problem to specifying suitable functional forms for the single-site conditionals $P(a_k | a_{<k}, \mathbf{y})$.

In the standard ArDCA model, these conditional distributions are parametrized by a Potts Hamiltonian involving pairwise couplings and local fields, as described in Chapter 2. In FeatureDCA, the presence of conditioning features requires an ansatz that explicitly couples the categorical variables representing amino acids with the continuous variables representing features. To motivate this ansatz, we first introduce a one-hot representation of amino-acid sequences such that $\mathbf{a} = (\mathbf{a}_1, \dots, \mathbf{a}_L)$, such that $\mathbf{a}_i \in \{0, 1\}^q$, and then we apply a Gaussian approximation. Withing this scheme, we relax the discrete nature of amino acids $\mathbf{a}_i$ to real values and we model the $P(\mathbf{a})$ and $P(\mathbf{y})$ as Gaussian distributions parameterized by mean and variance $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$ respectively, so that

$$P(\mathbf{a}) = \mathcal{N}(\mathbf{a} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad , \quad (6.3)$$

$$P(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}) \quad . \quad (6.4)$$

Suppose that, as in the case of PCA, there exists a unique mapping $U \in \mathbb{R}^{d, Lq}$ that projects one-hot encoded sequences $\mathbf{a} \in \mathbb{R}^{Lq}$ into the corresponding real vector of d -dimensional features $\mathbf{y} = U\mathbf{a}$. In this setting, the conditional distribution $P(\mathbf{y}|\mathbf{a})$

can be written as a Dirac delta function in a d -dimensional vector space:

$$P(\mathbf{y}|\mathbf{a}) = \delta^{(d)}(\mathbf{y} - U\tilde{\mathbf{a}}) \quad (6.5)$$

It then follows that the distribution of feature vectors $\mathbf{y}$ can be obtained as the marginal of the joint distribution $P(\mathbf{a}, \mathbf{y})$ as in:

$$\begin{aligned} P(\mathbf{y}) &= \int d\mathbf{a} P(\mathbf{a}, \mathbf{y}) = \int d\mathbf{a} P(\mathbf{y}|\mathbf{a}) P(\mathbf{a}) \\ &= \int d\mathbf{a} \delta^{(d)}(\mathbf{y} - U\mathbf{a}) \mathcal{N}(\mathbf{a}|\mu, \Sigma) = \mathcal{N}(\mathbf{y}|U\mu, U\Sigma U^\dagger) \quad , \end{aligned} \quad (6.6)$$

so that $\tilde{\mu} = U\mu$ and $\tilde{\Sigma} = U\Sigma U^\dagger$. The explicit calculation of this result can be found in Appendix C.1. Starting from Equations 6.3 and 6.6, one can compute marginal and conditional distributions in closed form. In particular, Bayes' theorem gives:

$$P(\mathbf{a}|\mathbf{y}) = \frac{P(\mathbf{a}, \mathbf{y})}{P(\mathbf{y})} = \frac{\delta^{(d)}(\mathbf{y} - U\mathbf{a}) \mathcal{N}(\mathbf{a}|\mu, \Sigma)}{\mathcal{N}(\mathbf{y}|U\mu, U\Sigma U^\dagger)} \quad . \quad (6.7)$$

From this expression, the marginal $P(a_i|\mathbf{a}_{<i}, \mathbf{y})$ can be computed iteratively. Starting with

$$P(\mathbf{a}_1|\mathbf{y}) \int d\mathbf{a}_{-1} \frac{\delta^{(d)}(\mathbf{y} - U\mathbf{a}) \mathcal{N}(\mathbf{a}|\mu, \Sigma)}{\mathcal{N}(\mathbf{y}|U\mu, U\Sigma U^\dagger)} \quad , \quad (6.8)$$

and continuing up to the k -th term

$$\begin{aligned} P(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}) &= \int d\mathbf{a}_{>k} P(\mathbf{a}_k, \dots, \mathbf{a}_L|\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) \\ &= \int d\mathbf{a}_{>k} \frac{P(\mathbf{a}|\mathbf{y})}{\prod_{i=1}^{k-1} P(a_i|\mathbf{a}_{<i}, \mathbf{y})} \quad . \end{aligned} \quad (6.9)$$

The details of the computation can be found in Appendix C.1. From this Gaussian approximation, the k -th conditional marginal takes the form

$$P(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}) = \frac{e^{H(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y})}}{Z_{\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}}} \quad , \quad (6.10)$$

where $Z_{\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}}$ is the normalization constant defined as

$$Z_{\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}} = \sum_{\mathbf{a}_k} e^{H(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y})} \quad , \quad (6.11)$$

and $H(\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y})$ is the energy function given by

$$H(\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}) = \frac{1}{2} \mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}}^{-1} \mathbf{a}_k + \quad (6.12)$$

$$- \mathbf{a}_k^\dagger \left\{ \sum_{i=1}^{k-1} \left[U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] - \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{a, >k}^{-1} U_{>k}^\dagger \mathbf{y} \right\}$$

Here, with the notations A_k and $A_{>k}$ we define the sub-matrices of A indexed by k and $>k$. This continuous approximation motivates the functional form of the conditional probabilities, which we now translate into a discrete Potts-like model suitable for protein sequences.

6.2.2 Discrete ansatz

The Gaussian approximation leading to the continuous form of the conditional marginals in Equation 6.10 naturally suggests an equivalent ansatz once the discrete nature of amino-acid sequences is restored. In particular, Equation 6.12 can be decomposed into three contributions, each with a clear geometric interpretation. The first term is quadratic in the amino-acid variables and corresponds to the pairwise couplings of a standard Potts model. The second is linear in the amino-acid states and can be identified with local fields. The third introduces a novel cross term that couples the amino-acid variables to the external feature space, thereby extending the Potts Hamiltonian with an explicit conditioning component. In the continuous formulation, the additional feature-dependent contribution takes the form

$$\mathbf{a}_k^\dagger \tilde{G}_k \mathbf{y} := \mathbf{a}_k^\dagger \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{a, >k}^{-1} U_{>k}^\dagger \mathbf{y} \quad , \quad (6.13)$$

with $\tilde{G}_k \in \mathbb{R}^{q,d}$. When returning to the discrete representation of amino-acid variables, this term can be expressed in the more compact form

$$G_k(a_k) \cdot \mathbf{y} \quad (6.14)$$

where $G_k(a_k)$ denotes a d -dimensional embedding vector associated with residue a_k at position k . The full conditional Hamiltonian can be written as

$$H(a_k|a_{<k}, \mathbf{y}) = - \sum_{i<k} J_{ki}(a_k, a_i) - h_k(a_k) - \mathbf{y} \cdot \mathbf{G}_k(a_k) \quad , \quad (6.15)$$

while the functional form of the k -th conditional probability distribution is

$$\begin{aligned} P(a_k|a_{<k}, \mathbf{y}) &= \frac{1}{Z_{a_k|a_{<k}, \mathbf{y}}} e^{\sum_{i<k} J_{ki}(a_k, a_i) + h_k(a_k) + \mathbf{y} \cdot \mathbf{G}_k(a_k)} \\ &= \frac{e^{\sum_{i<k} J_{ki}(a_k, a_i) + h_k(a_k) + \mathbf{y} \cdot \mathbf{G}_k(a_k)}}{\sum_{c=1}^q e^{\sum_{i<k} J_{ki}(c, a_i) + h_k(c) + \mathbf{y} \cdot \mathbf{G}_k(c)}} \end{aligned} \quad (6.16)$$

The parameters of the model are learned by maximizing the conditional likelihood of the dataset D defined in Equation 6.1, Explicitly, the objective function is given by

$$\mathcal{L} = \frac{1}{M} \sum_{m=1}^M \sum_{k=1}^L \log P(a_k^m | a_{<k}^m, \mathbf{y}^m) \quad , \quad (6.17)$$

with L_2 regularization applied to all parameters to prevent overfitting. The inference procedure is therefore analogous to the maximum likelihood learning introduced in Chapter 2 for arDCA and pseudolikelihood maximization in PlmDCA, with the sole difference that the conditional probabilities now include the additional feature-dependent contribution $\mathbf{y} \cdot \mathbf{G}_k(a_k)$.

In practice, this means that the couplings $\{J_{ki}\}$ and fields $\{h_k\}$ retain their interpretation as evolutionary constraints encoded in the alignment, while the new embedding vectors $\{G_k\}$ describe how external features bias the probability of observing each residue at a given position. Throughout this chapter, we will restrict our analysis to the case in which the conditioning variables $\mathbf{y}$ are taken to be the principal components of the MSA. Principal Component Analysis (PCA) provides a linear transformation of the alignment that identifies the dominant axes of sequence variability, ordered by the fraction of variance explained. In this way, each sequence can be associated with a low-dimensional vector $\mathbf{y}$ capturing its main statistical features in an unsupervised manner. However, the formulation of the model is general, and in principle other biologically meaningful features—such as fitness, stability, binding specificity, or phylogenetic information—could be incorporated in exactly the same way.

6.2.3 Sampling procedure

Once the model parameters have been inferred, new sequences can be generated in an autoregressive fashion. As in ArDCA, the conditional factorization of the sequence distribution allows us to sample one site at a time, proceeding sequentially from the first to the last position. In FeatureDCA, this procedure is extended by conditioning the sampling process on the user-defined feature vector $\mathbf{y}$. At each iterative step, residue a_k is drawn from the conditional distribution $P(a_k|a_{<k}, \mathbf{y})$ defined in Equation 6.15. In this way, the entire sequence is generated consistently with both the evolutionary constraints encoded by the Potts parameters and the bias imposed by the external features.

This scheme ensures that sampling is efficient, since it only requires evaluating univariate conditional distributions at each position, and at the same time it makes the conditioning explicit: by selecting a desired point $\mathbf{y}$ in feature space, one can guide the model toward generating sequences in specific regions of sequence variability. In the following section, we will assess the quality of these generated sequences across a range of tasks.

6.3 Evaluation tasks

As discussed in Chapter 2, the natural way to benchmark generative DCA models is to assess how well they reproduce the statistical properties of the sequence ensembles from which they are trained. FeatureDCA is no exception: its performance must first be evaluated on its ability to replicate the global statistics of the MSA, as any generative Potts model would do. Beyond this, the introduction of conditioning opens new types of tasks. In particular, we can test the model’s capacity to generate sequences at specific points of the feature space, here represented by the principal component projection of the MSA. For such sequences, we can quantify similarity to known wild types using Hamming distance, proximity in PCA space, and structural validation through AlphaFold and ESMFold. A detailed case study on the Response Regulator family (PF00072) illustrates this aspect, highlighting how conditioning can be used to recover distinct structural subclasses. Finally, as in Chapter 2, we also consider the task of predicting mutational effects from deep mutational scanning data, which provides a stringent test of whether the model retains the local evolutionary

constraints encoded in natural sequences. The following subsections describe these evaluation tasks in detail.

6.3.1 Reproducing global statistics

A first benchmark for any generative model is its ability to reproduce the global statistical properties of the training alignment. In the case of FeatureDCA, this requires that sequences sampled from the model display the same empirical frequencies and correlations observed in the natural MSA. This comparison is performed by constructing an artificial alignment that mirrors the natural one both in size and in feature coverage. Specifically, for each natural sequence $(\mathbf{a}, \mathbf{y}) \in D$, an artificial one is sampled conditionally to the specific feature vector $\mathbf{y}$. This procedure ensures that the natural and artificial alignments share the same empirical distribution of features, so that differences in statistics directly reflect the generative performance of the model.

The global statistical comparison is carried out at different levels. In the first place, we evaluate the Pearson correlation coefficient between the connected two-site correlations of empirical and model-generated MSAs as defined in Equation 2.49. Secondly, we exploit the fact that conditioning is defined in principal component space and compare the distributions of natural and generated sequences after projection onto the top principal components of the natural distribution. To evaluate the matching of the two, we use the Sinkhorn divergence, an entropic-regularized optimal transport distance that quantifies how well two normalized distributions overlap in their support space. Concretely, let $\mu(\mathbf{y}) = \frac{1}{M} \sum_{m=1}^M \delta(\mathbf{y}_{\text{nat}}^m, \mathbf{y})$ and $\nu(\mathbf{y}) = \frac{1}{M} \sum_{m=1}^M \delta(\mathbf{y}_{\text{gen}}^m, \mathbf{y})$ be the empirical distributions of natural and generated sequences in PC space. The regularized optimal transport cost $\text{OT}_\varepsilon(\mu, \nu)$ measures the minimal average squared distance required to transport μ into ν , with an entropic term controlled by ε that stabilizes the computation. The Sinkhorn divergence is then defined as

$$S_\varepsilon(\mu, \nu) = \text{OT}_\varepsilon(\mu, \nu) - \frac{1}{2}\text{OT}_\varepsilon(\mu, \mu) - \frac{1}{2}\text{OT}_\varepsilon(\nu, \nu), \quad (6.18)$$

which enforces the triangular inequality, symmetry and ensures that $S_\varepsilon(\mu, \mu) = 0$. Details on the exact computation of the Sinkhorn divergence are given in Appendix C.4. In practice, this metric provides a sensitive and geometry-aware measure of how

well FeatureDCA reproduces the variability of natural sequences along the principal components. Together with the correlation of the summary statistics in the form of connected correlations, it defines the baseline generative test that FeatureDCA must satisfy before considering more specific conditioning tasks.

6.3.2 Conditional generation in feature space

A central novelty of FeatureDCA compared to previous DCA-based models is its ability to generate sequences at specific locations of the feature space, rather than only reproducing ensemble statistics. This capability makes it possible to focus sampling on regions of variability associated with particular wild types or functional subclasses.

The task is defined as follows. Given a target feature vector $\mathbf{y}$, typically taken as the PCA coordinates of a wild-type sequence, we sample new sequences from the conditional distribution $P(\mathbf{a}|\mathbf{y})$. The result is a synthetic ensemble of sequences anchored to the chosen point in feature space, but still diversified by the autoregressive sampling process that allows for a certain level of variance. This procedure, which we refer to as in-place generativity, allows us to explore local neighborhoods of sequence space in a controlled manner.

The quality of conditional generation is evaluated using complementary metrics. First, having chosen a target, we compute the average Hamming distance between generated sequences and the corresponding wild type, which indicates the degree of novelty introduced by the model while remaining within the same feature space neighborhoods. Second, we measure the Euclidean distance in PCA space between the generated sequences and the target point $\mathbf{y}$, which quantifies how closely the conditioning keeps the samples localized. Finally, we validate structural similarity by folding the generated sequences with AlphaFold or ESMFold and comparing the resulting structures to the experimental PDB structure of the target wild type, using the root-mean-square deviation (RMSD) between atomic positions as a reference measure.

In the Results section, the sequence-level metrics (Hamming distance and PCA distance) will be evaluated across multiple protein families to provide a systematic assessment of conditional generativity. Structural validation, on the other hand, will be addressed through a dedicated case study on the Response Regulator (RR)

family (PF00072). These proteins share a conserved receiver domain but exhibit different structural modes of dimerization, which can be separated in PCA space. This makes PF00072 an ideal test case for FeatureDCA, since conditioning on the appropriate feature coordinates should allow the generation of sequences that fold into the corresponding structural arrangements.

6.3.3 Mutational effect prediction

In addition to generative tasks, FeatureDCA can be evaluated on its ability to capture the local evolutionary constraints that determine the impact of single-residue mutations. This benchmark, introduced in Chapter 2 for DCA models, is based on the comparison with experimental deep mutational scanning (DMS) data, where the fitness effects of large numbers of mutations are measured in parallel. The predictive score of a model is obtained by correlating its statistical energy changes with the experimental fitness values.

Similarly to the definition in Equation 2.50, for FeatureDCA the mutational effect of a substitution $a_i \rightarrow b_i$ in a sequence $\mathbf{a}$ is estimated by the log-likelihood difference

$$\Delta E(a_i \rightarrow b_i) = \log \frac{P(a_1, \dots, b_i, \dots, a_L, \tilde{\mathbf{y}})}{P(a_1, \dots, b_i, \dots, a_L, \mathbf{y})} \quad , \quad (6.19)$$

where $\mathbf{y}$ is the feature vector associated with the wild-type sequence and $\tilde{\mathbf{y}}$ that to the generate one. A negative value of ΔE indicates that the mutation is disfavored by the model, while positive values suggest that it is tolerated or beneficial. As in the case of ArDCA and bmDCA, model predictions are then compared to experimental DMS measurements by computing rank correlations, typically reported as Spearman coefficients. This task provides a stringent test of whether the conditioning introduced in FeatureDCA preserves the local evolutionary signals necessary for functional inference, complementing the global and structural benchmarks described above.

6.4 Results

In this section, we present the results of the evaluation tasks introduced above. The analysis is organized to cover complementary aspects of the model: first, the ability of FeatureDCA to reproduce the global statistical properties of protein families;

second, its performance in conditional generation at specific points of the feature space; third, the structural consistency of sequences generated under conditioning, illustrated through a case study on the Response Regulator family (PF00072); and finally, its predictive power on mutational effect data from deep mutational scanning experiments. Together, these results provide a comprehensive assessment of FeatureDCA as a generative and controllable model for protein sequence families.

6.4.1 Global generativity

As a first set of results, we examine the global generative performance of FeatureDCA, using bmDCA and ArDCA as reference models representing the state-of-the-art DCA implementation in terms of accuracy and efficiency, respectively.

For the beta-lactamase family PF13354, Figure 6.1A compares the projection of the natural MSA onto the first two principal components with those obtained from MSAs sampled by bmDCA, ArDCA, and FeatureDCA. In the case of FeatureDCA, each synthetic sequence is generated conditionally on the feature vector of a natural sequence, so that the artificial and natural alignments share the same empirical distribution of PCs. When trained with $d = 2$ principal components, FeatureDCA already reproduces a distribution in PCA space that is both qualitatively and quantitatively indistinguishable from that of bmDCA.

Figure 6.1B reports the Sinkhorn divergence between the natural and generated datasets projected in d -dimensional PC space. As expected, FeatureDCA matches the first d components included during training, resulting in a close overlap of the distributions, while bmDCA and ArDCA reproduce reliably only the lowest components and fail to capture higher-order variability.

Agreement at the level of pairwise statistics is shown in Figure 6.1C, where the Pearson correlation of connected correlations increases steadily with d for FeatureDCA, largely surpassing arDCA. When the number of principal components approaches $d_{\max} = 2525$, which corresponds to the number of PCs explaining 99% of the variance of the MSA, the model enters an overfitting regime. In this limit, FeatureDCA ceases to generalize and instead memorizes the training set, producing sequences that are nearly indistinguishable from the natural ones. This behavior is reflected by the collapse of Hamming distances between generated and training sequences toward zero, as shown in the following section.

Finally, Figure 6.1D shows the behavior of the effective depth M_{eff} , defined in Chapter 2 as the number of effectively unique sequences, as a function of the number of PCs used during training. At intermediate dimensionalities ($d \simeq 32\text{--}128$ for PF13354), the model achieves a favorable balance between fidelity to the natural statistics and diversity of the generated ensemble, confirming that FeatureDCA preserves the core generative strengths of autoregressive Potts models while introducing controllability.

6.4.2 In-place generativity

Beyond matching the global statistics of the natural sequences, the distinctive characteristic of FeatureDCA is to generate sequences localized around chosen points in feature space. For the case of the PC space, this is achieved by conditioning the model on the principal component coordinates of a given wild-type sequence $\mathbf{y}^{wt}$ and sampling from $P(\mathbf{a} \mid \mathbf{y}^{wt})$. The resulting ensembles of artificial sequences, once projected back onto the PC space, form clusters around their targets. The performance of the in-place generativity can be evaluated by standard sequence-level metrics.

Figure 6.2 summarizes the behavior of samples conditioned on three different wild types from PF13354 (beta-lactamase). At low dimensionality, when the model is trained with only a few principal components, generated sequences spread broadly around the wild type, with average Hamming distances in the range of 60–80%. As the number of components increases, the generated clusters contract toward their respective conditioning points, and their variance decreases. At the same time, the average Hamming distance between ensembles conditioned on different wild types approaches the true distance between the wild types themselves, showing that FeatureDCA preserves the separation of distinct variants in feature space. When the dimensionality approaches d_{max} , however, the model overfits: generated sequences collapse onto their conditioning wild types, and Hamming distances go to zero. An intermediate range, typically $d \simeq 32\text{--}128$, offers the most favorable trade-off, yielding ensembles centered on the conditioning point while maintaining diversity.

Beyond sequence-level metrics, in-place generativity can also be probed structurally by folding generated sequences and comparing them with experimental structures. Using prediction tools such as AlphaFold 3 (AF) and ESMFold (ESM),

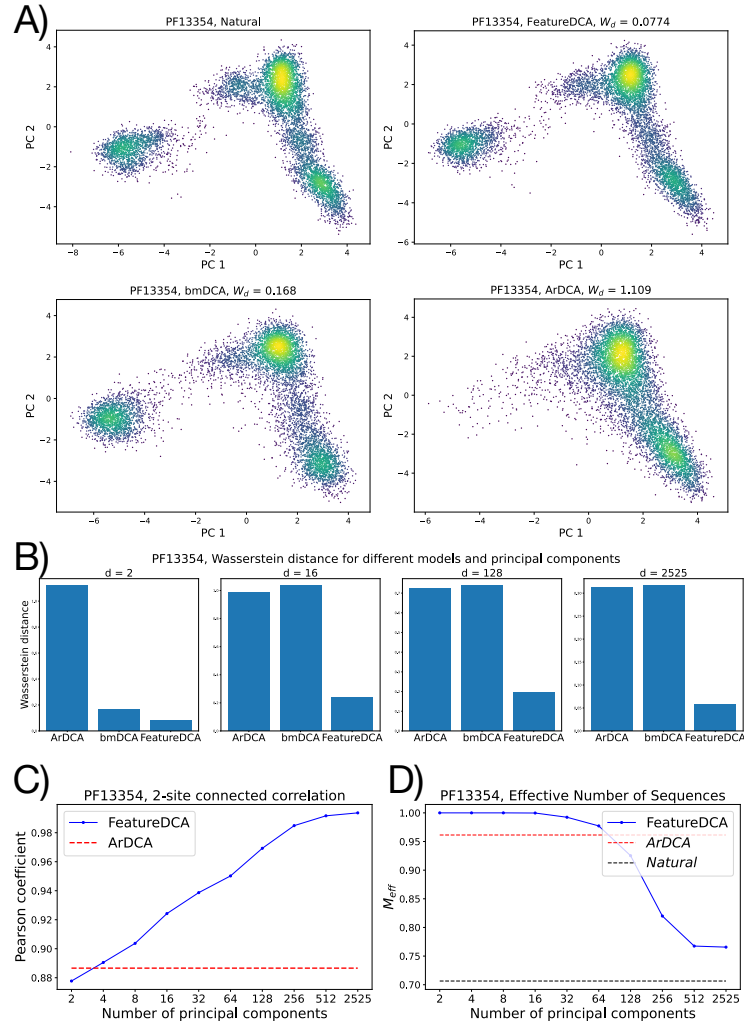


Fig. 6.1 Panel with global statistics. **A)** Projection onto the first two principal components of natural and generated MSAs from protein family PF13354. **B)** Wasserstein distance between the d -dimensional PCA distributions of the natural and generated MSA for different models. FeatureDCA was trained with the corresponding number of principal components. **C)** Pearson coefficient of the 2-site connected correlation between natural and generated data as a function of the number of principal components learned during training. **D)** Effective depth percentage of the generated dataset as a function of the number of principal components learned during training. Source [43]

we folded sequences generated from wild types with known PDB entries and quantified their similarity via the root-mean-square deviation (RMSD) of atomic positions. This type of analysis is nevertheless constrained by the accuracy of AF and ESM in distinguishing structurally divergent amino-acid sequences that belong to the same

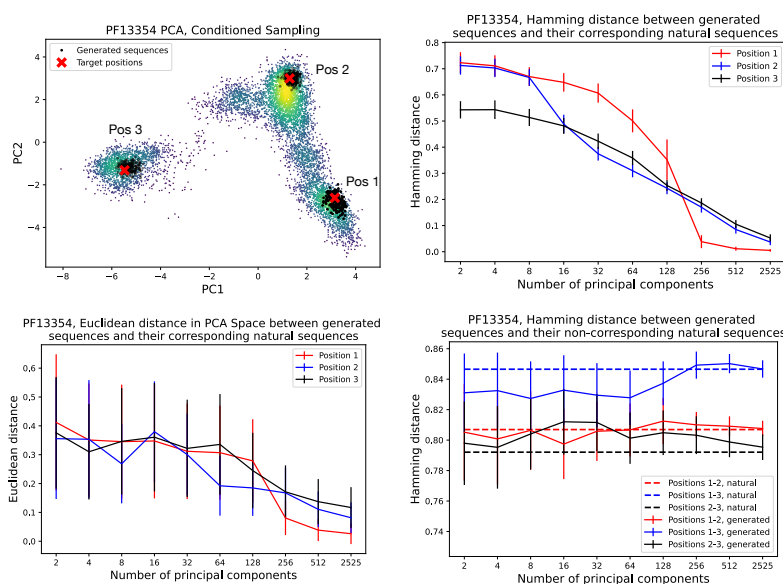


Fig. 6.2 In-place generativity for PF13354. **Top left:** red crosses mark three conditioning points in PCA space, with black dots showing sequences generated by FeatureDCA around them. **Bottom left:** Euclidean distances in the first two PC coordinates between generated sequences and their targets. **Top right:** average Hamming distance between generated sequences and their conditioning wild types. **Bottom right:** average Hamming distance between generated sequences and non-corresponding wild types. Source [43]

protein family. In certain cases, meaningful evaluation of FeatureDCA is not possible because the structure predictors at the base of this analysis fail to recover the distinct folds of aligned homologs. Figure 6.3 shows two illustrative examples: for PF00014, the RMSD between the predicted structures of the two wild types (blue dotted line) does not reflect the one that can be measured from the corresponding experimental structures (red dotted line), so that sequences generated from either wild type are folded into the same conformation as wild type 1. In contrast, for PF00076 the predictions provided by ESM are consistent with the crystal structures and capture the conformational diversity of the family. In this case, FeatureDCA-generated sequences fold into structures that remain close to their respective wild types.

6.4.3 The case of the Response Regulator family (PF00072)

To test the generative and controllable capabilities of FeatureDCA on a biologically and structurally diverse system, we considered the bacterial Response Regulator (RR)

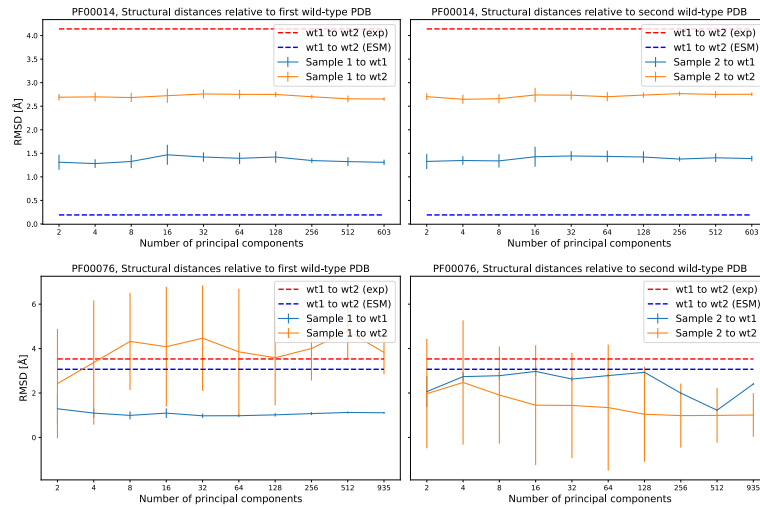


Fig. 6.3 Structural validation of in-place generativity. RMSD between predicted structures of generated sequences and their wild-type references in two protein families. **Top:** PF00014. **Bottom:** PF00076. **Left:** RMSD relative to the first wild type (wt1). **Right:** RMSD relative to the second wild type (wt2). Red dashed lines: RMSD between experimental wild-type structures. Blue dashed lines: RMSD between AF/ESM predictions of wild types. Solid curves: average RMSD of generated samples, with error bars showing standard deviation. FeatureDCA-generated sequences recover structural similarity in PF00076, but structural diversity is not captured in PF00014 due to limitations of current predictors. Source [43]

family (Pfam ID PF00072). This family presents the characteristic to be an ideal benchmark: it is large, well-characterized, and composed of members that share a conserved structural core but differ markedly in their quaternary arrangements. Most importantly, the dataset is sufficiently rich to allow accurate structural prediction across its subfamilies using current folding models such as AlphaFold (AF) and ESMFold (ESM).

RR proteins function as part of the two-component signal transduction systems that regulate bacterial responses to environmental stimuli. In these systems, a sensor histidine kinase (HK) detects an external signal, undergoes autophosphorylation, and transfers the phosphate group to its RR partner. Phosphorylation activates the RR by inducing a conformational change that typically modulates DNA binding and transcriptional regulation. While all RRs share a highly conserved receiver domain (PF00072), their DNA-binding domains vary, giving rise to distinct architectures and dimerization mechanisms that have been extensively studied in structural biology. Among the sequences in PF00072, three major subclasses are particularly well

defined, corresponding to proteins that form distinct homodimers upon activation. Each subclass combines the receiver domain with a characteristic DNA-binding domain: Trans_Reg_C (PF00486), GerE (PF00196), or LytTR (PF04397). These combinations give rise to markedly different dimerization geometries and are represented by a range of experimentally characterized structures. Figure 6.4 presents

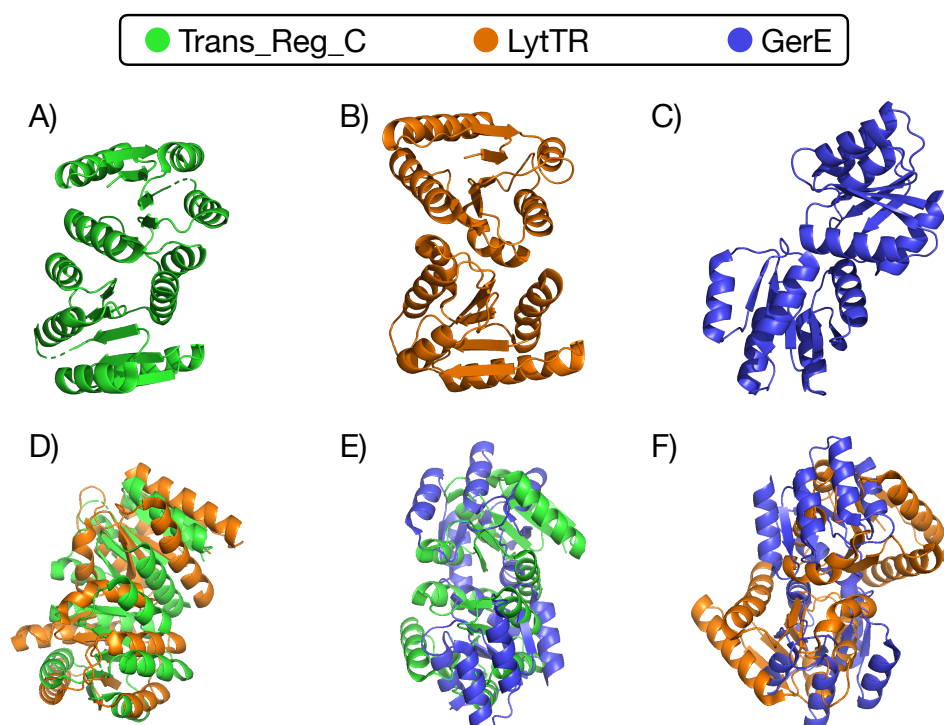


Fig. 6.4 Representative homodimer structures of Response Regulator (RR) subclasses with distinct dimerization modes. A–C) Experimentally determined homodimer structures containing the DNA-binding domains Trans_Reg_C (A, PDB ID: 1NXS), LytTR (B, PDB ID: 4CBV), and GerE (C, PDB ID: 4ZMS). D–F) Pairwise structural alignments between these dimers were performed with PyMOL's align function. The root-mean-square deviations (RMSDs) quantify the degree of structural divergence: 8.6 Å between Trans_Reg_C and LytTR (D), 16.1 Å between Trans_Reg_C and GerE (E), and 18.7 Å between LytTR and GerE (F), highlighting the pronounced differences in dimer geometry despite the conserved receiver domain. Source [43]

representative homodimer structures of the three subclasses (A-C) and their pairwise alignments (D-F), which confirm substantial divergence in dimer arrangement. The corresponding PDB entries are 1NXS (Trans_Reg_C), 4CBV (LytTR), and 4ZMS (GerE), while their pairwise RMSDs are 8.6 Å between Trans_Reg_C and LytTR

(D), 16.1 Å between Trans_Reg_C and GerE (E), and 18.7 Å between LytTR and GerE (F).

To evaluate the generativity capabilities of FeatureDCA, we first verified that AF and ESM can reproduce the experimentally determined structures of natural sequences from each subclass. From approximately 1.7 million UniProt sequences annotated with PF00072, we extracted 264,685 non-redundant entries with architectures consistent with one of the three subclasses: 160,585 for Trans_Reg_C, 69,401 for GerE, and 34,699 for LytTR. A multiple sequence alignment of 118 positions was built, slightly extending the 111-position HMM profile available in Pfam (now integrated in InterPro), to include sites contributing to the structural interfaces. Folding these aligned natural sequences with AlphaFold 3 produced structures highly consistent with the crystallographic templates: the RMSDs between predicted and experimental dimers were 0.35 Å for 1NXS, 1.1 Å for 4CBV, and 1.3 Å for 4ZMS. These values, within the resolution of the experimental data (1.5–4 Å), confirm that the alignment captures the relevant structural determinants for all three subclasses. Another interesting property of the dataset is that, when projected onto the first two principal components of the MSA, the three subclasses form clearly separated clusters corresponding to the Trans_Reg_C, GerE, and LytTR architectures. This separation, shown in Figure 6.5, demonstrates that structural diversity is encoded in sequence space and can, in principle, be captured by a PCA-based generative model such as FeatureDCA.

To evaluate whether FeatureDCA can learn and reproduce this structure–sequence relationship, we performed targeted conditional generation experiments. For each of the three subclasses, we used as conditioning input the PC coordinates of the sequences corresponding to the experimental structures (1NXS, 4ZMS, and 4CBV). The generated sequences were folded with AlphaFold 3 and compared to all three experimental structures using RMSD of atomic coordinates. The resulting trends are summarized in panels A–C in Figure 6.6. As expected, the RMSD between the generated and conditioning structures decreases with increasing dimensionality of the feature space, indicating that the model effectively learns to direct generation toward the intended structural class. For the Trans_Reg_C class (1NXS), which is the most represented in the dataset, FeatureDCA achieves the lowest RMSDs and highest structural consistency. The LytTR class (4CBV), being less represented, is more challenging: at low-dimensional conditioning, its generated sequences tend to adopt conformations closer to the Trans_Reg_C fold, but as the number of principal

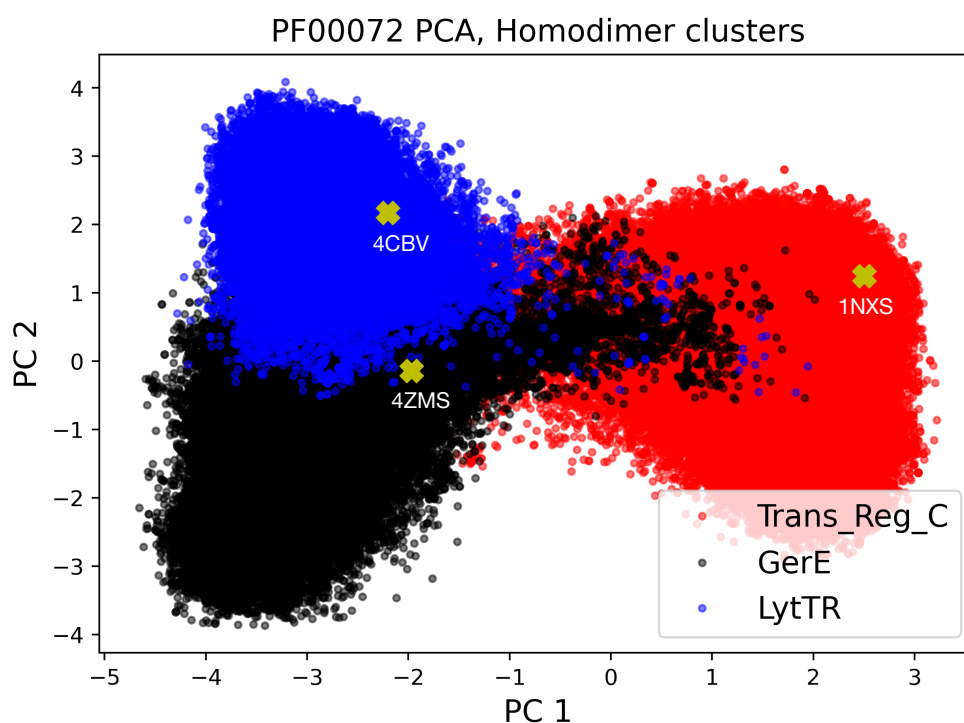


Fig. 6.5 Principal component projection of RR sequences reveals structurally distinct clusters. Projection of the PF00072 alignment onto the first two principal components. The three main clusters correspond to sequences containing the DNA-binding domains Trans_Reg_C (red), LytTR (blue), and GerE (black). Yellow crosses indicate the positions of the experimental PDB structures 1NXS, 4CBV, and 4ZMS, each located within its respective cluster. The clear separation of the three groups in sequence space supports the use of FeatureDCA for class-specific conditional generation. Source [43]

components increases, the model begins to discriminate correctly between the classes. These results suggest that both representation balance and feature dimensionality play critical roles in enabling class-specific control. The lower-right panel (D) in Figure 6.6 reports the fraction of successfully folded sequences, defined as those with RMSD below 10 Å relative to their conditioning structure. Equivalent trends are observed when using the predicted template modeling (pTM) score as a measure of confidence. Sequences generated under well-sampled conditions are more likely to yield accurate and high-confidence folds, reinforcing the conclusion that conditioning enables controlled and structurally meaningful generation.

In summary, the PF00072 case study demonstrates that FeatureDCA, when trained on a sufficiently representative alignment, can generate sequences that retain

RMSD Analysis Between AlphaFold-Predicted and Experimental Structures for Generated Sequences in PF00072 with FeatureDCA

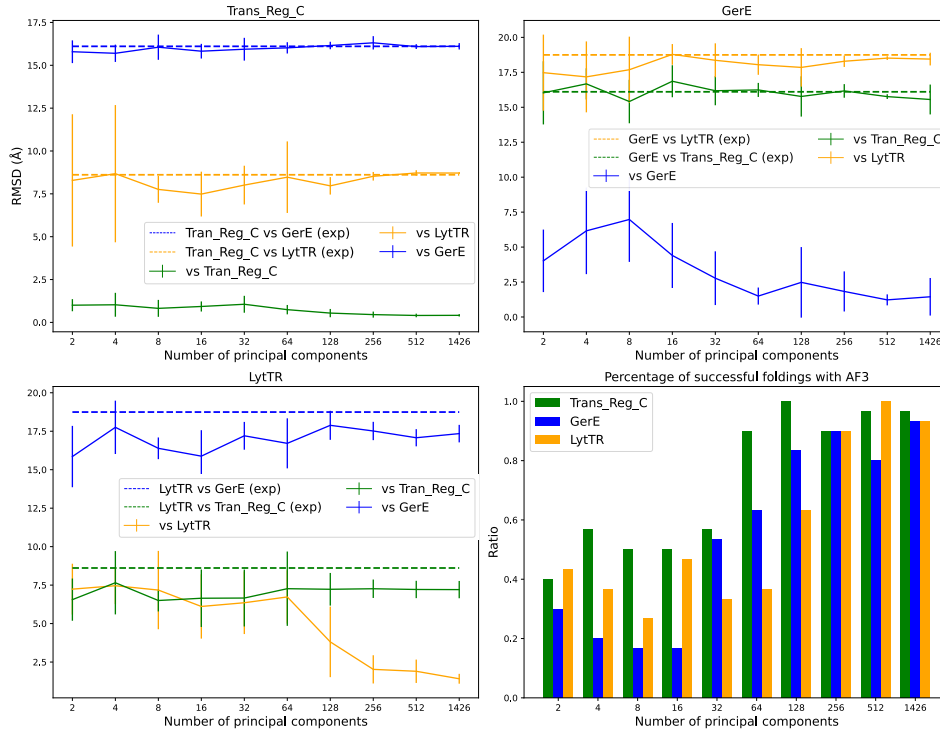


Fig. 6.6 Structural class specificity of FeatureDCA-generated sequences in the RR family (PF00072). **Top-left, top-right, and bottom-left:** RMSD between predicted structures of generated sequences from the Trans_Reg_C (PDB ID: 1NXS), GerE (4ZMS), and LytTR (4CBV) classes and the three experimental references. Each curve represents the average RMSD as a function of the number of principal components used in training and generation. In each plot, the curve corresponding to the conditioning class shows the lowest RMSD, while the others approach the experimental distances between the wild-type structures (dotted lines). **Bottom-right:** fraction of generated sequences whose predicted structures are within 10 Å RMSD of the conditioning reference, indicating successful fold recovery. Together, these analyses confirm that FeatureDCA preserves class-specific structural identity as a function of PC-based conditioning. Source [43]

the structural identity of their conditioning class. The model not only captures global statistical patterns but also learns to reproduce the fine-grained structural distinctions that separate protein subclasses within the same evolutionary family.

6.4.4 In-silico Deep Mutational Scanning

A further test of generative models is their ability to capture the functional impact of point mutations. As discussed in Chapter 2, this task can be addressed through in-silico deep mutational scanning (DMS), where the predicted effect of a single-residue substitution is estimated from the difference in log-likelihood between the wild-type and the mutant sequence. For FeatureDCA, the mutational score defined in Equation 6.19 explicitly accounts for the dependence of both wild type and mutation on their position in feature space.

We evaluated FeatureDCA against ArDCA on the well-characterized beta-lactamase family (PF13354), for which extensive experimental DMS datasets are available. Predictive performance was quantified through the Spearman rank correlation between the model-predicted ΔE values and experimentally measured fitness effects. As summarized in Appendix C.6, FeatureDCA achieves performance comparable to that of ArDCA when trained with a small number of conditioning components. Interestingly, as the dimensionality of the conditioning space increases, the predictive correlation exhibits a non-monotonic trend: the correlation initially decreases, suggesting that conditioning on an excessive number of PCs may overfit or dilute mutational signals. However, at larger values of the number of PCs, the performance begins to recover, indicating that the model eventually re-stabilizes and incorporates additional global variation effectively.

These results demonstrate that the inclusion of conditioning does not compromise the ability of the model to capture local evolutionary constraints. FeatureDCA thus retains the predictive reliability of autoregressive Potts models while extending them with a tunable and interpretable conditioning mechanism that links global sequence variability to local functional effects.

6.5 Conclusion and Perspectives

In this chapter, we introduced FeatureDCA, a principled extension of the Direct Coupling Analysis framework designed to achieve controllable generation of protein sequences within a given family. The model builds directly upon the autoregressive Potts formulation described in Chapter 2, extending it to include explicit dependence on low-dimensional, biologically meaningful features. FeatureDCA, along

with AttentionDCA, provides a practical realization of the broader goal outlined in Chapter 1: developing interpretable, data-efficient models that can emulate certain capabilities of deep generative architectures without losing the statistical transparency characteristic of DCA-based approaches. In this sense, it represents a synthesis of the theoretical and methodological threads developed throughout the thesis.

From a modeling perspective, FeatureDCA introduces a simple yet powerful mechanism to condition the sequence probability distribution on a continuous feature vector, implemented here as the principal components of the training alignment. The Gaussian approximation discussed in Section 6.2 provides the theoretical foundation for coupling discrete sequence variables to continuous conditioning features, while the resulting discrete ansatz preserves the interpretability of the Potts Hamiltonian. This framework allows one to steer the generative process toward selected regions of sequence space in a transparent and principled way.

The results presented in this chapter demonstrate that the addition of conditioning does not compromise the statistical fidelity of the model. FeatureDCA reproduces the empirical correlations and global statistics of natural alignments with accuracy comparable to or exceeding that of established approaches such as bmDCA and ArDCA, while maintaining substantial sequence diversity. More importantly, it enables targeted generation: sequences can be sampled at specific positions in feature space, reproducing both their sequence-level and structural characteristics, as shown for the case of the Response Regulator family PF00072. This study confirmed that conditioning allows FeatureDCA to navigate distinct structural subclasses within the same domain, demonstrating that the relevant structural variability of a protein family can be accessed through interpretable, low-dimensional features. In addition, FeatureDCA retains predictive power for mutational effects derived from deep mutational scanning data, confirming that the inclusion of conditioning does not disrupt the local evolutionary constraints encoded in the model.

Beyond these results, several directions emerge for further development. A natural extension of this work involves exploring alternative conditioning variables that can capture functional, structural, or biophysical properties beyond the principal components. Examples of these include experimentally derived features such as mutational fitness, catalytic activity, or stability, as well as computational metrics describing dynamics, solvent exposure, or binding specificity. Incorporating such quantities would allow FeatureDCA to move from an unsupervised to a semi-

supervised or supervised framework, bridging the gap between purely statistical inference and experimentally guided design. Moreover, one can think of integrating multi-feature learning models, where different layers of biological information act as control parameters. From a methodological point of view, the autoregressive structure of the model provides a flexible basis for further optimization. Future developments could explore alternative inference strategies beyond maximum likelihood, such as variational or contrastive approaches, which may improve training stability and scalability. The Gaussian approximation and the discrete ansatz for the marginal conditional distributions could be generalized to non-linear feature couplings or to other latent-variable formulations, enabling richer relationships between features and sequences. Finally, an important perspective is that of connecting FeatureDCA with recent developments in large-scale protein language models. Hybrid strategies, in which embeddings extracted from transformer models serve as conditioning features, could combine the interpretability of Potts-based inference with the representational richness of deep architectures. Conversely, the explicit feature coupling of FeatureDCA could help interpret neural generative models by providing low-dimensional, physically meaningful constraints.

In summary, FeatureDCA extends the family of DCA-based models toward controllable generativity, introducing a transparent and flexible mechanism to bias sequence generation along biologically interpretable dimensions. It demonstrates that statistical frameworks grounded in evolutionary data can achieve not only accurate and diverse generation but also directed exploration of protein sequence space. This work closes the methodological progression developed throughout the thesis from statistical inference and autoregressive formulation to interpretable and controllable generative modeling, laying the conceptual groundwork for future approaches that unify interpretability and control in computational protein design.

Chapter 7

Conclusion and outlook

In this thesis we discussed the use of statistical physics models for protein sequence analysis, with particular attention to their relation with modern deep learning architectures. Although deeply intertwined, these two classes of models are often developed with different goals and different scientific approaches. On the one hand, statistical physics models usually start from explicit assumptions on the data and define probability distributions with a clear biological or physical interpretation. In the case of protein sequences, for instance, this means writing a Hamiltonian over amino acid configurations and inferring parameters that reproduce the empirical statistics of natural protein families. On the other hand, deep learning architectures are typically built to maximize predictive performance by means of flexible and highly parametrized functions. Their success in computational biology is now undeniable, but their internal representations are often harder to interpret in terms of the physical and biological constraints acting on proteins.

The main direction of this work was to explore the space between these two approaches and to develop models that could compete with deep architectures starting from more principled and biology-inspired assumptions, rather than from architectural complexity alone. The goal was not to oppose statistical physics and deep learning, but to understand whether ideas developed in one framework could be transferred to the other. In particular, we asked whether mechanisms introduced in deep learning could be reformulated inside shallow and interpretable statistical physics models, and whether this reformulation could help clarify what those mechanisms learn from protein sequence data. At the same time, we investigated whether

the structure of statistical physics models could be enriched by ideas coming from modern machine learning, without losing the interpretability and data efficiency that make them useful.

The starting point of the thesis was Direct Coupling Analysis (DCA), a principled framework that models homologous protein sequences as samples from a Boltzmann distribution defined by a Potts Hamiltonian. By imposing constraints on the single- and pair-wise statistics of a Multiple Sequence Alignment, the model infers local fields and pairwise couplings that can be used to extract biological information about the protein family. In this sense, DCA represents a minimal but powerful example of inverse statistical physics applied to biological sequence data. Despite its simple form, which results in a high degree of interpretability, this model is expressive enough to capture co-evolutionary constraints, predict residue contacts, estimate mutational effects, and generate artificial sequences with natural-like statistical properties.

Starting from this framework, we developed AttentionDCA as an attempt to study the attention mechanism used in transformer architectures in a statistical physics setting. Attention mechanisms play a central role in modern protein models, including general protein language models and structure predictors such as AlphaFold. However, in deep architectures, they are embedded inside many layers of nonlinear transformations, making it difficult to isolate their specific contribution. In AttentionDCA, the interaction tensor of a Potts model is written through a factored attention parametrization. This makes it possible to analyze attention maps directly, compare them with standard DCA contact scores, and study the relation between attention-based representations and co-evolutionary couplings.

The results obtained with AttentionDCA show that the attention mechanism can be interpreted, at least in this simplified setting, as a structured parametrization of a Potts model. The model reaches contact-prediction performances comparable to state-of-the-art DCA methods while using a significantly reduced number of parameters. Moreover, the attention matrices themselves contain structural information and can be used to define contact scores. This provides a direct link between an object that is central in deep learning and an object that has a clear meaning in inverse statistical physics. In this sense, AttentionDCA is not only a new DCA variant, but also a tool to make the attention mechanism more transparent in the context of protein sequence modeling.

The second development of the thesis was CLAPP, which moved the focus from single protein families to protein-protein interaction specificity and paralog matching. CLAPP was built on the idea that biological information at the level of classes can be used to reduce the complexity of the matching problem. Instead of relying only on a generic interaction score, the method uses class-level organization to pool information and formulate a more efficient inference strategy. This makes CLAPP an example of how biological insight can be used to design a model that is both simpler and more effective. Its relevance within the thesis is therefore methodological as much as practical: it shows that good modeling does not necessarily require a larger architecture, but often a better use of the structure already present in the biological problem.

Finally, we introduced FeatureDCA, extending autoregressive DCA into a controllable generative model. Standard generative DCA methods can sample sequences from the learned distribution of a protein family, but they offer limited control over the region of sequence space that is explored. FeatureDCA addresses this limitation by conditioning the autoregressive probability on external features, which in this thesis were primarily defined by principal components of the Multiple Sequence Alignment. In this way, the model can generate sequences around specific regions of the protein family while preserving the statistical and structural constraints learned from natural sequences. FeatureDCA therefore completes the methodological path of the thesis by showing that shallow statistical physics models can also be adapted to modern generative-design tasks, while maintaining an explicit and interpretable structure.

Taken together, AttentionDCA, CLAPP, and FeatureDCA define a coherent path. AttentionDCA shows that the attention mechanism can be interpreted as a structured Potts-like parametrization and used for contact prediction. CLAPP shows that biological structure at the level of classes can be used to formulate an efficient model for interaction specificity and paralog matching. FeatureDCA shows that shallow statistical models can be made controllable without abandoning the interpretability and efficiency of the DCA framework. The common thread is not the rejection of deep learning, but the search for the minimal architecture that is appropriate to the biological question. In each case, the model is designed by asking which constraints are known, which degrees of freedom should be learned, and which assumptions are scientifically meaningful.

Several limitations remain open. The Potts framework is fundamentally based on pairwise interactions. This approximation has proved remarkably effective, but it cannot be expected to capture every aspect of the protein functional landscape. Higher-order constraints, collective effects, or latent variables may become relevant in specific families or design tasks. Future work should therefore clarify when pairwise models are sufficient, when additional complexity is needed, and how this complexity can be introduced without losing interpretability or overfitting limited data. A second limitation concerns controllable generation and validation. A generative model can reproduce sequence statistics and still fail to produce proteins that are stable, functional, or experimentally viable. Future developments of FeatureDCA-like models should therefore integrate additional forms of conditioning and validation, moving beyond PCA-based conditioning toward experimentally or biologically defined features such as fitness measurements, stability, binding specificity, phylogenetic labels, structural descriptors, or functional annotations. This would turn the model from a tool for sampling around specific regions of sequence space into a more general framework for constrained protein design. Since the conditioning variables enter the model explicitly, their effect on amino-acid preferences could be inspected directly, preserving interpretability as one of the main advantages of the statistical physics approach.

The broader perspective emerging from this thesis is that model complexity should be treated as a scientific and practical resource, not as an intrinsic measure of scientific progress. Deep protein language models and large structure predictors have transformed computational biology by providing general-purpose representations and impressive predictive accuracy. However, their scale also comes with costs: computational cost, energy consumption, difficulty of interpretation, dependence on massive training datasets, and limited accessibility for groups without large computational infrastructure. In many cases, these costs are justified by the difficulty of the task. In others, a smaller, transparent, and data-efficient model may be the more appropriate choice.

In conclusion, the future of protein modeling should not be understood only as a race toward larger architectures. It should also be a search for the right model at the right scale. For some tasks, this will require deep networks trained on massive datasets. For others, it may require a carefully designed Hamiltonian, a biologically informed grouping of sequences, or a low-dimensional set of interpretable features. The work presented in this thesis supports the view that shallow statistical physics

models, far from being obsolete, remain powerful and responsible tools for protein sequence analysis and design. In a field increasingly driven by scale, this form of parsimony is not a limitation, but a scientific strength.

References

- [1] B. Alberts et al. *Molecular Biology of the Cell*. 4th. Garland Science, 2002. ISBN: 978-0-8153-3218-3 978-0-8153-4072-0.
- [2] G. A. Petsko and D. Ringe. *Protein Structure and Function*. New Science Press, 2004. ISBN: 978-1-4051-1922-1.
- [3] F. Crick. “Central Dogma of Molecular Biology”. In: *Nature* 227.5258 (Aug. 1970), pp. 561–563. ISSN: 1476-4687. DOI: [10.1038/227561a0](https://doi.org/10.1038/227561a0).
- [4] *Essential Amino Acids: Chart, Abbreviations and Structure*.
- [5] L. Pauling and R. B. Corey. “The Pleated Sheet, A New Layer Configuration of Polypeptide Chains”. In: *Proceedings of the National Academy of Sciences* 37.5 (May 1951), pp. 251–256. DOI: [10.1073/pnas.37.5.251](https://doi.org/10.1073/pnas.37.5.251).
- [6] L. Pauling, R. B. Corey, and H. R. Branson. “The Structure of Proteins: Two Hydrogen-Bonded Helical Configurations of the Polypeptide Chain”. In: *Proceedings of the National Academy of Sciences* 37.4 (Apr. 1951), pp. 205–211. DOI: [10.1073/pnas.37.4.205](https://doi.org/10.1073/pnas.37.4.205).
- [7] D. Eisenberg. “The Discovery of the α -Helix and β -Sheet, the Principal Structural Features of Proteins”. In: *Proceedings of the National Academy of Sciences* 100.20 (Sept. 2003), pp. 11207–11210. DOI: [10.1073/pnas.2034522100](https://doi.org/10.1073/pnas.2034522100).
- [8] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. “Stereochemistry of Polypeptide Chain Configurations”. In: *Journal of Molecular Biology* 7 (July 1963), pp. 95–99. ISSN: 0022-2836. DOI: [10.1016/s0022-2836\(63\)80023-6](https://doi.org/10.1016/s0022-2836(63)80023-6).
- [9] J. Hardy and D. J. Selkoe. “The Amyloid Hypothesis of Alzheimer’s Disease: Progress and Problems on the Road to Therapeutics”. In: *Science* 297.5580 (July 2002), pp. 353–356. DOI: [10.1126/science.1072994](https://doi.org/10.1126/science.1072994).
- [10] F. Chiti and C. M. Dobson. “Protein Misfolding, Functional Amyloid, and Human Disease”. In: *Annual Review of Biochemistry* 75 (2006), pp. 333–366. ISSN: 0066-4154. DOI: [10.1146/annurev.biochem.75.101304.123901](https://doi.org/10.1146/annurev.biochem.75.101304.123901).
- [11] C. Soto and S. Pritzkow. “Protein Misfolding, Aggregation, and Conformational Strains in Neurodegenerative Diseases”. In: *Nature Neuroscience* 21.10 (Oct. 2018), pp. 1332–1340. ISSN: 1546-1726. DOI: [10.1038/s41593-018-0235-9](https://doi.org/10.1038/s41593-018-0235-9).

- [12] S. H. White. “Biophysical Dissection of Membrane Proteins”. In: *Nature* 459.7245 (May 2009), pp. 344–346. ISSN: 1476-4687. DOI: [10.1038/nature08142](https://doi.org/10.1038/nature08142).
- [13] R. M. Bill et al. “Overcoming Barriers to Membrane Protein Structure Determination”. In: *Nature Biotechnology* 29.4 (Apr. 2011), pp. 335–340. ISSN: 1546-1696. DOI: [10.1038/nbt.1833](https://doi.org/10.1038/nbt.1833).
- [14] K. A. Dill and J. L. MacCallum. “The Protein-Folding Problem, 50 Years On”. In: *Science* 338.6110 (Nov. 2012), pp. 1042–1046. DOI: [10.1126/science.1219021](https://doi.org/10.1126/science.1219021).
- [15] C. Levinthal. “Are There Pathways for Protein Folding?” In: *Journal de Chimie Physique* 65 (1968), pp. 44–45. ISSN: 0021-7689. DOI: [10.1051/jcp/1968650044](https://doi.org/10.1051/jcp/1968650044).
- [16] J. N. Onuchic and P. G. Wolynes. “Theory of Protein Folding”. In: *Current Opinion in Structural Biology* 14.1 (Feb. 2004), pp. 70–75. ISSN: 0959-440X. DOI: [10.1016/j.sbi.2004.01.009](https://doi.org/10.1016/j.sbi.2004.01.009).
- [17] K. A. Dill et al. “The Protein Folding Problem”. In: *Annual Review of Biophysics* 37. Volume 37, 2008 (June 2008), pp. 289–316. ISSN: 1936-122X, 1936-1238. DOI: [10.1146/annurev.biophys.37.092707.153558](https://doi.org/10.1146/annurev.biophys.37.092707.153558).
- [18] A. W. Senior et al. “Improved Protein Structure Prediction Using Potentials from Deep Learning”. In: *Nature* 577.7792 (Jan. 2020), pp. 706–710. ISSN: 1476-4687. DOI: [10.1038/s41586-019-1923-7](https://doi.org/10.1038/s41586-019-1923-7).
- [19] J. Jumper et al. “Highly Accurate Protein Structure Prediction with AlphaFold”. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: [10.1038/s41586-021-03819-2](https://doi.org/10.1038/s41586-021-03819-2).
- [20] E. T. Jaynes. “Information Theory and Statistical Mechanics”. In: *Physical Review* 106.4 (1957), pp. 620–630. DOI: [10.1103/PhysRev.106.620](https://doi.org/10.1103/PhysRev.106.620).
- [21] F. Morcos et al. “Direct-Coupling Analysis of Residue Coevolution Captures Native Contacts across Many Protein Families”. In: *Proceedings of the National Academy of Sciences* 108.49 (Dec. 2011), E1293–E1301. DOI: [10.1073/pnas.1111471108](https://doi.org/10.1073/pnas.1111471108).
- [22] M. Weigt et al. “Identification of Direct Residue Contacts in Protein–Protein Interaction by Message Passing”. In: *Proceedings of the National Academy of Sciences* 106.1 (Jan. 2009), pp. 67–72. DOI: [10.1073/pnas.0805923106](https://doi.org/10.1073/pnas.0805923106).
- [23] S. Cocco et al. “Inverse Statistical Physics of Protein Sequences: A Key Issues Review”. In: *Reports on Progress in Physics* 81.3 (Jan. 2018), p. 032601. ISSN: 0034-4885. DOI: [10.1088/1361-6633/aa9965](https://doi.org/10.1088/1361-6633/aa9965).
- [24] A. P. Muntoni et al. “adabmDCA: Adaptive Boltzmann Machine Learning for Biological Sequences”. In: *BMC Bioinformatics* 22.1 (Oct. 2021), p. 528. ISSN: 1471-2105. DOI: [10.1186/s12859-021-04441-9](https://doi.org/10.1186/s12859-021-04441-9).
- [25] W. P. Russ et al. “An Evolution-Based Model for Designing Chorismate Mutase Enzymes”. In: *Science* 369.6502 (July 2020), pp. 440–445. DOI: [10.1126/science.aba3304](https://doi.org/10.1126/science.aba3304).

- [26] J. Trinquier et al. “Efficient Generative Modeling of Protein Sequences Using Simple Autoregressive Models”. In: *Nature Communications* 12.1 (Oct. 2021), p. 5800. ISSN: 2041-1723. DOI: [10.1038/s41467-021-25756-4](https://doi.org/10.1038/s41467-021-25756-4).
- [27] D. T. Jones et al. “PSICOV: Precise Structural Contact Prediction Using Sparse Inverse Covariance Estimation on Large Multiple Sequence Alignments”. In: *Bioinformatics* 28.2 (Jan. 2012), pp. 184–190. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btr638](https://doi.org/10.1093/bioinformatics/btr638).
- [28] D. T. Jones and S. M. Kandathil. “High Precision in Protein Contact Prediction Using Fully Convolutional Neural Networks and Minimal Sequence Features”. In: *Bioinformatics* 34.19 (Oct. 2018), pp. 3308–3315. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/bty341](https://doi.org/10.1093/bioinformatics/bty341).
- [29] J. Xu. “Distance-Based Protein Folding Powered by Deep Learning”. In: *Proceedings of the National Academy of Sciences* 116.34 (Aug. 2019), pp. 16856–16865. DOI: [10.1073/pnas.1821309116](https://doi.org/10.1073/pnas.1821309116).
- [30] Y. Li et al. “Deducing High-Accuracy Protein Contact-Maps from a Triplet of Coevolutionary Matrices through Deep Residual Convolutional Networks”. In: *PLOS Computational Biology* 17.3 (Mar. 2021), e1008865. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1008865](https://doi.org/10.1371/journal.pcbi.1008865).
- [31] K. He et al. *Deep Residual Learning for Image Recognition*. Dec. 2015. DOI: [10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385).
- [32] A. J. Riesselman, J. B. Ingraham, and D. S. Marks. “Deep Generative Models of Genetic Variation Capture the Effects of Mutations”. In: *Nature Methods* 15.10 (Oct. 2018), pp. 816–822. ISSN: 1548-7105. DOI: [10.1038/s41592-018-0138-4](https://doi.org/10.1038/s41592-018-0138-4).
- [33] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. “A Learning Algorithm for Boltzmann Machines”. In: *Cognitive Science* 9.1 (Jan. 1985), pp. 147–169. ISSN: 0364-0213. DOI: [10.1016/S0364-0213\(85\)80012-4](https://doi.org/10.1016/S0364-0213(85)80012-4).
- [34] J. Tubiana, S. Cocco, and R. Monasson. “Learning Protein Constitutive Motifs from Sequence Data”. In: *eLife* 8 (Mar. 2019). Ed. by L. J. Colwell and D. Weigel, e39397. ISSN: 2050-084X. DOI: [10.7554/eLife.39397](https://doi.org/10.7554/eLife.39397).
- [35] E. C. Alley et al. “Unified Rational Protein Engineering with Sequence-Based Deep Representation Learning”. In: *Nature Methods* 16.12 (Dec. 2019), pp. 1315–1322. ISSN: 1548-7105. DOI: [10.1038/s41592-019-0598-1](https://doi.org/10.1038/s41592-019-0598-1).
- [36] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [37] R. Rao et al. “Evaluating Protein Transfer Learning with TAPE”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.
- [38] Z. Lin et al. “Evolutionary-Scale Prediction of Atomic-Level Protein Structure with a Language Model”. In: *Science* 379.6637 (Mar. 2023), pp. 1123–1130. DOI: [10.1126/science.ade2574](https://doi.org/10.1126/science.ade2574).

- [39] R. M. Rao et al. “MSA Transformer”. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, July 2021, pp. 8844–8856.
- [40] A. Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.
- [41] D. Bahdanau, K. Cho, and Y. Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. May 2016. DOI: [10.48550/arXiv.1409.0473](https://doi.org/10.48550/arXiv.1409.0473).
- [42] F. Caredda and A. Pagnani. “Direct Coupling Analysis and the Attention Mechanism”. In: *BMC Bioinformatics* 26.1 (Feb. 2025), p. 41. ISSN: 1471-2105. DOI: [10.1186/s12859-025-06062-y](https://doi.org/10.1186/s12859-025-06062-y).
- [43] F. Caredda et al. *Controllable Protein Design via Autoregressive Direct Coupling Analysis Conditioned on Principal Components*. Aug. 2025. DOI: [10.1101/2025.08.18.669886](https://doi.org/10.1101/2025.08.18.669886).
- [44] J. D. Bloom, A. Raval, and C. O. Wilke. “Thermodynamics of Neutral Protein Evolution”. In: *Genetics* 175.1 (Jan. 2007), pp. 255–266. ISSN: 0016-6731. DOI: [10.1534/genetics.106.061754](https://doi.org/10.1534/genetics.106.061754).
- [45] N. Halabi et al. “Protein Sectors: Evolutionary Units of Three-Dimensional Structure”. In: *Cell* 138.4 (Aug. 2009), pp. 774–786. ISSN: 1097-4172. DOI: [10.1016/j.cell.2009.07.038](https://doi.org/10.1016/j.cell.2009.07.038).
- [46] J. Echave, S. J. Spielman, and C. O. Wilke. “Causes of Evolutionary Rate Variation among Protein Sites”. In: *Nature Reviews Genetics* 17.2 (Feb. 2016), pp. 109–121. ISSN: 1471-0064. DOI: [10.1038/nrg.2015.18](https://doi.org/10.1038/nrg.2015.18).
- [47] J. Mistry et al. “Pfam: The Protein Families Database in 2021”. In: *Nucleic Acids Research* 49.D1 (Jan. 2021), pp. D412–D419. ISSN: 1362-4962. DOI: [10.1093/nar/gkaa913](https://doi.org/10.1093/nar/gkaa913).
- [48] M. Blum et al. “InterPro: The Protein Sequence Classification Resource in 2025”. In: *Nucleic Acids Research* 53.D1 (Nov. 2024), pp. D444–D456. ISSN: 0305-1048. DOI: [10.1093/nar/gkae1082](https://doi.org/10.1093/nar/gkae1082).
- [49] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge: Cambridge University Press, 1998. ISBN: 978-0-521-62971-3. DOI: [10.1017/CBO9780511790492](https://doi.org/10.1017/CBO9780511790492).
- [50] A. Krogh et al. “Hidden Markov Models in Computational Biology. Applications to Protein Modeling”. In: *Journal of Molecular Biology* 235.5 (Feb. 1994), pp. 1501–1531. ISSN: 0022-2836. DOI: [10.1006/jmbi.1994.1104](https://doi.org/10.1006/jmbi.1994.1104).
- [51] D. W. Mount. “Using Hidden Markov Models to Align Multiple Sequences”. In: *Cold Spring Harbor Protocols* 2009.7 (Jan. 2009), pdb.top41. ISSN: 1940-3402, 1559-6095. DOI: [10.1101/pdb.top41](https://doi.org/10.1101/pdb.top41).
- [52] S. R. Eddy. “Accelerated Profile HMM Searches”. In: *PLOS Computational Biology* 7.10 (Oct. 2011), e1002195. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1002195](https://doi.org/10.1371/journal.pcbi.1002195).

- [53] T. Plefka. “Convergence Condition of the TAP Equation for the Infinite-Ranged Ising Spin Glass Model”. In: *Journal of Physics A: Mathematical and General* 15.6 (June 1982), p. 1971. ISSN: 0305-4470. DOI: [10.1088/0305-4470/15/6/035](https://doi.org/10.1088/0305-4470/15/6/035).
- [54] C. Baldassi et al. “Fast and Accurate Multivariate Gaussian Modeling of Protein Families: Predicting Residue Contacts and Protein-Interaction Partners”. In: *PLOS ONE* 9.3 (Mar. 2014), e92721. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0092721](https://doi.org/10.1371/journal.pone.0092721).
- [55] M. Figliuzzi, P. Barrat-Charlaix, and M. Weigt. “How Pairwise Coevolutionary Models Capture the Collective Residue Variability in Proteins?” In: *Molecular Biology and Evolution* 35.4 (Apr. 2018), pp. 1018–1027. ISSN: 0737-4038. DOI: [10.1093/molbev/msy007](https://doi.org/10.1093/molbev/msy007).
- [56] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).
- [57] M. Ekeberg. “Improved Contact Prediction in Proteins: Using Pseudolikelihoods to Infer Potts Models”. In: *Physical Review E* 87.1 (2013). DOI: [10.1103/PhysRevE.87.012707](https://doi.org/10.1103/PhysRevE.87.012707).
- [58] H. B. Callen. “Green Function Theory of Ferromagnetism”. In: *Physical Review* 130.3 (May 1963), pp. 890–898. DOI: [10.1103/PhysRev.130.890](https://doi.org/10.1103/PhysRev.130.890).
- [59] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. “High-Dimensional Ising Model Selection Using ℓ_1 -Regularized Logistic Regression”. In: *The Annals of Statistics* 38.3 (June 2010), pp. 1287–1319. ISSN: 0090-5364, 2168-8966. DOI: [10.1214/09-AOS691](https://doi.org/10.1214/09-AOS691).
- [60] S. Balakrishnan et al. “Learning Generative Models for Protein Fold Families”. In: *Proteins* 79.4 (Apr. 2011), pp. 1061–1078. ISSN: 1097-0134. DOI: [10.1002/prot.22934](https://doi.org/10.1002/prot.22934).
- [61] H. Jacquier et al. “Capturing the Mutational Landscape of the Beta-Lactamase TEM-1”. In: *Proceedings of the National Academy of Sciences* 110.32 (Aug. 2013), pp. 13067–13072. DOI: [10.1073/pnas.1215206110](https://doi.org/10.1073/pnas.1215206110).
- [62] C. E. Gonzalez, P. Roberts, and M. Ostermeier. “Fitness Effects of Single Amino Acid Insertions and Deletions in TEM-1 β -Lactamase”. In: *Journal of Molecular Biology* 431.12 (May 2019), pp. 2320–2330. ISSN: 0022-2836. DOI: [10.1016/j.jmb.2019.04.030](https://doi.org/10.1016/j.jmb.2019.04.030).
- [63] E. Firnberg et al. “A Comprehensive, High-Resolution Map of a Gene’s Fitness Landscape”. In: *Molecular Biology and Evolution* 31.6 (June 2014), pp. 1581–1592. ISSN: 0737-4038. DOI: [10.1093/molbev/msu081](https://doi.org/10.1093/molbev/msu081).
- [64] L. Sesta et al. “Inference of Annealed Protein Fitness Landscapes with AnnealDCA”. In: *PLOS Computational Biology* 20.2 (Feb. 2024), e1011812. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1011812](https://doi.org/10.1371/journal.pcbi.1011812).
- [65] M. De Leonardis et al. “Unsupervised Modeling of Mutational Landscapes of Adeno-Associated Viruses Viability”. In: *BMC Bioinformatics* 25.1 (July 2024), p. 229. ISSN: 1471-2105. DOI: [10.1186/s12859-024-05823-5](https://doi.org/10.1186/s12859-024-05823-5).

- [66] A. Procaccini et al. “Dissecting the Specificity of Protein-Protein Interaction in Bacterial Two-Component Signaling: Orphans and Crosstalks”. In: *PLOS ONE* 6.5 (May 2011), e19729. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0019729](https://doi.org/10.1371/journal.pone.0019729).
- [67] T. A. Hopf et al. “Sequence Co-Evolution Gives 3D Contacts and Structures of Protein Complexes”. In: *eLife* 3 (Sept. 2014). Ed. by J. Kuriyan, e03430. ISSN: 2050-084X. DOI: [10.7554/eLife.03430](https://doi.org/10.7554/eLife.03430).
- [68] Q. Cong et al. “Protein Interaction Networks Revealed by Proteome Coevolution”. In: *Science* 365.6449 (July 2019), pp. 185–189. DOI: [10.1126/science.aaw6718](https://doi.org/10.1126/science.aaw6718).
- [69] E. De Leonardis et al. “Direct-Coupling Analysis of Nucleotide Coevolution Facilitates RNA Secondary and Tertiary Structure Prediction”. In: *Nucleic Acids Research* 43.21 (Dec. 2015), pp. 10444–10455. ISSN: 1362-4962. DOI: [10.1093/nar/gkv932](https://doi.org/10.1093/nar/gkv932).
- [70] S. W. Lockless and R. Ranganathan. “Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families”. In: *Science* 286.5438 (Oct. 1999), pp. 295–299. DOI: [10.1126/science.286.5438.295](https://doi.org/10.1126/science.286.5438.295).
- [71] M. Figliuzzi et al. “Coevolutionary Landscape Inference and the Context-Dependence of Mutations in Beta-Lactamase TEM-1”. In: *Molecular Biology and Evolution* 33.1 (Jan. 2016), pp. 268–280. ISSN: 0737-4038. DOI: [10.1093/molbev/msv211](https://doi.org/10.1093/molbev/msv211).
- [72] M. De Leonardis, A. Pagnani, and P. Barrat-Charlaix. “Reconstruction of Ancestral Protein Sequences Using Autoregressive Generative Models”. In: *Molecular Biology and Evolution* 42.4 (Apr. 2025), msaf070. ISSN: 1537-1719. DOI: [10.1093/molbev/msaf070](https://doi.org/10.1093/molbev/msaf070).
- [73] R. Rende. “Mapping of Attention Mechanisms to a Generalized Potts Model”. In: *Physical Review Research* 6.2 (2024). DOI: [10.1103/PhysRevResearch.6.023057](https://doi.org/10.1103/PhysRevResearch.6.023057).
- [74] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. 2025.
- [75] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, May 1999. ISBN: 978-0-262-13360-9.
- [76] R. Kneser and H. Ney. “Improved Backing-off for M-gram Language Modeling”. In: *1995 International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. May 1995, 181–184 vol.1. DOI: [10.1109/ICASSP.1995.479394](https://doi.org/10.1109/ICASSP.1995.479394).
- [77] S. F. Chen and J. Goodman. “An Empirical Study of Smoothing Techniques for Language Modeling”. In: *Computer Speech & Language* 13.4 (Oct. 1999), pp. 359–394. ISSN: 0885-2308. DOI: [10.1006/csla.1999.0128](https://doi.org/10.1006/csla.1999.0128).

- [78] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing”. In: *Commun. ACM* 18.11 (Nov. 1975), pp. 613–620. ISSN: 0001-0782. DOI: [10.1145/361219.361220](https://doi.org/10.1145/361219.361220).
- [79] S. Deerwester et al. “Indexing by Latent Semantic Analysis”. In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407. ISSN: 1097-4571. DOI: [10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-ASII>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASII>3.0.CO;2-9).
- [80] T. Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. Sept. 2013. DOI: [10.48550/arXiv.1301.3781](https://doi.org/10.48550/arXiv.1301.3781).
- [81] J. L. Elman. “Finding Structure in Time”. In: *Cognitive Science* 14.2 (Apr. 1990), pp. 179–211. ISSN: 0364-0213. DOI: [10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E).
- [82] K. Cho et al. “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by A. Moschitti, B. Pang, and W. Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: [10.3115/v1/D14-1179](https://doi.org/10.3115/v1/D14-1179).
- [83] F. Caredda. “Attention Based Direct Coupling Analysis for Protein Structure Prediction”. MA thesis. Politecnico di Torino, July 2022.
- [84] I. Sutskever, O. Vinyals, and Q. V. Le. *Sequence to Sequence Learning with Neural Networks*. Dec. 2014. DOI: [10.48550/arXiv.1409.3215](https://doi.org/10.48550/arXiv.1409.3215).
- [85] M.-T. Luong, H. Pham, and C. D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*. Sept. 2015. DOI: [10.48550/arXiv.1508.04025](https://doi.org/10.48550/arXiv.1508.04025).
- [86] D. Ofer, N. Brandes, and M. Linial. “The Language of Proteins: NLP, Machine Learning & Protein Sequences”. In: *Computational and Structural Biotechnology Journal* 19 (Jan. 2021), pp. 1750–1758. ISSN: 2001-0370. DOI: [10.1016/j.csbj.2021.03.022](https://doi.org/10.1016/j.csbj.2021.03.022).
- [87] A. Rives et al. “Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (Apr. 2021), e2016239118. DOI: [10.1073/pnas.2016239118](https://doi.org/10.1073/pnas.2016239118).
- [88] E. Asgari and M. R. K. Mofrad. “Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics”. In: *PLOS ONE* 10.11 (Nov. 2015), e0141287. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0141287](https://doi.org/10.1371/journal.pone.0141287).
- [89] T. Bepler and B. Berger. *Learning Protein Sequence Embeddings Using Information from Structure*. Oct. 2019. DOI: [10.48550/arXiv.1902.08661](https://doi.org/10.48550/arXiv.1902.08661).
- [90] A. Elnaggar et al. *ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Deep Learning and High Performance Computing*. May 2021. DOI: [10.48550/arXiv.2007.06225](https://doi.org/10.48550/arXiv.2007.06225).
- [91] R. Rao et al. *Transformer Protein Language Models Are Unsupervised Structure Learners*. Dec. 2020. DOI: [10.1101/2020.12.15.422761](https://doi.org/10.1101/2020.12.15.422761).

- [92] J. Vig et al. *BERTology Meets Biology: Interpreting Attention in Protein Language Models*. Mar. 2021. DOI: [10.48550/arXiv.2006.15222](https://doi.org/10.48550/arXiv.2006.15222).
- [93] R. Evans et al. *Protein Complex Prediction with AlphaFold-Multimer*. Mar. 2022. DOI: [10.1101/2021.10.04.463034](https://doi.org/10.1101/2021.10.04.463034).
- [94] J. Abramson et al. “Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3”. In: *Nature* 630.8016 (June 2024), pp. 493–500. ISSN: 1476-4687. DOI: [10.1038/s41586-024-07487-w](https://doi.org/10.1038/s41586-024-07487-w).
- [95] N. Bhattacharya et al. “Interpreting Potts and Transformer Protein Models Through the Lens of Simplified Attention”. In: *Biocomputing 2022*. WORLD SCIENTIFIC, Sept. 2021, pp. 34–45. ISBN: 978-981-12-5046-0. DOI: [10.1142/9789811250477_0004](https://doi.org/10.1142/9789811250477_0004).
- [96] L. Gennai et al. *Partner Determination from Protein Sequences Using Class Information with CLAPP*. May 2026. DOI: [10.64898/2026.05.06.723283](https://doi.org/10.64898/2026.05.06.723283).
- [97] S. Cocco, R. Monasson, and V. Sessak. “High-Dimensional Inference with the Generalized Hopfield Model: Principal Component Analysis and Corrections”. In: *Physical Review E* 83.5 (May 2011), p. 051123. ISSN: 1539-3755, 1550-2376. DOI: [10.1103/PhysRevE.83.051123](https://doi.org/10.1103/PhysRevE.83.051123).
- [98] S. Cocco, R. Monasson, and M. Weigt. “From Principal Component to Direct Coupling Analysis of Coevolution in Proteins: Low-Eigenvalue Modes Are Needed for Structure Prediction”. In: *PLoS Computational Biology* 9.8 (Aug. 2013), e1003176. ISSN: 1553-734X. DOI: [10.1371/journal.pcbi.1003176](https://doi.org/10.1371/journal.pcbi.1003176).
- [99] P. M. Wolanin, P. A. Thomason, and J. B. Stock. “Histidine Protein Kinases: Key Signal Transducers Outside the Animal Kingdom”. In: *Genome Biology* 3.10 (Sept. 2002), reviews3013.1. ISSN: 1474-760X. DOI: [10.1186/gb-2002-3-10-reviews3013](https://doi.org/10.1186/gb-2002-3-10-reviews3013).
- [100] J. Zhang. “Evolution by Gene Duplication: An Update”. In: *Trends in Ecology & Evolution* 18.6 (June 2003), pp. 292–298. ISSN: 0169-5347. DOI: [10.1016/S0169-5347\(03\)00033-8](https://doi.org/10.1016/S0169-5347(03)00033-8).
- [101] G. C. Conant and K. H. Wolfe. “Turning a Hobby into a Job: How Duplicated Genes Find New Functions”. In: *Nature Reviews Genetics* 9.12 (Dec. 2008), pp. 938–950. ISSN: 1471-0064. DOI: [10.1038/nrg2482](https://doi.org/10.1038/nrg2482).
- [102] R. Szklarczyk, M. A. Huynen, and B. Snel. “Complex Fate of Paralogs”. In: *BMC Evolutionary Biology* 8.1 (Dec. 2008), p. 337. ISSN: 1471-2148. DOI: [10.1186/1471-2148-8-337](https://doi.org/10.1186/1471-2148-8-337).
- [103] M. Y. Galperin and E. V. Koonin. “Who’s Your Neighbor? New Computational Approaches for Functional Genomics”. In: *Nature Biotechnology* 18.6 (June 2000), pp. 609–613. ISSN: 1546-1696. DOI: [10.1038/76443](https://doi.org/10.1038/76443).
- [104] T. Gueudré et al. “Simultaneous Identification of Specifically Interacting Paralogs and Interprotein Contacts by Direct Coupling Analysis”. In: *Proceedings of the National Academy of Sciences* 113.43 (Oct. 2016), pp. 12186–12191. DOI: [10.1073/pnas.1607570113](https://doi.org/10.1073/pnas.1607570113).

- [105] A.-F. Bitbol et al. “Inferring Interaction Partners from Protein Sequences”. In: *Proceedings of the National Academy of Sciences* 113.43 (Oct. 2016), pp. 12180–12185. DOI: [10.1073/pnas.1606762113](https://doi.org/10.1073/pnas.1606762113).
- [106] S. Ovchinnikov, H. Kamisetty, and D. Baker. “Robust and Accurate Prediction of Residue–Residue Interactions across Protein Interfaces Using Evolutionary Information”. In: *eLife* 3 (May 2014). Ed. by B. Roux, e02030. ISSN: 2050-084X. DOI: [10.7554/eLife.02030](https://doi.org/10.7554/eLife.02030).
- [107] D. Repecka et al. “Expanding Functional Protein Sequence Spaces Using Generative Adversarial Networks”. In: *Nature Machine Intelligence* 3.4 (Apr. 2021), pp. 324–333. ISSN: 2522-5839. DOI: [10.1038/s42256-021-00310-5](https://doi.org/10.1038/s42256-021-00310-5).
- [108] A. Madani et al. “Large Language Models Generate Functional Protein Sequences across Diverse Families”. In: *Nature Biotechnology* 41.8 (Aug. 2023), pp. 1099–1106. ISSN: 1546-1696. DOI: [10.1038/s41587-022-01618-2](https://doi.org/10.1038/s41587-022-01618-2).
- [109] B. L. Trippe et al. *Diffusion Probabilistic Modeling of Protein Backbones in 3D for the Motif-Scaffolding Problem*. Mar. 2023. DOI: [10.48550/arXiv.2206.04119](https://doi.org/10.48550/arXiv.2206.04119).
- [110] J. L. Watson et al. “De Novo Design of Protein Structure and Function with RFDiffusion”. In: *Nature* 620.7976 (Aug. 2023), pp. 1089–1100. ISSN: 1476-4687. DOI: [10.1038/s41586-023-06415-8](https://doi.org/10.1038/s41586-023-06415-8).

Appendix A

Statistical Inference Models

A.1 Derivation of the Mean-Field Approximation

In Section 2.4 we introduced the mean-field Direct Coupling Analysis (mfDCA) as an approximate strategy to infer Potts model couplings by inverting the covariance matrix of the data. Here we present a detailed derivation of this approximation by means of a high-temperature (weak-coupling) expansion known as the Plefka expansion. This derivation is valid for the Potts model, regardless of the context of its application.

Let us consider the standard Potts distribution

$$P(\mathbf{a}) = \frac{1}{Z} \exp \left(\sum_{i=1}^L h_i(a_i) + \sum_{i < j} J_{ij}(a_i, a_j) \right), \quad (\text{A.1})$$

with q possible states at each site. From statistical mechanics, we can relate the single-site marginals and the connected correlations to the partition function as:

$$\frac{\partial \log Z}{\partial h_j(a)} = \frac{1}{Z} \sum_{\mathbf{s}} \delta(s_j, a) \exp \left\{ \sum_{i=1}^L h_i(s_i) + \sum_{i < j} J_{ij}(s_i, s_j) \right\} = P_j(a) \quad , \quad (\text{A.2})$$

$$\frac{\partial^2 \log Z}{\partial h_j(a) \partial h_k(b)} = P_{jk}(a, b) - P_j(a) P_k(b) = C_{jk}(a, b) \quad . \quad (\text{A.3})$$

To implement a small coupling expansion we need to introduce a perturbative parameter $\alpha \ll 1$ that interpolates between a site-independent profile model (for

$\alpha = 0$) and a full Potts model (for $\alpha = 1$). At this point, we can introduce the Helmholtz free energy $F = -\log Z$ and its Legendre transform, the Gibbs potential. Both functions depend on the perturbative parameter α , but while the Helmholtz free energy depends on the local fields $\mathbf{h}$, the Gibbs potential depends on their conjugate variables, i.e. the single-site marginal probabilities as shown in Equation A.2. The relationship between the two thermodynamical potentials is given by:

$$F(\alpha, \mathbf{h}) = \min_{\mathbf{P}} \left[G(\alpha, \mathbf{P}) - \sum_{i=1}^L \sum_{a=1}^{q-1} h_i(a) P_i(a) \right] , \quad (\text{A.4})$$

$$G(\alpha, \mathbf{P}) = \sup_{\mathbf{h}} \left[F(\alpha, \mathbf{h}) + \sum_{i=1}^L \sum_{a=1}^{q-1} h_i(a) P_i(a) \right] . \quad (\text{A.5})$$

The extremum condition in Equation A.4 can be written as $\partial_{h_j(a)} [F(\alpha, \mathbf{h}) + \mathbf{h} \cdot \mathbf{P}]$ if F is differentiable. In this case, a fluctuation-dissipation theorem can be written as

$$\frac{\partial G(\alpha)}{\partial P_j(b)} = h_j(b) , \quad (\text{A.6})$$

$$\frac{\partial h_j(b)}{\partial P_i(a)} = (C^{-1})_{ij}(a, b) = \frac{\partial^2 G(\alpha, \mathbf{P})}{\partial P_i(a) \partial P_j(b)} . \quad (\text{A.7})$$

At this point, we need to perform the small coupling expansion of the Gibbs potential with respect to α . In particular, it is more convenient to perform the expansion of the potential written as $\mathcal{G}(\alpha) = -\log Z + \mathbf{h} \cdot \mathbf{Q}$, and applying the extremization condition afterwards, so as to finally get

$$G(\alpha) = G(0) + \alpha \left. \frac{dG(\alpha)}{d\alpha} \right|_{\alpha=0} + \mathcal{O}(\alpha^2) . \quad (\text{A.8})$$

At zeroth order, the pre-extremization potential reads $\mathcal{G}(0) = -\log \sum_{\mathbf{s}} \exp\{\sum_i h_i(s_i)\} + \sum_i \sum_a h_i(a) P_i(a)$. Using Equations A.2 and A.3 we can write the zeroth order term as

$$G(0) = \sum_{i=1}^L \sum_{a=1}^q P_i(a) \log P_i(a) \quad (\text{A.9})$$

In order to compute the first derivative, it is important to observe that the fields can be explicitly written as a function of α , by inverting Equation A.2 in its form after

the introduction of the perturbative parameter. In this case we have

$$\begin{aligned} \frac{\partial \mathcal{G}}{\partial \alpha} &= -\frac{d \log Z(\alpha)}{d\alpha} + \sum_{i=1}^L \sum_{a=1}^q \frac{dh_i(a)}{d\alpha} P_i(\alpha) = \\ &= \frac{1}{Z(\alpha)} \sum_{\mathbf{s}} \left[\sum_{i<j} J_{ij}(s_i, s_j) + \sum_{i=1}^L \frac{dh_i(s_i)}{d\alpha} \right] e^{-H(\alpha)} + \sum_{i=1}^L \sum_{a=1}^q \frac{dh_i(a)}{d\alpha} P_i(\alpha) \quad . \quad (\text{A.10}) \end{aligned}$$

When evaluating this in $\alpha = 0$ we get

$$\left. \frac{dG(\alpha)}{d\alpha} \right|_{\alpha=0} = -\sum_{i<j} \langle J_{ij} \rangle_0 = -\sum_{i<j} \sum_{a,b=1}^q J_{ij}(a,b) P_i(a) P_j(b) \quad . \quad (\text{A.11})$$

The final expansion can be written as

$$G(\alpha, \mathbf{P}) = \sum_{i=1}^L \sum_{a=1}^q P_i(a) \log P_i(a) - \alpha \sum_{i<j} \sum_{a,b=1}^q J_{ij}(a,b) P_i(a) P_j(b) + \mathcal{O}(\alpha^2). \quad (\text{A.12})$$

The original model can be recovered by setting the perturbative parameter $\alpha = 1$. As mentioned in Section 2.3, the number of parameters in the Potts model is redundant and a choice of gauge is required. This is even more evident by inspecting the degeneracy of the connected correlation function that makes it not invertible:

$$\sum_{b=1}^q C_{ij}(a,b) = \sum_{b=1}^q P_{ij}(a,b) - P_i(a) \sum_{b=1}^q P_j(b) = P_i(a) - P_i(a) = 0 \quad (\text{A.13})$$

Fixing a lattice-gas gauge, by setting the single-site marginal of the q color as $P_i(q) = 1 - \sum_{a=1}^{q-1} P_i(a)$, we can derive a set of self-consistent equations for the Gibbs potential:

$$\frac{\partial G(\mathbf{P})}{\partial P_k(c)} = h_k(c) = \log \frac{P_k(c)}{P_k(q)} - \sum_{j \neq k} \sum_{a=1}^{q-1} J_{kj}(c,a) P_j(a) \quad , \quad (\text{A.14})$$

$$\frac{\partial^2 G(\mathbf{P})}{\partial P_k(c) \partial P_l(d)} = (C^{-1})_{kl}(c,d) = \begin{cases} -J_{kl}(c,d) & k \neq l \\ \frac{\delta(c,d)}{P_k(c)} + \frac{1}{P_k(q)} & k = l \end{cases} \quad . \quad (\text{A.15})$$

From these equations we can conclude that, provided we can invert the empirical correlation matrix $C_{\text{emp}} \in \mathbb{R}^{\mathbb{L}, \mathbb{L}, \|\cdot\| - \mathbb{K}, \|\cdot\| - \mathbb{K}}$, we can extract the fields and couplings within

the mean field approximation of the Potts Hamiltonian. However, the approximation has clear limitations, as it neglects higher-order correlations, which can lead to poor accuracy in regimes of strong couplings. Moreover, covariance matrix inversion is notoriously sensitive to sampling noise, making mfDCA unstable for shallow alignments. For these reasons, mfDCA is now of mainly historical and pedagogical interest.

A.2 Derivation of the Gaussian Approximation

In Section 2.4.3 we introduced the Gaussian approximation in the context of Direct Coupling Analysis (GaussDCA) as a tractable alternative to the discrete inverse Potts problem. The key idea is to relax the discrete nature of sequences into a one-hot representation and model them as independent samples from a multivariate Gaussian distribution. In particular, let $D = \{\mathbf{a}^1, \dots, \mathbf{a}^M\}$ be an alignment of M sequences of length L over an alphabet of size q (amino acids plus gap). We can define one-hot (indicator) variables

$$x_i^m(a) = \mathbf{1}\{a_i^m = a\} \in \{0, 1\}, \quad i = 1, \dots, L, \quad a \in \{1, \dots, q\}.$$

To remove the linear dependence $\sum_a x_i^m(a) = 1$ at each site, we fix a gauge by selecting a reference state q (often the gap) and drop it, so that the one-hot encoding of the reference state is the null vector. Consequently, each sequence m is represented by a vector

$$\mathbf{x}^m \in \mathbb{R}^p, \quad p = L(q - 1).$$

The multivariate Gaussian distribution used to model the data is uniquely determined by its mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$, which are the parameters to be inferred within the Gaussian approximation scheme. This can be done by maximizing the likelihood function associated to the multivariate distribution and the data D :

$$P(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{Mp/2} (\det \boldsymbol{\Sigma})^{M/2}} \exp\left(-\frac{1}{2} \sum_{m=1}^M (\mathbf{x}^m - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}^m - \boldsymbol{\mu})\right) \quad (\text{A.16})$$

A Maximum Likelihood Estimate of the parameters would result in the empirical observables defined as

$$\boldsymbol{\mu}_{\text{MLE}} = \bar{\mathbf{x}} = \sum_{m=1}^M \mathbf{x}^m, \quad (\text{A.17})$$

$$\Sigma_{\text{MLE}} = \bar{C} = \sum_{m=1}^M \mathbf{x}^{mT} \mathbf{x}^m - \bar{\mathbf{x}}^T \bar{\mathbf{x}}. \quad (\text{A.18})$$

However, due to the undersampling of sequence space in the observations D , matrix $\bar{C}$ is rank deficient and thus not invertible. A solution to this practical problem is to define a prior distribution of the parameters and perform a maximum a posteriori estimate, in which the object of the optimization is the posterior distribution defined as

$$P(\boldsymbol{\mu}, \Sigma | D) \propto P(D | \boldsymbol{\mu}, \Sigma) P(\boldsymbol{\mu}, \Sigma). \quad (\text{A.19})$$

A convenient choice of the prior introduced in [54] is the Normal Inverse Wishart (NIW), for which

$$P(\boldsymbol{\mu} | \Sigma) = (2\pi)^{-\frac{p}{2}} \kappa^{\frac{p}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp \left\{ -\frac{\kappa}{2} (\boldsymbol{\mu} - \boldsymbol{\eta})^T (\Sigma)^{-1} (\boldsymbol{\mu} - \boldsymbol{\eta}) \right\}, \quad (\text{A.20})$$

$$P(\Sigma) \propto \det(\Sigma)^{-\frac{v+p+1}{2}} \exp \left\{ -\frac{1}{2} \text{Tr}(\Lambda \Sigma^{-1}) \right\}. \quad (\text{A.21})$$

Here, the prior over the mean vector $\boldsymbol{\mu}$ is still a multivariate Gaussian with mean $\boldsymbol{\eta}$ and covariance matrix Σ/κ . The average values of $\boldsymbol{\mu}$ and Σ over the prior distributions are

$$\langle \boldsymbol{\mu} \rangle_{P(\boldsymbol{\mu}, \Sigma)} = \boldsymbol{\eta}, \quad (\text{A.22})$$

$$\langle \Sigma \rangle_{P(\boldsymbol{\mu}, \Sigma)} = \frac{\Lambda}{v - p - 1}. \quad (\text{A.23})$$

Since the NIW is the conjugate prior distribution of the multivariate Gaussian, it means that the posterior is a NIW as well, with parameters

$$\begin{cases} \kappa' = \kappa + M \\ \boldsymbol{\eta}' = \frac{\kappa}{\kappa+M} \boldsymbol{\eta} + \frac{M}{\kappa+M} \bar{\mathbf{x}} \\ v' = v + M \\ \Lambda' = \Lambda + M\bar{C} + \frac{\kappa M}{\kappa+M} (\bar{\mathbf{x}} - \boldsymbol{\eta})^T (\bar{\mathbf{x}} - \boldsymbol{\eta}) \end{cases} \quad (\text{A.24})$$

Then, the a posteriori estimates of mean and variance become

$$\langle \boldsymbol{\mu} \rangle_{P(\boldsymbol{\mu}, \Sigma) | D} = \boldsymbol{\eta}' = \frac{\kappa}{\kappa + M} \boldsymbol{\eta} + \frac{M}{\kappa + M} \bar{\mathbf{x}} \quad , \quad (\text{A.25})$$

$$\langle \Sigma \rangle_{P(\boldsymbol{\mu}, \Sigma) | D} = \frac{\Lambda'}{\nu' - p - 1} = \frac{\Lambda + M\bar{C} + \frac{\kappa M}{\kappa + M} (\bar{\mathbf{x}} - \boldsymbol{\eta})^T (\bar{\mathbf{x}} - \boldsymbol{\eta})}{\nu + M - p - 1} \quad . \quad (\text{A.26})$$

In order to have a prior as uninformative as possible, we set its mean and covariance matrix as those of a uniform distribution, so that $\eta_i = 1/q \forall i \in 1, \dots, p$ and $U = \Lambda/(\nu - p - 1)$ is a block-matrix composed of $L \times L$ blocks of size $q \times q$ such that out-of-diagonal blocks are zero, and diagonal blocks are distributed as

$$U_{(k+1)(q-1)+a, (l-1)(q-1)+b} = \frac{\delta(k, l)}{q} \left(\delta(a, b) - \frac{1}{q} \right) \quad , \quad (\text{A.27})$$

where $k, l \in \{1, \dots, L\}$ and $a, b \in \{1, \dots, q\}$. Finally, if we choose $\nu = \kappa + p + 1$ we interpret $\lambda = \kappa / (\kappa + M)$ as the pseudo-count introduced in Section 2.5. The covariance matrix can be written as

$$\Sigma = \langle \Sigma \rangle_{P(\boldsymbol{\mu}, \Sigma) | D} = \lambda U + (1 - \lambda) \bar{C} + \lambda (1 - \lambda) (\bar{\mathbf{x}} - \boldsymbol{\eta})^T (\bar{\mathbf{x}} - \boldsymbol{\eta}) \quad (\text{A.28})$$

and $J = \Sigma^{-1}$, similarly to the mean-field case.

A.3 Gauge-Fixing Regularization in PlmDCA

In Section 2.3.3 we shown that the generalized Potts model is invariant under a set of gauge transformations of its parameters, field h and couplings J . These transformations leave the model distribution invariant but modify the individual parameters, revealing that the Inverse Potts problem is intrinsically over-parameterized. While the choice of gauge has no effect on the underlying probability distribution, it becomes essential whenever quantities depending explicitly on the couplings are evaluated, such as in the case of contact prediction. In practice, for this task, the zero-sum gauge represents a common choice as it minimizes the Frobenious norm, effectively shifting site-specific biases into the fields and removing spurious contributions.

When the gauge is not imposed a posteriori, it is fixed automatically by the regularization scheme employed during inference. This can be shown explicitly

for the asymmetric pseudo-likelihood implementation of the Potts model. Starting from the regularised pseudo-likelihood defined in Equations 2.38, 2.39, and 2.47, the gradients of the site-wise objective $\mathcal{L}_i$ with respect to the fields and couplings are

$$\frac{\partial \mathcal{L}_i}{\partial h_i(c)} = \frac{1}{M} \sum_{m=1}^M \left[\delta(c, a_i^m) - P(c \mid \mathbf{a}_{\setminus i}^m) \right] - 2\lambda_h h_i(c), \quad (\text{A.29})$$

$$\frac{\partial \mathcal{L}_i}{\partial J_{ij}(c, d)} = \frac{1}{M} \sum_{m=1}^M \delta(d, a_j^m) \left[\delta(c, a_i^m) - P(c \mid \mathbf{a}_{\setminus i}^m) \right] - 2\lambda_J J_{ij}(c, d). \quad (\text{A.30})$$

At convergence, when the gradients vanish, the fields can be expressed explicitly from Equation A.29 as

$$h_i(c) = \frac{1}{2\lambda_h M} \sum_{m=1}^M \left[\delta(c, a_i^m) - P(c \mid \mathbf{a}_{\setminus i}^m) \right]. \quad (\text{A.31})$$

Summing this expression over all amino acids $c = 1, \dots, q$ yields

$$\sum_{c=1}^q h_i(c) = 0, \quad (\text{A.32})$$

which is precisely the zero-sum gauge condition for the fields. A similar argument applies to the couplings: summing Eq. A.30 over c and or d gives the gauge conditions

$$\begin{cases} \sum_{c=1}^q J_{ij}(c, d) = 0, \\ \sum_{d=1}^q J_{ij}(c, d) = \frac{\lambda_h}{\lambda_J} h_i(c), \end{cases} \quad (\text{A.33})$$

showing that the combined regularization of fields and couplings enforces a unique parameterisation of the model.

An analogous conclusion holds for the case of symmetrised couplings: $L2$ regularization removes the gauge redundancy and fixes the parameters in a unique representation. Thus, even when an explicit gauge transformation is not applied after inference, the regularization employed during pseudo-likelihood maximization acts as an implicit gauge-fixing condition.

A.4 Pseudo-likelihood versus Autoregressive Factorizations in Potts Models

In this section we provide a detailed mathematical comparison between the pseudo-likelihood and autoregressive approximations introduced in Section 2.4. Although the two approaches lead to optimization problems that share structural similarities, they differ fundamentally in the level at which the approximation is introduced: pseudo-likelihood approximates the joint distribution, while the autoregressive approach approximates the conditional distributions. Throughout, we consider the q -state symmetric Potts model over sequences $\mathbf{a} = (a_1, \dots, a_L)$:

$$P(\mathbf{a}) = \frac{1}{Z} \exp \left(\sum_{i=1}^L h_i(a_i) + \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j) \right) . \quad (\text{A.34})$$

A.4.1 Pseudo-likelihood factorization

Consider first the conditional distribution of residue a_i given all other positions. From A.34, one finds

$$P(a_i | \mathbf{a}_{\setminus i}) = \frac{P(\mathbf{a})}{\sum_{b=1}^q P(a_i=b, \mathbf{a}_{\setminus i})} = \frac{\exp \left(h_i(a_i) + \sum_{j \neq i} J_{ij}(a_i, a_j) \right)}{\sum_{b=1}^q \exp \left(h_i(b) + \sum_{j \neq i} J_{ij}(b, a_j) \right)} . \quad (\text{A.35})$$

Thus, the exact conditional $P(a_i | \mathbf{a}_{\setminus i})$ is a multinomial logistic (softmax) distribution with linear logits in the fields and couplings. Each conditional is therefore itself a Potts distribution over the q possible states of site i . The pseudo-likelihood approximation replaces the intractable joint likelihood by the product of these conditionals:

$$\begin{aligned} P_{\text{PL}}(\mathbf{a}) &= \prod_{i=1}^L P(a_i | \mathbf{a}_{\setminus i}) \\ &= \prod_{i=1}^L \frac{\exp \left(h_i(a_i) + \sum_{j \neq i} J_{ij}(a_i, a_j) \right)}{\sum_{b=1}^q \exp \left(h_i(b) + \sum_{j \neq i} J_{ij}(b, a_j) \right)} \\ &= \frac{1}{Z_i(\mathbf{a}_{\setminus i})} \exp \left(h_i(a_i) + \sum_{j \neq i} J_{ij}(a_i, a_j) \right) . \end{aligned} \quad (\text{A.36})$$

However, a comparison between A.36 and A.34 yields

$$\frac{P_{\text{PL}}(\mathbf{a})}{P(\mathbf{a})} = \frac{\prod_{i=1}^L Z_i(\mathbf{a}_{\setminus i})}{Z} . \quad (\text{A.37})$$

Since the local normalization factors $Z_i(\mathbf{a}_{\setminus i})$ depend on the full configuration $\mathbf{a}$, this ratio is not constant over sequence space. Hence P_{PL} is not proportional to, and therefore not equal to, the true joint probability distribution. This provides the precise sense in which the pseudo-likelihood is a non-exact factorization of the Potts model. Despite this, maximizing the average log-pseudo-likelihood

$$\ell_{\text{PL}} = \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^L \log P(a_i^m \mid \mathbf{a}_{\setminus i}^m) \quad (\text{A.38})$$

amounts to solving a collection of convex logistic-regression subproblems, and under standard regularity assumptions yields statistically consistent estimates as $M \rightarrow \infty$.

A.4.2 Autoregressive factorization

We now turn to the autoregressive construction. For any ordering π of $\{1, \dots, L\}$, the chain rule provides an exact factorization of the joint distribution:

$$P(\mathbf{a}) = \prod_{k=1}^L P(a_{\pi(k)} \mid a_{\pi(1)}, \dots, a_{\pi(k-1)}) . \quad (\text{A.39})$$

As a consequence, the log-likelihood of the model can be factorized exactly as

$$\ell_{\text{AR}} = \frac{1}{M} \sum_{m=1}^M \sum_{k=1}^L \log P(a_{\pi(k)}^m \mid a_{\pi(1)}^m, \dots, a_{\pi(k-1)}^m) . \quad (\text{A.40})$$

Unlike the pseudo-likelihood, this is an exact decomposition of the true likelihood. However, the corresponding conditionals cannot in general be written in Potts form. To see this, let us fix the natural order $\pi(k) = k$ and compute the conditional

distribution associated with site i , namely $P(a_i | \mathbf{a}_{<i})$. By definition,

$$\begin{aligned} P(a_i | \mathbf{a}_{<i}) &= \frac{\sum_{\mathbf{a}_{>i}} P(a_1, \dots, a_{i-1}, \mathbf{a}_{>i})}{\sum_{a_i=1}^q \sum_{\mathbf{a}_{>i}} P(a_1, \dots, a_{i-1}, a_i, \mathbf{a}_{>i})} \\ &= \frac{\sum_{\mathbf{a}_{>i}} \exp\left(\sum_{j=1}^L h_j(a_j) + \sum_{j<k} J_{jk}(a_j, a_k)\right)}{\sum_{a_i=1}^q \sum_{\mathbf{a}_{>i}} \exp\left(\sum_{j=1}^L h_j(a_j) + \sum_{j<k} J_{jk}(a_j, a_k)\right)} . \end{aligned} \quad (\text{A.41})$$

Although exact, this expression does not simplify to a Potts-like conditional of the form

$$P(a_i | \mathbf{a}_{<i}) \propto \exp\left(h_i(a_i) + \sum_{j<i} J_{ij}(a_i, a_j)\right) ,$$

because marginalizing over $\mathbf{a}_{>i}$ introduces additional, sequence-dependent contributions that cannot be absorbed into simple single-site fields or pairwise couplings. Equation A.41 therefore demonstrates that the exact autoregressive conditionals of a Potts model are, in general, not Potts conditionals.

Appendix B

AttentionDCA: Supplementary Information

The content of this appendix is based on the Supporting Information accompanying the paper “*Direct Coupling Analysis and the Attention Mechanism*” [42]. All sections have been adapted and reworded to fit the structure of the present thesis.

B.1 Contact scores

B.1.1 Attention and Frobenius Scores

The key distinction between the Frobenius Score (FS) and the Attention Score (AS) is that the former integrates the amino-acid information inside the Value matrices, while the latter depends only on the family-dependent attention heads. However, as shown in the main text, the performances of these two metrics are almost identical across different families. This observation suggests that the Value matrices play a little role in the construction of the FS, and therefore in contact prediction. Analytically,

considering the definition of the FS for the factored attention model, we can write

$$\begin{aligned}
 F_{ij} &= \sqrt{\sum_{a,b} J_{ij}^2(a,b)} \\
 &= \sqrt{\sum_{a,b} \left(\sum_{h=1}^H A_{ij}^h V_{ab}^h \right)^2} = \sqrt{\sum_{a,b=1}^q \sum_{h,k=1}^H A_{ij}^h A_{ij}^k V_{ab}^h V_{ab}^k} \\
 &= \sqrt{\sum_{h,k=1}^H A_{ij}^h A_{ij}^k \sum_{a,b=1}^q V_{ab}^h V_{ab}^k} \quad .
 \end{aligned} \tag{B.1}$$

We can introduce the overlap matrix $\Lambda \in \mathbb{R}^{H,H}$ as the scalar product between Value matrices from different heads $\{V^h\}_{h=1,\dots,H}$:

$$\Lambda^{h,k} = \sum_{a,b} V_{a,b}^h V_{a,b}^k \quad . \tag{B.2}$$

Figure B.1 shows that the overlap Λ computed on a set of inferred Value matrices is approximately diagonal, indicating near orthogonality of the heads:

$$\Lambda^{h,k} \propto \begin{cases} \gamma(h) & h = k \\ 0 & h \neq k \end{cases} \tag{B.3}$$

In this context, the FS can be written as

$$F_{ij} = \sqrt{\sum_h^H \gamma(h) A_{ij}^{h^2}} \quad . \tag{B.4}$$

In this form, FS appears as a weighted average of the squared attention heads, where the relative weight of each head is encoded inside the Value matrices. However, as shown in Figure B.1, the relative weight of each head is almost equally distributed across the diagonal. Hence, the contribution of the Value matrices to the FS is approximately a constant factor. This explains why FS and AS perform similarly.

Although the Value matrices $\{V^h\}_{h=1,\dots,H}$ appear to have little influence on contact prediction itself, they remain crucial as trainable parameters of the model. To test this, we constructed a variant where only the $\{Q^h, K^h\}_{h=1,\dots,H}$ are learned, while the Value matrices are kept fixed. When V matrices are frozen to random initial

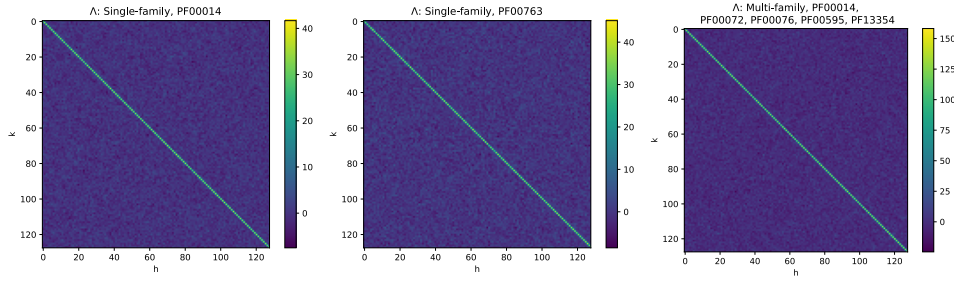


Fig. B.1 Overlap matrix of the set of Value matrices computed on PF00014 and PF00763 as single-family and on PF00014, PF00072, PF00076, PF00595, PF13354 as multi-family. Source [42]

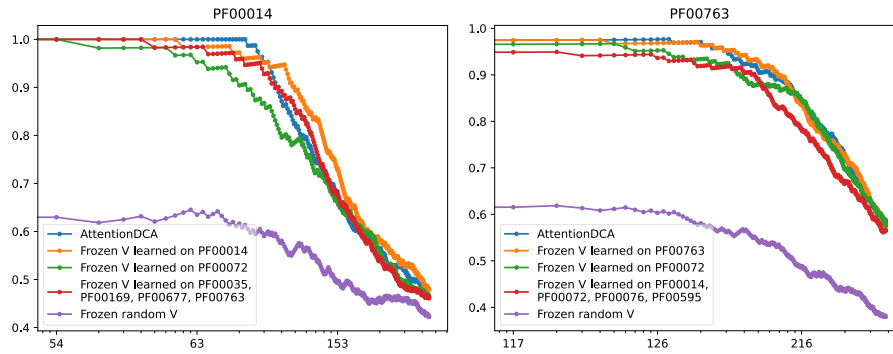


Fig. B.2 Standard AttentionDCA and AttentionDCA with frozen Value matrices for protein families PF00014 and PF00763. Source [42]

values, the PPV drops sharply compared to the standard case where all parameters are trained. On the other hand, when the Value matrices are pre-trained, either on the same family, on another family, or in a multi-family setting, and then frozen, the resulting accuracy is close to that of the standard FS and AS. This is shown in Figure B.2, left panel, for PF00014: the blue curve represents the standard model with all parameters trained; the orange, red and green curves correspond to frozen Value matrices pre-trained respectively on PF00014, PF00072, and a multi-family dataset including PF00035, PF00169, PF00677 and PF00763. Among these, the family-specific pre-training (PF00014) performs best, while the multi-family case slightly outperforms the PF00072 pre-training. By contrast, randomly initialized frozen Value matrices (violet curve) lead to a breakdown of predictive power. The same qualitative behavior is observed for PF00763 (right panel of Figure B.2).

B.1.2 Epistatic Score

As described in Section. 4.1.3 of the main text, in the generative formulation of the model the Epistatic Score (ES) substitutes the interaction tensor when computing the Frobenius norm for contact prediction. For a given pair of residues (i, j) , and candidate amino acids b_i, b_j , the ES is defined as the difference between the effect of mutating both sites simultaneously and the sum of the effects of mutating them individually, with respect to a chosen wild-type sequence $\mathbf{a} = (a_1, a_2, \dots, a_L)$:

$$\Delta\Delta E_{ij}(b_i, b_j) = \Delta E(a_i \rightarrow b_i, a_j \rightarrow b_j) - \Delta E(a_i \rightarrow b_i) - \Delta E(a_j \rightarrow b_j) \quad . \quad (\text{B.5})$$

In practice, $\Delta\Delta E_{ij}(b_i, b_j)$ takes the place of the coupling tensor $J_{ij}(a_i, a_j)$ in the Frobenius norm, yielding the Epistatic Score ES_{ij} .

An important property of this definition is that the choice of wild-type sequence does not influence the final contact prediction. We verified that even if the wild-type is replaced by a randomly generated sequence, the resulting PPV is essentially identical to that obtained with a homolog extracted from the MSA. This demonstrates that the relevant information is already embedded in the inferred interaction terms, and that the ES is invariant with respect to the specific wild-type used in its computation.

For each protein family, Figure B.3 shows the Epistatic Score PPV@L obtained using different wild-type sequences, as discussed in Equation 4.12 of the main text. In the figure, the first hundred sequences are randomly selected from the MSA, while the remaining hundred are generated uniformly at random. The curves confirm that the accuracy of contact prediction is unaffected by whether the wild-type is taken from the family or produced synthetically.

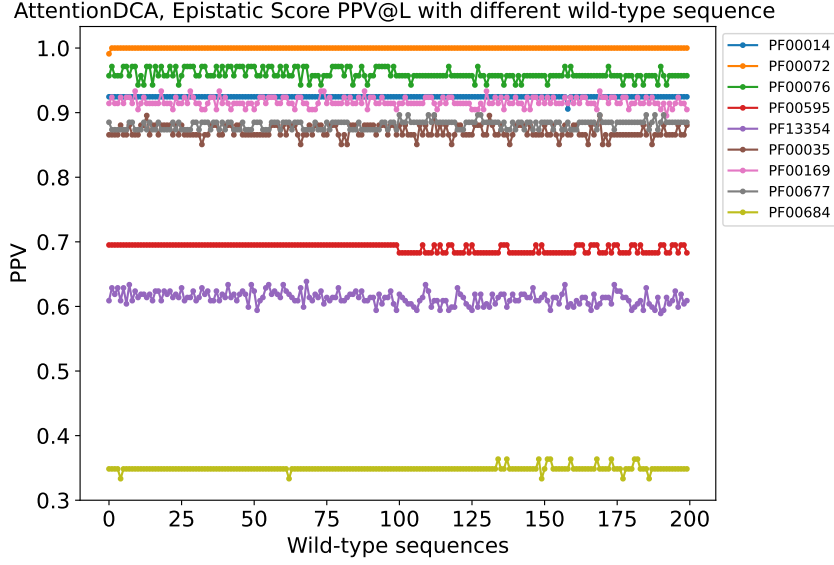


Fig. B.3 Epistatic Score PPV@L for AttentionDCA using different initial wild-type sequences for the computation of the Epistatic Score. The first half of the wild-type sequences are randomly chosen from the MSA, while the second half sequences are generated from a uniform distribution. Each curve represents a protein family according to the legend chart. Source [42]

B.2 Attention matrix sparsity analysis

As discussed in the main text, the structural information captured by each attention head of the model turns out to be highly sparse. A way to quantify this sparsity is as follows: given the set $\{\mathbf{A}^h\}_{h=1,\dots,H}$ (with $A_{ij}^h \geq 0$ for all h and $i, j = 1, \dots, L$), we select from each head the k largest entries. Collecting these values yields a list S of $k \times H$ elements. For each position pair (i, j) we then form a new matrix $\tilde{A}_{i,j}$ either by taking the maximum or by averaging across the subset $S_{(i,j)} \subset S$ corresponding to that pair:

$$\tilde{A}_{i,j} = \begin{cases} \max\{A_{ij}^h \mid A_{ij}^h \in S_{(i,j)}\} \\ \frac{1}{|S_{(i,j)}|} \sum_{A_{ij}^h \in S_{(i,j)}} A_{ij}^h \end{cases} . \quad (\text{B.6})$$

This reconstructed matrix is then symmetrized, corrected by the average-product correction, and used as a contact score to derive PPV curves and contact maps. By varying k , i.e. the number of retained elements from each head, one can measure how much of the attention information is actually required to reach a given level

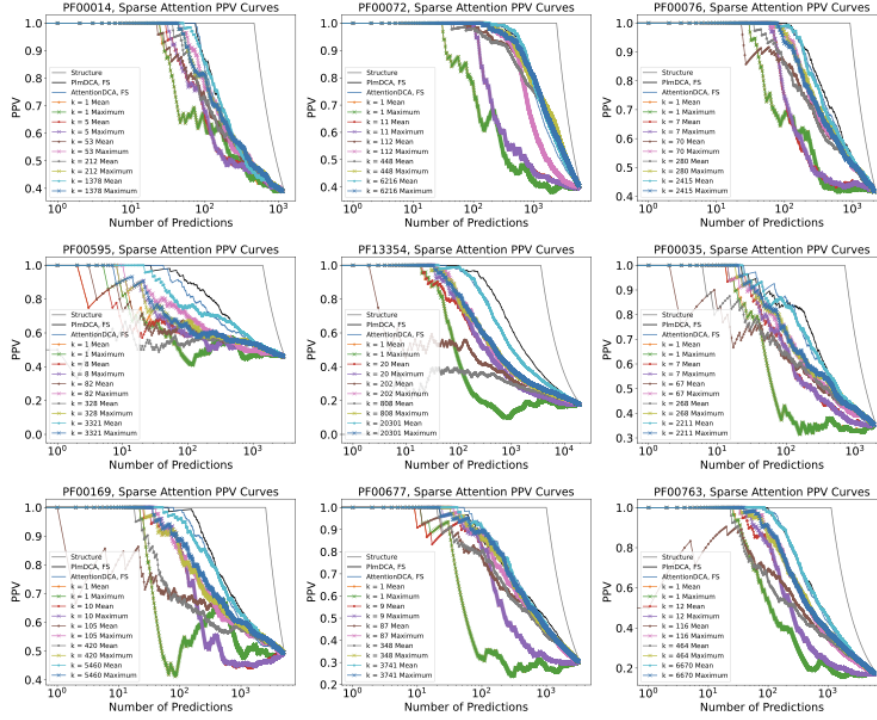


Fig. B.4 PPV curves of different protein families for the averaged and maximized sparse Attention Score computed for $k = \{1, L/10, L, 4L, (\frac{L}{2})\}$ and compared to the standard Frobenius Score (FS) from PfimDCA and AttentionDCA (black and blue curves respectively). Source [42]

of predictive accuracy. Figure B.5 reports the PPV evaluated at L and $2L$ for all the protein families considered, in comparison with the results obtained from the Frobenius norm score in AttentionDCA.

The results show that when the sparse matrix is built using the maximum rule, even a very small fraction of the total entries is sufficient to recover PPVs close to those obtained with the Frobenius norm. In the averaging case, the PPVs rise more slowly with k , but again for $k \ll k_{\max} = (\frac{L}{2})$ they reach the same level of accuracy. Thus, whether one uses the max- or average-based construction, a limited number of entries per head already carries enough information to match the contact prediction accuracy of the full Frobenius approach. Figure B.4 illustrates the complete PPV curves for a selected set of k values, namely $k = \{1, L/10, L, 4L, (\frac{L}{2})\}$. Additional visualizations, including the sparse heatmap at $k = L$, are reported in Figures B.10 and panel C in Figure 4.3 of the main text.

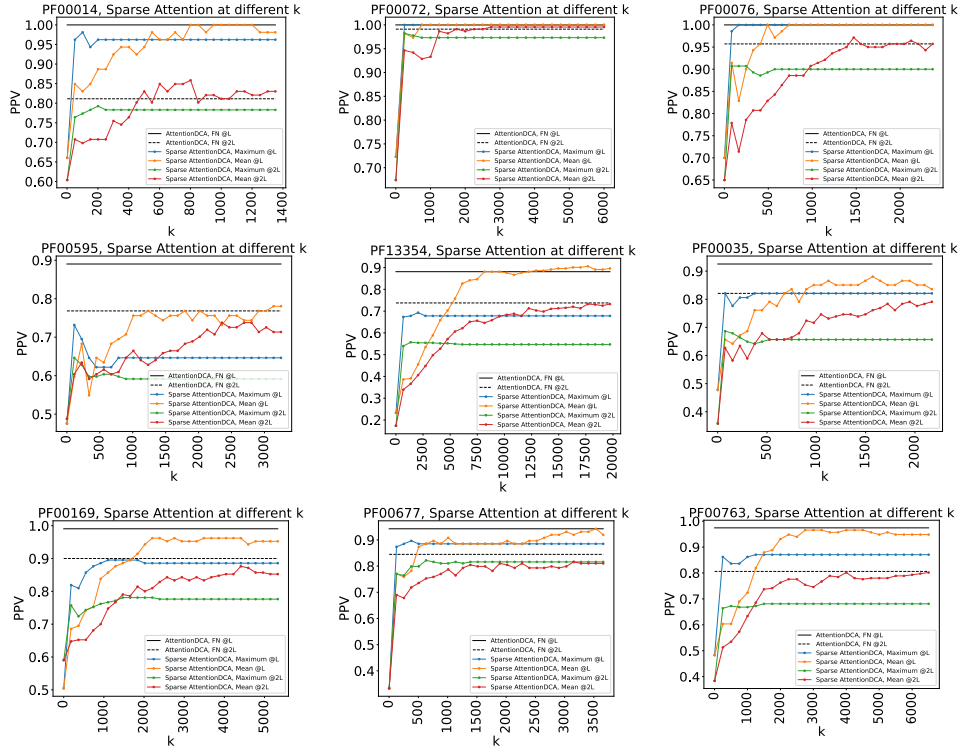


Fig. B.5 PPV curves @L and @2L for the averaged and maximized sparse Attention Score as the value of k varies. Blue and orange curves represent the PPV @L for the average- and maximum-sparse versions respectively, to be compared with the solid black line from the Frobenius Score. Similarly, green and red curves show the PPV @2L, to be compared with the dashed black line. Source [42]

B.3 Implementation details

B.3.1 Convergence Noise

Since the optimization landscape is highly non-convex and contains numerous local stationary points, independent training runs initialized with different random seeds can converge to slightly different solutions. To account for this variability, we adopt a multiple-run inference strategy. In this scheme, the final contact score for a pair (i, j) is defined as the maximum score obtained across all runs (see Section. 4.1.1 of the main text). Empirically, we find that a subset of the predicted contacts is reproduced consistently, regardless of initialization. As shown in Figure B.6, for every protein family considered, at least L of the top $2L$ predicted contacts are shared across the 20 independent runs, where L denotes the sequence length. This persistent backbone of contacts reflects the robust signal present in the MSA, while the remaining predictions are more sensitive to run-to-run fluctuations.

B.3.2 Learning Parameters

Table B.1 reports the training parameters used for both the standard and autoregressive versions of AttentionDCA. Unless stated otherwise, the same values were applied to both models. For all families, the mini-batch size was fixed at $b_s = 1000$, the compression ratio at $c_r = 0.20$, and the number of heads at $H = 128$. The number of training epochs and the regularization strength were tuned individually for each family in order to maximize contact prediction performance.

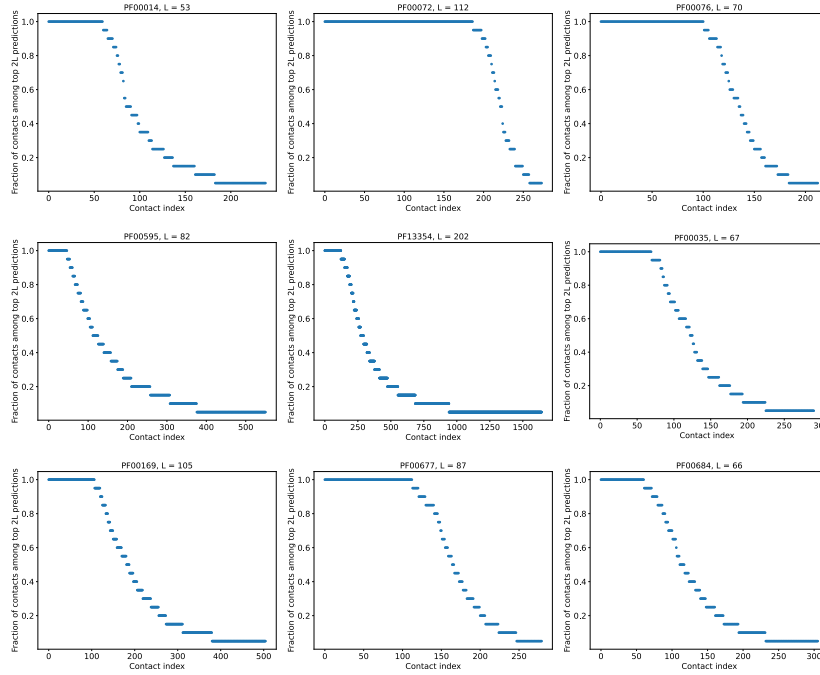


Fig. B.6 Frequency of the top $2L$ predicted contacts across 20 independent inference runs for nine protein families. Each bar indicates how often a given contact appears among the top $2L$ predictions. All families display a stable set of recurrent contacts (a "backbone") shared across runs, with PF00072 showing the largest and PF13354 the smallest backbone. Source [42]

Table B.1 Inference parameters and hyper-parameters of the model for each protein family. Source [42]

	H	d	λ Standard	λ Autoregressive	# epochs
PF00014	128	5	0.001	1.0×10^{-5}	100
PF00035	128	8	0.001	1.0×10^{-5}	100
PF00072	128	17	0.001	1.0×10^{-5}	20
PF00076	128	9	0.001	1.0×10^{-5}	100
PF00169	128	16	0.001	1.0×10^{-5}	100
PF00595	128	11	0.001	0.01	150
PF00677	128	12	0.001	1.0×10^{-5}	100
PF00763	128	18	0.001	1.0×10^{-5}	50
PF13354	128	34	0.001	0.01	200

B.4 Figures

In this section, we provide supplementary figures that extend the results to the full set of protein families considered in this study. Specifically, Figure B.7 complements the analysis of contact prediction for both the single- and multi-family versions of the model on the remaining families, extending the results shown in Figure 4.2 of the main text. Figures B.8, B.9, and B.10 reproduce for additional families the comparison between contact maps and attention maps presented in Figure 4.3. Figures B.8 and B.12 report the results obtained with the autoregressive model, illustrating contact prediction and generative performance respectively, in direct analogy with Figure 4.4 of the main text. Finally, Figure B.13 provides the Principal Component Analysis for the full dataset of protein families, expanding the results of Figure 4.5.

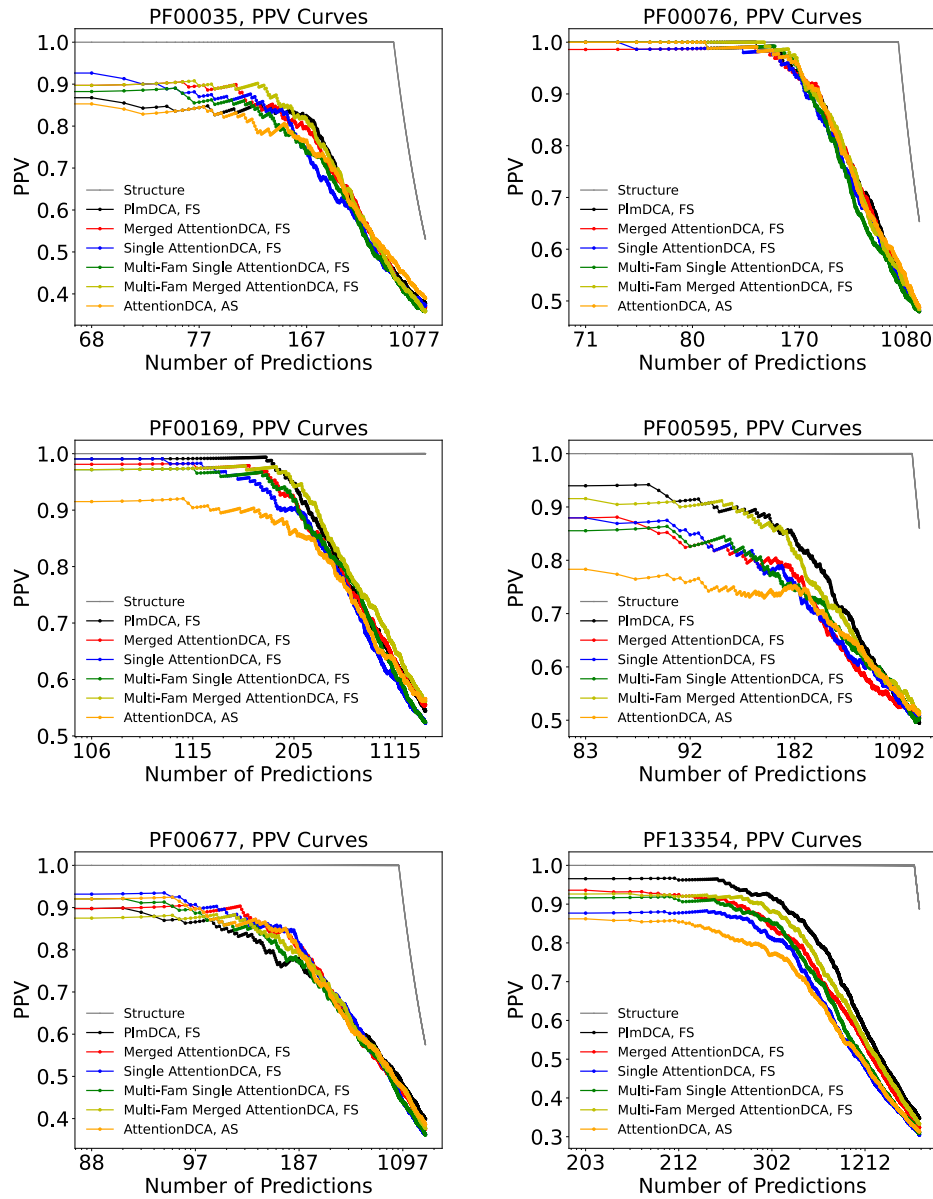


Fig. B.7 Positive Predicted Value Curves for different families. Each plot compares PPVs computed through the Frobenius Score (FS) and the Attention Score (AS), in particular: PlmDCA with FS (black curve); merged- and single-realization of AttentionDCA with FS (red and blue curves respectively); Multi-Family AttentionDCA with merged- and single-realization (green and yellow curves respectively); AttentionDCA with AS (orange curve). The gray curve represents the PPV of an ideal model that predicts all and only positive contacts, hence it depicts the prediction of the full structure of the family. Source [42]

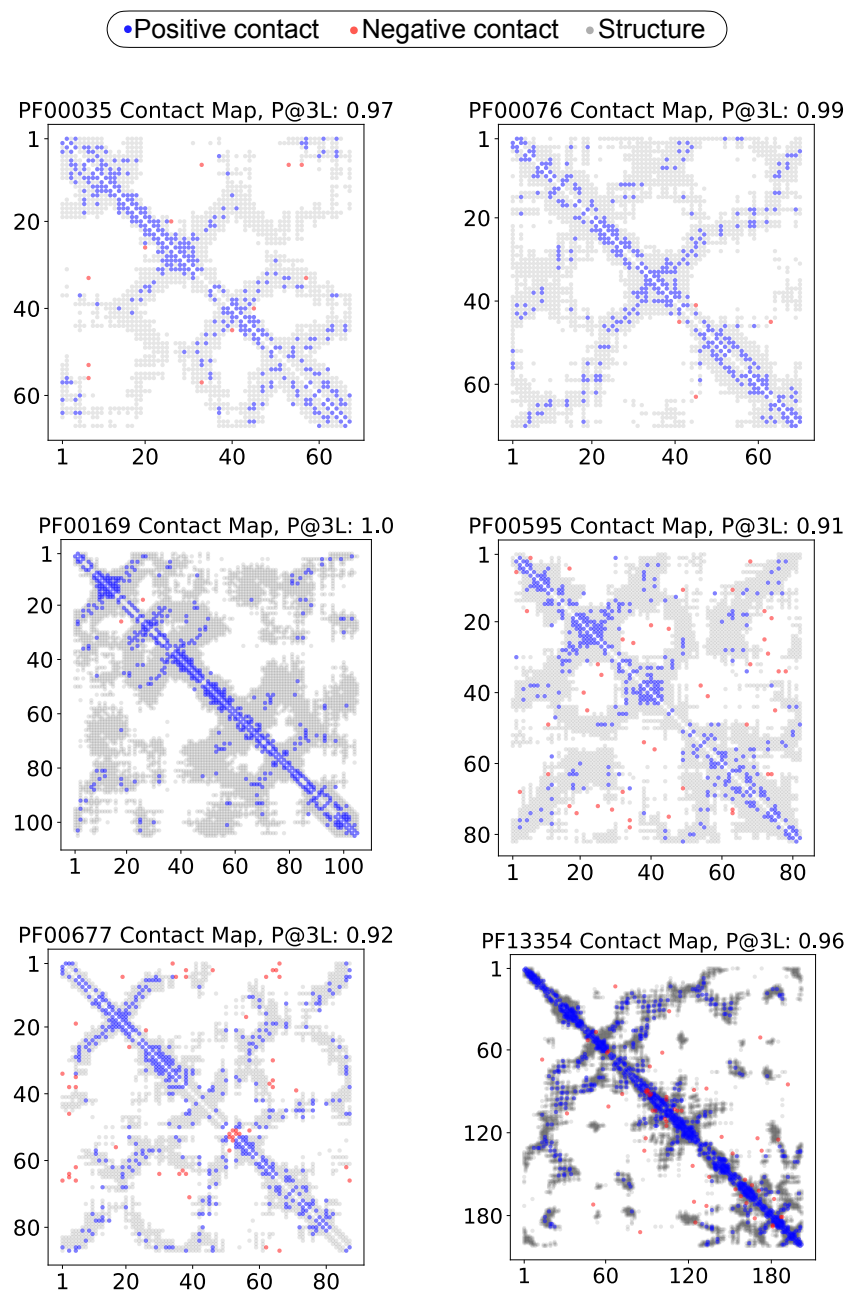


Fig. B.8 Contact map built from the Frobenius Score applied to the interaction tensor inferred by AttentionDCA. Blue and red dots represent positive and negative predictions respectively, while gray dots reproduce the structure of the family. Source [42]

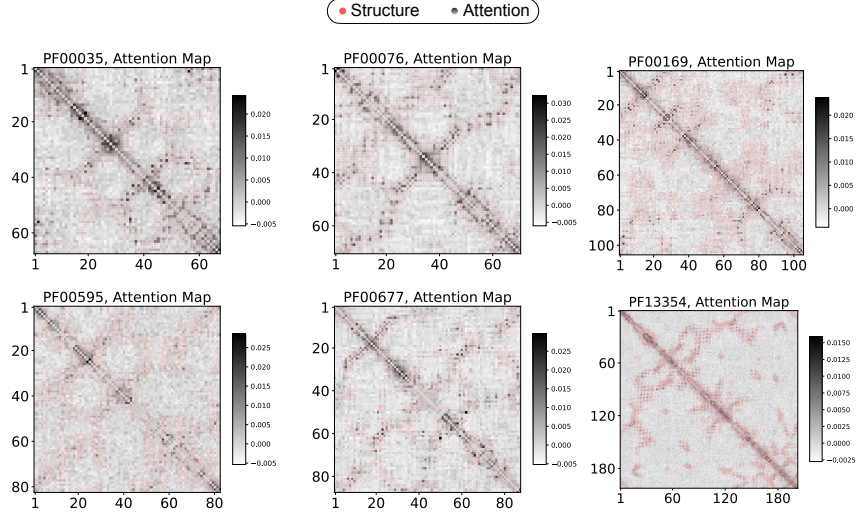


Fig. B.9 Grayscale heat map of the averaged attention matrix built from the $\{Q^h, K^h\}_{h=1, \dots, H}$ matrices by AttentionDCA. Structural information is depicted in red. Source [42]

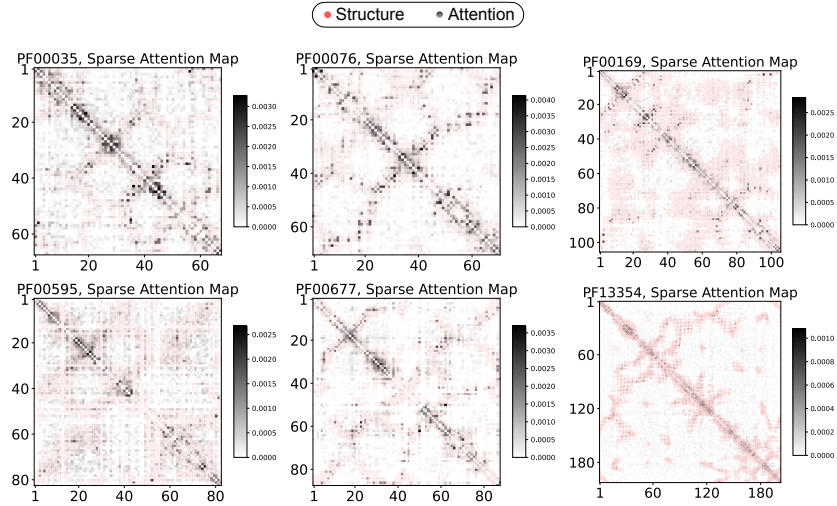


Fig. B.10 Grayscale heat map for the sparse attention matrix constructed by maximizing over the k highest elements from each head, using $k = L$. Structural information is depicted in red. Source [42]

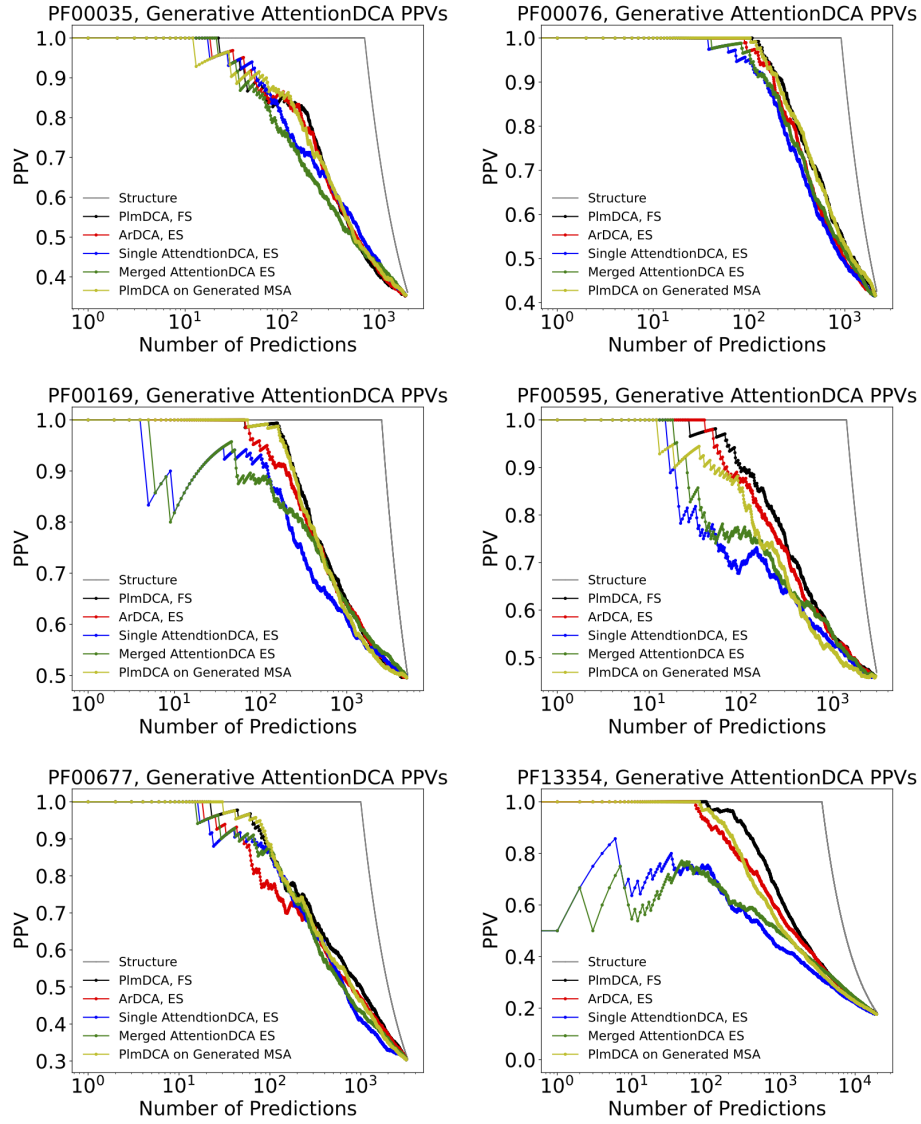


Fig. B.11 Contact prediction results from Autoregressive AttentionDCA applied to different families. ES stands for Epistatic Score, FS for Frobenius Score. The plots compare the PPV curves from PlmDCA (FS, black curve), ArDCA (ES, red curve) with the single- and merged-realizations of AttentionDCA with ES (blue and green curves respectively) and the PPV from the FS of PlmDCA applied to a Generated MSA sampled from Autoregressive AttentionDCA. The gray curve represents the PPV of an ideal model that predicts all and only positive contacts, hence it depicts the prediction of the full structure of the family. Source [42]

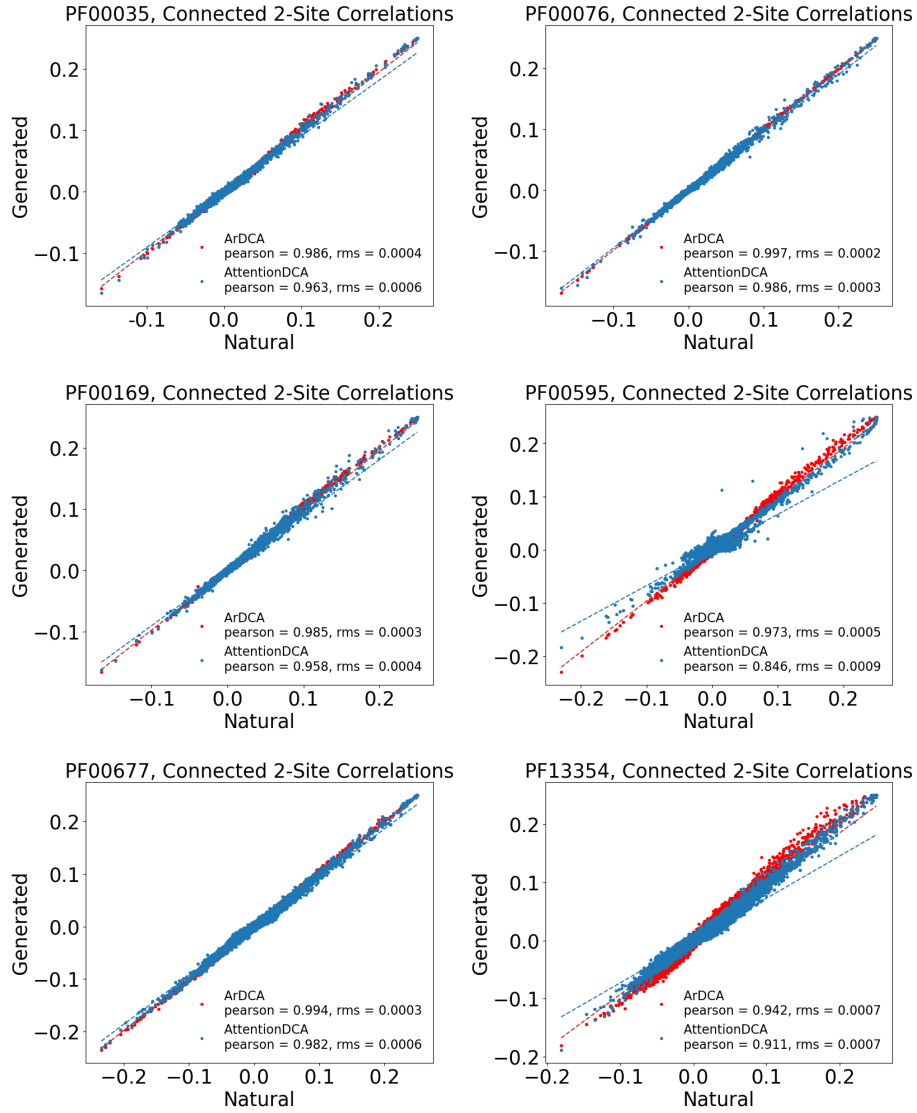


Fig. B.12 Generativity results from Autoregressive AttentionDCA applied to different families. The plots show a comparison between the two-site connected correlations of the natural and artificial MSAs generated by Autoregressive AttentionDCA and ArDCA (blue and red dots respectively). Source [42]

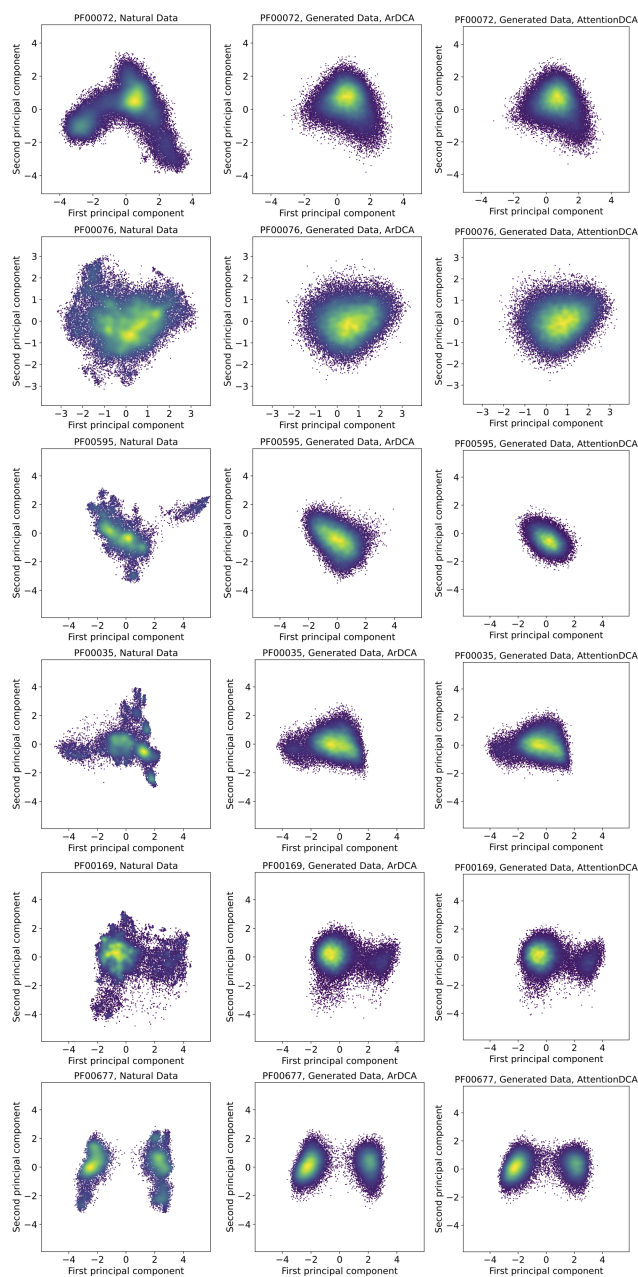


Fig. B.13 Principal Component Analysis on different protein families. Comparison between natural data and artificial data generated by using ArDCA and AttentionDCA. Source [42]

Appendix C

FeatureDCA: Supplementary Information

This appendix presents supplementary material derived from the Supporting Information of the paper “*Controllable Protein Design via Autoregressive Direct Coupling Analysis Conditioned on Principal Components*” [43], rephrased and reorganized for clarity within the context of this thesis.

C.1 Mathematical foundation of the feature-conditioned autoregressive model

Let us assume that we have an MSA of M one-hot encoded sequences of length L over an alphabet of q symbols ($q = 21$ for proteins). We can extract the rotation matrix $U \in \mathbb{R}^{d \times Lq}$, such that the application $U\mathbf{a} = \mathbf{y}$ is the projection of an amino-acid sequence $\mathbf{a}$ in the d -dimensional space spanned by the eigenvectors associated to the largest d eigenvalues of covariance matrix C^{emp} . Let us define the sequence

$$\begin{aligned} \mathbf{x}^\alpha &:= (\mathbf{a}_1^\alpha, \dots, \mathbf{a}_L^\alpha, y_1^\alpha, \dots, y_d^\alpha) := (\mathbf{a}^\alpha, \mathbf{y}^\alpha) \\ \begin{cases} \mathbf{a}^\alpha := (\mathbf{a}_1^\alpha, \dots, \mathbf{a}_L^\alpha) & \text{is an } Lq \text{ dimensional vector} \\ \mathbf{y}^\alpha = U\mathbf{a}^\alpha & \text{is a } d \text{ dimensional vector} \end{cases} \end{aligned}$$

where $\mathbf{a}_i$ is the q -dimensional one-hot encoded representation of amino-acid a_i . We aim at learning the joint probability distribution $P(\mathbf{a}, \mathbf{y})$ from our data. We note that

$$P(\mathbf{a}, \mathbf{y}) = P(\mathbf{y}|\mathbf{a})P(\mathbf{a}) = \underbrace{\delta^{(d)}(\mathbf{y} - U\mathbf{a})}_{P(\mathbf{y}|\mathbf{a})} \frac{e^{-H(\mathbf{a})}}{Z} \quad (\text{C.1})$$

where: (i) we assume in the last step that the prior distribution of $\mathbf{y}$ is uniform, $\delta^{(d)}(\mathbf{x}) := \prod_{i=1}^d \delta(x_i)$ is the d -dimensional Kronecker delta, and (ii) we model $P(\mathbf{a})$ as a Potts model defined by the energy function $H(\mathbf{a})$. The joint probability distribution $P(\mathbf{a}, \mathbf{y})$ is amenable to autoregressive representation:

$$P(\mathbf{a}, \mathbf{y}) = P(\mathbf{y})P(\mathbf{a}_1|\mathbf{y})P(\mathbf{a}_2|\mathbf{y}, \mathbf{a}_1) \cdots P(\mathbf{a}_L|\mathbf{y}, \mathbf{a}_1, \dots, \mathbf{a}_{L-1}) := P(\mathbf{y}) \prod_{i=1}^L P(\mathbf{a}_i|\mathbf{y}, \mathbf{a}_{<i})$$

where $\mathbf{a}_{<i} := \{a_1, \dots, a_{i-1}\}$ with the convention that $\mathbf{a}_{<1}$ is the empty set.

C.1.1 Gaussian Approximation

To proceed analytically, we make use of a Gaussian approximation for $P(\mathbf{a}) = \exp(-H)/Z$. Also, we relax the one-hot-encoded nature of the variables $\mathbf{a}_i$ to real values. In this case, we have that

$$P(\mathbf{a}) = \mathcal{N}(\mathbf{a}|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^{Lq} \det \Sigma}} e^{-\frac{1}{2}(\mathbf{a}-\mu)^\dagger \Sigma^{-1}(\mathbf{a}-\mu)} \quad (\text{C.2})$$

Eq. C.1 becomes:

$$P(\mathbf{a}, \mathbf{y}) = P(\mathbf{y}|\mathbf{a})P(\mathbf{a}) = \delta^{(d)}(\mathbf{y} - U\mathbf{a}) \mathcal{N}(\mathbf{a}|\mu, \Sigma)$$

We can compute the marginal $P(\mathbf{y})$ which is a Gaussian

$$P(\mathbf{y}) = \int d\mathbf{a} P(\mathbf{a}, \mathbf{y}) = \mathcal{N}(\mathbf{y} | U\mu, U\Sigma U^\dagger)$$

Computation of the marginal $P(\mathbf{y})$

Let us consider the case $P(\mathbf{y}|\mathbf{a}) = \delta^{(d)}(\mathbf{y}-U\mathbf{a})$. We have that

$$\begin{aligned}
P(\mathbf{y}) &= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int d\mathbf{a} e^{-\frac{1}{2}(\mathbf{a}-\mu)^\dagger \Sigma^{-1}(\mathbf{a}-\mu)} \delta^{(d)}(\mathbf{y}-U\mathbf{a}) \\
&= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int d\mathbf{a} \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}(\mathbf{a}-\mu)^\dagger \Sigma^{-1}(\mathbf{a}-\mu)} e^{i\mathbf{k}^\dagger(\mathbf{y}-U\mathbf{a})} \\
&= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int \frac{d\mathbf{k}}{(2\pi)^d} \int d\mathbf{a} e^{-\frac{1}{2}(\mathbf{a}-\mu)^\dagger \Sigma^{-1}(\mathbf{a}-\mu) + i\mathbf{k}^\dagger(\mathbf{y}-U\mathbf{a})} \\
&= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int \frac{d\mathbf{k}}{(2\pi)^d} \int d\mathbf{a} e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1} \mathbf{a} + \mu^\dagger \Sigma^{-1} \mathbf{a} - \frac{1}{2}\mu^\dagger \Sigma^{-1} \mu + i\mathbf{k}^\dagger \mathbf{y} - i\mathbf{k}^\dagger U \mathbf{a}} \\
&= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}\mu^\dagger \Sigma^{-1} \mu + i\mathbf{k}^\dagger \mathbf{y}} \int d\mathbf{a} e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1} \mathbf{a} + (\mu^\dagger \Sigma^{-1} - i\mathbf{k}^\dagger U) \mathbf{a}} \\
&= \frac{1}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}\mu^\dagger \Sigma^{-1} \mu + i\mathbf{k}^\dagger \mathbf{y}} \int d\mathbf{a} e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1} \mathbf{a} + (\Sigma^{-1} \mu - iU^\dagger \mathbf{k})^\dagger \mathbf{a}} \\
&= \frac{\sqrt{(2\pi)^{Lq} \det(\Sigma)}}{\sqrt{(2\pi)^{Lq} \det(\Sigma)}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}\mu^\dagger \Sigma^{-1} \mu + i\mathbf{k}^\dagger \mathbf{y} + \frac{1}{2}(\Sigma^{-1} \mu - iU^\dagger \mathbf{k})^\dagger \Sigma (\Sigma^{-1} \mu - iU^\dagger \mathbf{k})} \\
&= \underbrace{e^{-\frac{1}{2}\mu^\dagger \Sigma^{-1} \mu + \frac{1}{2}(\Sigma^{-1} \mu)^\dagger \Sigma (\Sigma^{-1} \mu)}}_{=1} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger \mathbf{y} - \frac{1}{2}(U^\dagger \mathbf{k})^\dagger \Sigma U^\dagger \mathbf{k} - i(\Sigma^{-1} \mu)^\dagger \Sigma U^\dagger \mathbf{k}} \\
&= \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}(U^\dagger \mathbf{k})^\dagger \Sigma U^\dagger \mathbf{k} + i(\mathbf{y}^\dagger - \mu^\dagger \Sigma^{-1} \Sigma U^\dagger) \mathbf{k}} \\
&= \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}\mathbf{k}^\dagger U \Sigma U^\dagger \mathbf{k} + i(\mathbf{y} - U\mu)^\dagger \mathbf{k}} \\
&= \frac{\sqrt{(2\pi)^d}}{(2\pi)^d \sqrt{\det(U \Sigma U^\dagger)}} e^{-\frac{1}{2}(\mathbf{y} - U\mu)^\dagger (U \Sigma U^\dagger)^{-1} (\mathbf{y} - U\mu)} \\
&= \frac{1}{\sqrt{(2\pi)^d \det(U \Sigma U^\dagger)}} e^{-\frac{1}{2}(\mathbf{y} - U\mu)^\dagger (U \Sigma U^\dagger)^{-1} (\mathbf{y} - U\mu)} \\
&= \mathcal{N}(\mathbf{y} | U\mu, U \Sigma U^\dagger)
\end{aligned}$$

Alternatively, we could have computed the same result by observing that $P(\mathbf{y})$ is Gaussian and its first two moments can be computed as:

$$\begin{aligned}\langle \mathbf{y} \rangle_{P(\mathbf{y})} &= \int d\mathbf{a} P(\mathbf{a}) \int d\mathbf{y} \delta^{(d)}(\mathbf{y} - U\mathbf{a}) \mathbf{y} = U\boldsymbol{\mu} \\ \left\langle \left(\mathbf{y} - \langle \mathbf{y} \rangle_{P(\mathbf{y})} \right) \left(\mathbf{y} - \langle \mathbf{y} \rangle_{P(\mathbf{y})} \right)^\dagger \right\rangle_{P(\mathbf{y})} &= \int d\mathbf{a} P(\mathbf{a}) \int d\mathbf{y} \delta^{(d)}(\mathbf{y} - U\mathbf{a}) \left(\mathbf{y} - \langle \mathbf{y} \rangle_{P(\mathbf{y})} \right) \left(\mathbf{y} - \langle \mathbf{y} \rangle_{P(\mathbf{y})} \right)^\dagger \\ &= U\Sigma U^\dagger\end{aligned}$$

In the following, we will use this technique extensively.

Computation of $P(\mathbf{a}|\mathbf{y})$ and its marginals

So far, we can write the conditioned distribution $P(\mathbf{a}|\mathbf{y})$ as

$$\begin{aligned}P(\mathbf{a}|\mathbf{y}) &= \frac{P(\mathbf{a}, \mathbf{y})}{P(\mathbf{y})} = \frac{\overbrace{\delta^{(d)}(\mathbf{y} - U\mathbf{a})}^{P(\mathbf{y}|\mathbf{a})} \overbrace{\mathcal{N}(\mathbf{a}|\boldsymbol{\mu}, \Sigma)}^{P(\mathbf{a})}}{\underbrace{\mathcal{N}(\mathbf{y}|U\boldsymbol{\mu}, U\Sigma U^\dagger)}_{P(\mathbf{y})}} \\ &= \sqrt{\frac{(2\pi)^d \det(U\Sigma U^\dagger)}{(2\pi)^{Lq} \det(\Sigma)}} e^{-\frac{1}{2}(\mathbf{a} - \boldsymbol{\mu})^\dagger \Sigma^{-1}(\mathbf{a} - \boldsymbol{\mu}) + \frac{1}{2}(\mathbf{y} - U\boldsymbol{\mu})^\dagger (U\Sigma U^\dagger)^{-1}(\mathbf{y} - U\boldsymbol{\mu})} \delta^{(d)}(\mathbf{y} - U\mathbf{a}) \\ &\propto e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1}\mathbf{a} + \boldsymbol{\mu}^\dagger \Sigma^{-1}\mathbf{a} + \frac{1}{2}\mathbf{y}^\dagger U\Sigma^{-1}U^\dagger \mathbf{y} + \boldsymbol{\mu}^\dagger \Sigma^{-1}U^\dagger \mathbf{y}} \delta^{(d)}(\mathbf{y} - U\mathbf{a})\end{aligned}$$

so that

$$P(\mathbf{a}|\mathbf{y}) = K e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1}\mathbf{a} + \boldsymbol{\mu}^\dagger \Sigma^{-1}\mathbf{a} + \frac{1}{2}\mathbf{y}^\dagger U\Sigma^{-1}U^\dagger \mathbf{y} + \boldsymbol{\mu}^\dagger \Sigma^{-1}U^\dagger \mathbf{y}} \delta^{(d)}(\mathbf{y} - U\mathbf{a})$$

where K is a normalization constant that does not affect the following computations.

First marginal The first marginal of the conditioned distribution $P(\mathbf{a}|\mathbf{y})$ can be computed as

$$\begin{aligned}
P(\mathbf{a}_1|\mathbf{y}) &= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y}} \int d\mathbf{a}_{-1} e^{-\frac{1}{2}\mathbf{a}^\dagger \Sigma^{-1} \mathbf{a} + \mu^\dagger \Sigma^{-1} \mathbf{a}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U\mathbf{a})} \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1} \int d\mathbf{a}_{-1} e^{-\frac{1}{2}\mathbf{a}_{-1}^\dagger (\Sigma^{-1})_{-1} \mathbf{a}_{-1} + \mu_{-1}^\dagger (\Sigma^{-1})_{-1} \mathbf{a}_{-1}} \times \\
&\quad \times \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U_1 \mathbf{a}_1 - U_{-1} \mathbf{a}_{-1})} = \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1} \times \\
&\quad \times \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i(\mathbf{y}^\dagger - \mathbf{a}_1^\dagger U_1^\dagger) \mathbf{k}} \int d\mathbf{a}_{-1} e^{-\frac{1}{2}\mathbf{a}_{-1}^\dagger (\Sigma^{-1})_{-1} \mathbf{a}_{-1} + ((\Sigma^{-1})_{-1} \mu_{-1} - iU_{-1}^\dagger \mathbf{k})^\dagger \mathbf{a}_{-1}} = \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1} \times \\
&\quad \times \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i(\mathbf{y}^\dagger - \mathbf{a}_1^\dagger U_1^\dagger) \mathbf{k} + \frac{1}{2}((\Sigma^{-1})_{-1} \mu_{-1} - iU_{-1}^\dagger \mathbf{k})^\dagger (\Sigma^{-1})_{-1} ((\Sigma^{-1})_{-1} \mu_{-1} - iU_{-1}^\dagger \mathbf{k})} = \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1} \times \\
&\quad \times \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i(\mathbf{y}^\dagger - \mathbf{a}_1^\dagger U_1^\dagger) \mathbf{k} + \frac{1}{2}\mu_{-1}^\dagger (\Sigma^{-1})_{-1} \mu_{-1} - \frac{1}{2}\mathbf{k}^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger \mathbf{k} - i\mu_{-1} U_{-1}^\dagger \mathbf{k}} = \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2}\mathbf{k}^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger \mathbf{k} + i(\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger \mathbf{k}} \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2}(\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger (U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger)^{-1} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)} \\
&= Ke^{\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2}(\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)}
\end{aligned}$$

The marginal distribution $P(\mathbf{a}_1|\mathbf{y})$ remains a Gaussian distribution that can be fully characterized by its mean and variance. The mean can be computed as the maximum of the argument of the exponential, while the variance as the opposite of the inverse of its second derivative with respect to the variable $\mathbf{a}_1$:

$$\begin{aligned}
\mu_{\mathbf{a}_1|\mathbf{y}} &:= \mathbf{a}_1^* = \arg \max_{\mathbf{a}_1} \left[\frac{1}{2}\mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2}\mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \right. \\
&\quad \left. + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2}(\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right]
\end{aligned}$$

$$\begin{aligned}\Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} := & -\frac{d^2}{d\mathbf{a}_1^2} \left[\frac{1}{2} \mathbf{y}^\dagger U \Sigma_a^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2} \mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \right. \\ & \left. + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right]\end{aligned}$$

Recalling the identities:

$$\frac{d\mathbf{x}^T \mathbf{a}}{d\mathbf{x}} = \frac{d\mathbf{a}^T \mathbf{x}}{d\mathbf{x}} = \mathbf{a}$$

$$\frac{d(\mathbf{B}\mathbf{x} + \mathbf{b})^T \mathbf{C} (\mathbf{D}\mathbf{x} + \mathbf{d})}{d\mathbf{x}} = \mathbf{B}^T \mathbf{C} (\mathbf{D}\mathbf{x} + \mathbf{d}) + \mathbf{D}^T \mathbf{C}^T (\mathbf{B}\mathbf{x} + \mathbf{b})$$

we have that the mean is given by:

$$\begin{aligned}\frac{d}{d\mathbf{a}_1} \left[\frac{1}{2} \mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2} \mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \right. \\ \left. + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma_a^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right] = \\ \frac{d}{d\mathbf{a}_1} \left[-\frac{1}{2} \mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \right. \\ \left. - \frac{1}{2} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right] = \\ -(\Sigma^{-1})_1 \mathbf{a}_1 + (\Sigma^{-1})_1 \mu_1 - (U_1)^T U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) = 0\end{aligned}$$

$$\begin{aligned}\implies & (\Sigma^{-1})_1 \mathbf{a}_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger U_1 \mathbf{a}_1 = \\ & = (\Sigma^{-1})_1 \mu_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1}) \\ \mu_{\mathbf{a}_1|\mathbf{y}}^* := & \mathbf{a}_1^* = \left[(\Sigma^{-1})_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger U_1 \right]^{-1} \times \\ & \times \left[(\Sigma^{-1})_1 \mu_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1}) \right] \\ & = \Sigma_{\mathbf{a}_1|\mathbf{y}} \left[(\Sigma^{-1})_1 \mu_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1}) \right] \in \mathbb{R}^{q,1}\end{aligned}$$

while the variance is:

$$\begin{aligned}
\Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} &:= -\frac{d^2}{d\mathbf{a}_1^2} \left[\frac{1}{2} \mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y} - \frac{1}{2} \mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 + \right. \\
&\quad \left. + \mu_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right] \\
&= -\frac{d^2}{d\mathbf{a}_1^2} \left[-\frac{1}{2} \mathbf{a}_1^\dagger (\Sigma^{-1})_1 \mathbf{a}_1 - \frac{1}{2} (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right] \\
&= -\frac{d}{d\mathbf{a}_1} \left[-(\Sigma_a^{-1})_1 \mathbf{a}_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1} - U_1 \mathbf{a}_1) \right] \\
&= (\Sigma^{-1})_1^\dagger - \left((U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger U_1 \right)^\dagger \\
&= (\Sigma^{-1})_1 - \left(U_1^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (U_1) \right) \in \mathbb{R}^{q,q}
\end{aligned}$$

So the marginal can be written as

$$\begin{aligned}
P(\mathbf{a}_1|\mathbf{y}) &= \frac{1}{\sqrt{(2\pi)^q \det(\Sigma_{1|\mathbf{y}})}} e^{-\frac{1}{2} (\mathbf{a}_1 - \mu_{\mathbf{a}_1|\mathbf{y}})^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} (\mathbf{a}_1 - \mu_{\mathbf{a}_1|\mathbf{y}})^T} = \\
&= \frac{e^{\mu_{\mathbf{a}_1|\mathbf{y}}^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} \mu_{\mathbf{a}_1|\mathbf{y}}}}{\sqrt{(2\pi)^q \det(\Sigma_{\mathbf{a}_1|\mathbf{y}})}} e^{-\frac{1}{2} \mathbf{a}_1^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} \mathbf{a}_1 + \mathbf{a}_1^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} \mu_{\mathbf{a}_1|\mathbf{y}}} \\
&= \frac{e^{\mu_{\mathbf{a}_1|\mathbf{y}}^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} \mu_{\mathbf{a}_1|\mathbf{y}}}}{\sqrt{(2\pi)^q \det(\Sigma_{\mathbf{a}_1|\mathbf{y}})}} e^{-\frac{1}{2} \mathbf{a}_1^T \Sigma_{\mathbf{a}_1|\mathbf{y}}^{-1} \mathbf{a}_1 + \mathbf{a}_1^T \left[(\Sigma^{-1})_1 \mu_1 - (U_1)^\dagger U_{-1} (\Sigma^{-1})_{-1} U_{-1}^\dagger (\mathbf{y} - U_{-1} \mu_{-1}) \right]}
\end{aligned}$$

In this context the notation $(\Sigma^{-1})_1 \in \mathbb{R}^{q,q}$ refers to the first block of matrix Σ^{-1} , while $(\Sigma^{-1})_{-1} \in \mathbb{R}^{q(L-1),q(L-1)}$ refers to the sub-matrix defined by removing the blocks relative to the first amino-acid. The same goes for $U_1 \in \mathbb{R}^{d,q}$ and $U_{-1} \in \mathbb{R}^{d,q(L-1)}$.

k -th marginal Generalizing the previous computation, we can compute the k -th marginal of the distribution $P(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y})$ conditioned to all previous positions and to

PCA components $\mathbf{y}$. Using Bayes' rule we can write:

$$\begin{aligned}
 P(\mathbf{a}|\mathbf{y}) &= \frac{P(\mathbf{a}, \mathbf{y})}{P(\mathbf{y})} \\
 &= \frac{P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) P(\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y})}{P(\mathbf{y})} \\
 &= \frac{P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) \prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y}) P(\mathbf{y})}{P(\mathbf{y})} \\
 &= P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) \prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y})
 \end{aligned}$$

$$P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) = \frac{P(\mathbf{a}|\mathbf{y})}{\prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y})}$$

From this, we can define the marginal as:

$$P(\mathbf{a}_k | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) = \int d\mathbf{a}_{>k} P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) = \int d\mathbf{a}_{>k} \frac{P(\mathbf{a}|\mathbf{y})}{\prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y})}$$

The actual computation is quite straightforward

$$\begin{aligned}
P(\mathbf{a}_k | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) &= \int d\mathbf{a}_{>k} P(\mathbf{a}_k, \dots, \mathbf{a}_L | \mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{y}) = \int d\mathbf{a}_{>k} \frac{P(\mathbf{a} | \mathbf{y})}{\prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y})} \\
&= \frac{1}{\prod_{i=1}^{k-1} P(\mathbf{a}_i | \mathbf{a}_{<i}, \mathbf{y})} \int d\mathbf{a}_{>k} P(\mathbf{a} | \mathbf{y}) = \\
&\propto \int d\mathbf{a}_{>k} e^{-\frac{1}{2} \mathbf{a}^\dagger \Sigma^{-1} \mathbf{a} + \mu^\dagger \Sigma^{-1} \mathbf{a} + \frac{1}{2} \mathbf{y}^\dagger U \Sigma^{-1} U^\dagger \mathbf{y} + \mu^\dagger \Sigma^{-1} U^\dagger \mathbf{y}} \delta^{(d)}(\mathbf{y} - U \mathbf{a}) \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k}} \int d\mathbf{a}_{>k} e^{-\frac{1}{2} \mathbf{a}_{>k}^\dagger \Sigma_{>k}^{-1} \mathbf{a}_{>k} + \mu_{>k}^\dagger \Sigma_{>k}^{-1} \mathbf{a}_{>k}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U \mathbf{a})} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k})} \int d\mathbf{a}_{>k} e^{-\frac{1}{2} \mathbf{a}_{>k}^\dagger \Sigma_{>k}^{-1} \mathbf{a}_{>k} + \left(\Sigma_{>k}^{-1} \mu_{>k} - i U_{>k}^\dagger \mathbf{k} \right)^\dagger \mathbf{a}_{>k}} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k}) + \frac{1}{2} \left(\Sigma_{>k}^{-1} \mu_{>k} - i U_{>k}^\dagger \mathbf{k} \right)^\dagger \Sigma_{>k} \left(\Sigma_{>k}^{-1} \mu_{>k} - i U_{>k}^\dagger \mathbf{k} \right)} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{i\mathbf{k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k}) - \frac{1}{2} \mathbf{k}^\dagger U_{>k} \Sigma_{>k} U_{>k}^\dagger \mathbf{k} - i \mu_{>k}^\dagger U_{>k}^\dagger \mathbf{k}} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k}} \int \frac{d\mathbf{k}}{(2\pi)^d} e^{-\frac{1}{2} \mathbf{k}^\dagger U_{>k} \Sigma_{>k} U_{>k}^\dagger \mathbf{k} + i (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})^\dagger \mathbf{k}} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \frac{1}{2} (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})^\dagger (U_{>k} \Sigma_{>k} U_{>k}^\dagger)^{-1} (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})} \\
&\propto e^{-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \frac{1}{2} (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})}
\end{aligned}$$

Now it is essential to isolate the terms depending on $\mathbf{a}_k$ alone in order to being able to compute the first and second derivatives to define mean and variance of the gaussian. In the following, indices i, j refer to block elements of the matrices they are applied to, while the term *constant* is used for terms that do not explicetely depend on $\mathbf{a}_k$. Let's analyze each term in the exponential of the marginal distribution:

$$\begin{aligned}
-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} &= -\frac{1}{2} \sum_{ij} (\mathbf{a}_{\leq k})_i \left(\Sigma_{\leq k}^{-1} \right)_{ij} (\mathbf{a}_{\leq k})_j = -\frac{1}{2} \mathbf{a}_k^\dagger (\Sigma^{-1})_k \mathbf{a}_k - \sum_{i=1}^{k-1} \mathbf{a}_i^\dagger (\Sigma^{-1})_{ik} \mathbf{a}_k + \text{constant} \\
\mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} &= \sum_{i=1}^k \mu_i^\dagger (\Sigma^{-1})_{ik} \mathbf{a}_k + \text{constant}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2} (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{> k} \boldsymbol{\mu}_{> k})^\dagger U_{> k} \Sigma_{a, > k}^{-1} U_{> k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{> k} \boldsymbol{\mu}_{> k}) = \\
& - \mathbf{y}^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_{\leq k} \mathbf{a}_{\leq k} + \frac{1}{2} \mathbf{a}_{\leq k}^\dagger U_{\leq k}^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_{\leq k} \mathbf{a}_{\leq k} + \mathbf{a}_{\leq k}^\dagger U_{\leq k}^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger \boldsymbol{\mu}_{> k} + \text{constant} = \\
& - \sum_{i=1}^k \mathbf{y}^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \mathbf{a}_i + \frac{1}{2} \mathbf{a}_k^\dagger U_k^\dagger U_{> k} \Sigma_{a, > k}^{-1} U_{> k}^\dagger U_k \mathbf{a}_k + \\
& + \sum_{i=1}^{k-1} \mathbf{a}_i^\dagger U_i^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_k \mathbf{a}_k + \sum_{i=1}^k \boldsymbol{\mu}_{> k}^\dagger \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \mathbf{a}_i + \text{constant}
\end{aligned}$$

Computing the first derivative with respect to $\mathbf{a}_k$ of the argument of the exponential and setting it equal to zero, we get:

$$\begin{aligned}
& \frac{d}{d\mathbf{a}_k} \left[-\frac{1}{2} \mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \boldsymbol{\mu}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \right. \\
& \left. + \frac{1}{2} (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{> k} \boldsymbol{\mu}_{> k})^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{> k} \boldsymbol{\mu}_{> k}) \right] = \\
& = -(\Sigma^{-1})_k \mathbf{a}_k - \sum_{i=1}^k \left(\mathbf{y}^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \right)^\dagger + \\
& + \sum_{i=1}^{k-1} \left[\left(\mathbf{a}_i^\dagger U_i^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_k \right)^\dagger + \left(\boldsymbol{\mu}_{> k}^\dagger \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \right)^\dagger - \sum_{i=1}^{k-1} \left(\mathbf{a}_i^\dagger (\Sigma^{-1})_{ik} \right)^\dagger \right] + U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_k \mathbf{a}_k \\
& - (\Sigma^{-1})_k \mathbf{a}_k - \sum_{i=1}^k U_i^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger \mathbf{y} + \\
& + \sum_{i=1}^{k-1} \left[U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{> k} \Sigma_{> k}^{-1} \boldsymbol{\mu}_{> k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] + U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_k \mathbf{a}_k = 0
\end{aligned}$$

which gives a mean $\mu_{\mathbf{a}_k | \mathbf{a}_{< k}, \mathbf{y}}$ given by:

$$\begin{aligned}
& \mu_{\mathbf{a}_k | \mathbf{a}_{< k}, \mathbf{y}} := \mathbf{a}_k^* = \\
& = \left[(\Sigma^{-1})_k - U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_k \right]^{-1} \left\{ \sum_{i=1}^{k-1} \left[U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{> k} \Sigma_{> k}^{-1} \boldsymbol{\mu}_{> k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] + \right. \\
& \left. - \sum_{i=1}^k U_i^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger \mathbf{y} \right\} = \\
& = \Sigma_{\mathbf{a}_k | \mathbf{a}_{< k}, \mathbf{y}} \left\{ \sum_{i=1}^{k-1} \left[U_k^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{> k} \Sigma_{> k}^{-1} \boldsymbol{\mu}_{> k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] - \sum_{i=1}^k U_i^\dagger U_{> k} \Sigma_{> k}^{-1} U_{> k}^\dagger \mathbf{y} \right\}
\end{aligned}$$

The second derivative returns the inverse of the covariance matrix

$$\begin{aligned}
\Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} &:= -\frac{d^2}{d\mathbf{a}_k^2} \left[-\frac{1}{2}\mathbf{a}_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \mu_{\leq k}^\dagger \Sigma_{\leq k}^{-1} \mathbf{a}_{\leq k} + \right. \\
&\quad \left. + \frac{1}{2}(\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k})^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger (\mathbf{y} - U_{\leq k} \mathbf{a}_{\leq k} - U_{>k} \mu_{>k}) \right] = \\
&= -\frac{d}{d\mathbf{a}_k} \left[-(\Sigma^{-1})_k \mathbf{a}_k - \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k} \mathbf{y} + \right. \\
&\quad \left. + \sum_{i=1}^{k-1} \left[U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] + U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_k \mathbf{a}_k \right] \\
&= (\Sigma^{-1})_k - U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_k
\end{aligned}$$

Having characterized the k -th marginal, we have a solid foundation to define the k -th term in the autoregressive decomposition of the full joint distribution $P(\mathbf{a}, \mathbf{y})$:

$$\begin{aligned}
P(\mathbf{a}_k|\mathbf{a}_{<k}, \mathbf{y}) &= \frac{1}{\sqrt{(2\pi)^q \det(\Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}})}} e^{-\frac{1}{2}(\mathbf{a}_k - \mu_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}})^\dagger \Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} (\mathbf{a}_k - \mu_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}})} \\
&= \frac{e^{\mu_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^\dagger \Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} \mu_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}}}{\sqrt{(2\pi)^q \det(\Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}})}} e^{-\frac{1}{2}\mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} \mathbf{a}_k + \mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} \mu_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}} \\
&\propto e^{-\frac{1}{2}\mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}^{-1} \mathbf{a}_k + \mathbf{a}_k^\dagger \left\{ \sum_{i=1}^{k-1} [U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k} - (\Sigma^{-1})_{ik} \mathbf{a}_i] - \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \mathbf{y} \right\}} \\
&= \frac{1}{Z_{\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y}}} e^{-H(\mathbf{a}_k|\mathbf{a}_{<k},\mathbf{y})}
\end{aligned}$$

where the energy function $H(\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y})$ can be written as:

$$\begin{aligned}
H(\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}) &= \frac{1}{2} \mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}}^{-1} \mathbf{a}_k + \\
&- \mathbf{a}_k^\dagger \left\{ \sum_{i=1}^{k-1} \left[U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \mathbf{a}_i + U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k} - (\Sigma^{-1})_{ik} \mathbf{a}_i \right] + \right. \\
&- \left. \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{a,>k}^{-1} U_{>k}^\dagger \mathbf{y} \right\} = \\
&= \frac{1}{2} \mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}}^{-1} \mathbf{a}_k + \mathbf{a}_k^\dagger \sum_{i=1}^{k-1} \left[(\Sigma^{-1})_{ik} - U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \right] \mathbf{a}_i + \\
&- \mathbf{a}_k^\dagger \sum_{i=1}^{k-1} U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k} + \mathbf{a}_k^\dagger \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{a,>k}^{-1} U_{>k}^\dagger \mathbf{y}
\end{aligned}$$

C.1.2 Discrete model

Now that we have a principled formulation of the energy function of the $\mathbf{k}$ -th term in the autoregressive decomposition, we can restore the discrete nature of the amino-acid data and produce an ansatz for a discrete version of the model in its autoregressive implementation. Let's consider each of the three contributions in the energy function:

1) Quadratic term in the amino-acid space

$$\frac{1}{2} \mathbf{a}_k^\dagger \Sigma_{\mathbf{a}_k | \mathbf{a}_{<k}, \mathbf{y}}^{-1} \mathbf{a}_k + \mathbf{a}_k^\dagger \sum_{i=1}^{k-1} \left[(\Sigma^{-1})_{ik} - U_k^\dagger U_{>k} \Sigma_{>k}^{-1} U_{>k}^\dagger U_i \right] \mathbf{a}_i$$

This can be thought of as an interaction term:

$$\mathbf{a}_{\leq k}^\dagger J_{\leq k} \mathbf{a}_{\leq k}$$

$$\begin{cases} \mathbf{a}_{\leq k} \in [0, 1]^{qk} & \text{one-hot sub-vector representing the first } k \text{ positions of the sequence} \\ J_{\leq k} \in \mathbb{R}^{qk, qk} & \text{interaction matrix composed of } q \times q \text{ blocks} \end{cases}$$

Using a *one-cold* representation where $\mathbf{a} \in \{0, 1, 2, \dots, q\}^L$, the same can be written as

$$\sum_{k < i} J_{ik}(a_i, a_k)$$

2) Linear term in the the amino-acid space

$$\mathbf{a}_k^\dagger \sum_{i=1}^{k-1} U_i^\dagger U_{>k} \Sigma_{>k}^{-1} \mu_{>k}$$

This can be easily interpreted as a local field. In a one-hot representation this is $\mathbf{a}_k^\dagger \mathbf{h}_k$ with $\mathbf{h}_k \in \mathbb{R}^q$, while in a *one-cold* representation this is

$$h_k(a_k).$$

3) Cross interaction term between the amino-acid space and the principal component space:

$$\mathbf{a}_k^\dagger \sum_{i=1}^k U_i^\dagger U_{>k} \Sigma_{a,>k}^{-1} U_{>k}^\dagger \mathbf{y}$$

This can be written as

$$\mathbf{a}_k^\dagger G \mathbf{y} \quad G \in \mathbb{R}^{q,d}$$

where G acts as an embedding mapping from the amino-acid space to the principal component space. In a *one-cold* representation, this is

$$G(a_k) \cdot \mathbf{y}$$

The full one-cold energy function thus reads:

$$H(a_k | a_{<k}, \mathbf{y}) = - \sum_{i < k} J_{ki}(a_k, a_i) - h_k(a_k) - \mathbf{y} \cdot \mathbf{G}_k(a_k)$$

and the probability distribution of the k -th term is

$$\begin{aligned} P(a_k | a_{<k}, \mathbf{y}) &= \frac{1}{Z_{a_k | a_{<k}, \mathbf{y}}} e^{\sum_{i < k} J_{ki}(a_k, a_i) + h_k(a_k) + \mathbf{y} \cdot \mathbf{G}_k(a_k)} \\ &= \frac{e^{\sum_{i < k} J_{ki}(a_k, a_i) + h_k(a_k) + \mathbf{y} \cdot \mathbf{G}_k(a_k)}}{\sum_{c=1}^q e^{\sum_{i < k} J_{ki}(c, a_i) + h_k(c) + \mathbf{y} \cdot \mathbf{G}_k(c)}} \\ Z_{a_k | a_{<k}, \mathbf{y}} &= \sum_{c=1}^q e^{\sum_{i < k} J_{ki}(c, a_i) + h_k(c) + \mathbf{y} \cdot \mathbf{G}_k(c)} \end{aligned}$$

Given a MSA $D \in \{1, \dots, q\}^{L,M} = \{a_i^m\}$ concatenated to a matrix $\mathbf{Y} \in \mathbb{R}^{d,M} = \{y_\alpha^m\}$ representing the principal components of each sequence in D , the likelihood of the

model can be written as

$$\mathcal{L} = \frac{1}{M} \sum_{m=1}^M \log P(\mathbf{a}^m, \mathbf{y}^m) = \frac{1}{M} \sum_{m=1}^M \left\{ \log P(\mathbf{y}^m) + \log P(a_1^m | \mathbf{y}^m) + \sum_{k=2}^L \log P(a_k^m | \mathbf{a}_{<k}^m, \mathbf{y}^m) \right\}$$

The first term of the likelihood can be thought of as an empirical frequency of the sequences in the PC space. Since it does not depend on the learning parameters of the model, it can be discarded.

$$\begin{aligned} \mathcal{L} &= \frac{1}{M} \sum_{m=1}^M \left\{ h_1(a_1^m) + G_1(a_1^m) \cdot \mathbf{y}^m + \log \sum_{c=1}^q e^{h_i(c) + \mathbf{y}^m \cdot G_1(c)} + \right. \\ &\quad \left. + \sum_{k=2}^L \left[\sum_{i < k} J_{ki}(a_k^m, a_i^m) + h_k(a_k^m) + \mathbf{y}^m \cdot G_k(a_k^m) - \log \sum_{c=1}^q e^{\sum_{i < k} J_{ki}(c, a_i^m) + h_k(c) + \mathbf{y}^m \cdot G_k(c)} \right] \right\} = \\ &= \frac{1}{M} \sum_{k=1}^L \sum_{m=1}^M \mathcal{L}_k^m(\mathbf{a}^m, \mathbf{y}^m | \{\mathbf{h}_k, J_{k,*}, G_k\}) \end{aligned}$$

Since the parameters relative to each site of the sequence are factorized, the full likelihood can be optimized in parallel. So, each single optimization problem is defined by the following objective function:

$$\begin{aligned} \mathcal{L}_k(D | \{\mathbf{h}_k, J_{k,*}, G_k\}) &= \\ &= \frac{1}{M} \sum_{m=1}^M \left\{ \sum_{i < k} J_{ki}(a_k^m, a_i^m) + h_k(a_k^m) + \mathbf{y}^m \cdot \mathbf{G}_k(a_k^m) - \log \sum_{c=1}^q e^{\sum_{i < k} J_{ki}(c, a_i^m) + h_k(c) + \mathbf{y}^m \cdot G_k(c)} \right\} \end{aligned}$$

C.2 Implementation Details

All numerical experiments and model development were carried out in the Julia programming language. The complete source code, together with documentation and scripts to reproduce all results, is publicly available at <https://github.com/francescocaredda/FeatureDCA.jl>.

Model training was performed by minimizing the negative log-likelihood of the conditional distributions defined at each sequence position k , following the autoregressive factorization introduced in Section 6.2. Optimization was carried out using deterministic gradient descent with the LD_LBFGS algorithm from the NLOpt.jl package, with a convergence tolerance of 10^{-5} . An L_2 regularization

term was applied to the local fields $h_i(a)$, to the coupling tensors $J_{ij}(a, b)$, and to the feature embedding matrix G , with a single regularization coefficient of $\lambda = 10^{-4}$ used for all terms. Principal component analysis (PCA) was conducted using the `MultivariateStats.jl` library. Multiple sequence alignments (MSAs) were one-hot encoded and centered prior to PCA, and the top d principal components were retained as conditioning inputs during both training and generation. Thanks to the autoregressive formulation, both training and sampling can be parallelized across sequences or positional blocks, which substantially reduces the computational cost. Sinkhorn divergences were computed through a custom implementation of the Sinkhorn–Knopp algorithm included in the `OptimalTransport.jl` package.

C.3 Data Processing and MSA Construction

The analyses presented in Section 6.4 were performed using the multiple sequence alignments (MSAs) made available in the arDCA benchmark dataset, accessible at <https://github.com/pagnani/arDCADData>. These alignments provide a common reference for evaluating generative models on standardized protein families.

For the analysis of the Response Regulator (RR) family (Pfam ID PF00072), a new multiple sequence alignment was constructed to capture the structural diversity of the main functional subclasses. All sequences containing the conserved receiver domain PF00072 in combination with one of the three DNA-binding domains that define the principal RR subclasses, PF00486 (Trans_Reg_C), PF04397 (LytTR), and PF00196 (GerE), were retrieved from UniProt. This search yielded a total of 160,585 non-redundant sequences. To ensure structural relevance and consistency with experimentally characterized RRs, the 200 UniProt sequences most similar to the crystallographic representatives 1NXS (Trans_Reg_C), 4CBV (LytTR), and 4ZMS (GerE) were identified using the `jackhmmer` search tool. These sequences were used to build a profile Hidden Markov Model (HMM) with `hmmbuild`, which was then employed as the reference for sequence alignment.

The resulting HMM spanned 118 positions, slightly extending the 111-position profile from the Pfam database to include regions relevant to dimerization and structural variability. All RR sequences were aligned to this profile using `hmmalign`. Insertions relative to the consensus were removed to maintain positional consistency, producing a clean, fixed-length alignment suitable for statistical modeling. This

MSA served as the training dataset for the generative and conditional experiments described in Section 6.4, particularly in the case study on RR homodimers. All data used in this analysis are publicly available at <https://github.com/francescocaredda/FeatureDCADData>.

C.4 Wasserstein distance and Sinkhorn divergence

Comparing the projections of natural and generated sequences in principal component (PC) space provides an intuitive way to assess the fidelity of a generative model. Visual inspection of two-dimensional projections often gives a first indication of whether the generated sequences reproduce the overall geometry, clustering, and density of the natural distribution. To complement this qualitative comparison with a quantitative and geometrically meaningful measure, we rely on the Wasserstein distance, a concept originating from Optimal Transport theory, which defines a distance between two probability distributions over a metric space.

Let us consider two empirical distributions of points $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_M\}$, both embedded in the same Euclidean space $\mathbb{R}^N$, equipped with the standard Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}.$$

The associated cost matrix $\mathbf{C} \in \mathbb{R}^{M \times M}$ is defined as $C_{ij} = d(\mathbf{x}_i, \mathbf{y}_j)^2$, representing the squared cost of transporting unit probability mass from $\mathbf{x}_i$ to $\mathbf{y}_j$. We then define uniform discrete measures over the two point clouds as

$$\mu = \frac{1}{M} \sum_{i=1}^M \delta_{\mathbf{x}_i}, \quad \nu = \frac{1}{M} \sum_{j=1}^M \delta_{\mathbf{y}_j},$$

where $\delta_{\mathbf{x}}$ denotes the Dirac delta centered at point $\mathbf{x}$.

The Wasserstein distance of order two, also known as the Earth Mover's Distance, is defined as the minimum total cost of transporting mass from μ to ν , subject to

conservation of probability:

$$W_2^2(\mu, \nu) = \min_{\gamma \in \Gamma(\mu, \nu)} \sum_{i=1}^M \sum_{j=1}^M \gamma_{ij} C_{ij},$$

where $\Gamma(\mu, \nu)$ is the set of all couplings (joint probability distributions) with marginals μ and ν . The optimal transport plan γ specifies how probability mass is redistributed, and the term $\gamma_{ij} C_{ij}$ represents the cost of moving mass from $\mathbf{x}_i$ to $\mathbf{y}_j$.

Although the Wasserstein distance offers a geometrically grounded comparison between distributions, its exact computation on empirical datasets becomes increasingly demanding as the number of samples grows. To overcome these limitations, we employ the *Sinkhorn divergence*, a regularized variant of the Wasserstein distance that maintains its geometric interpretation while improving numerical stability and computational efficiency.

Given two discrete measures μ and ν and the cost matrix $\mathbf{C}$, the entropically regularized optimal transport cost is defined as

$$\text{OT}_\varepsilon(\mu, \nu) = \min_{\gamma \in \Gamma(\mu, \nu)} \sum_{i,j} \gamma_{ij} C_{ij} + \varepsilon KL(\gamma || \mu \otimes \nu),$$

where $KL(\gamma || \mu \otimes \nu) = \sum_{i,j} \gamma_{ij} \log \frac{\gamma_{ij}}{\mu_i \nu_j}$ is the Kullback–Leibler divergence that penalizes deviations of γ from the independent coupling $\mu \otimes \nu$, and $\varepsilon > 0$ controls the trade-off between accuracy and smoothness of the transport plan.

The regularized cost, however, does not define a true distance, as it lacks symmetry and does not necessarily vanish when $\mu = \nu$. The *Sinkhorn divergence* corrects this by symmetrizing and debiasing the regularized cost:

$$S_\varepsilon(\mu, \nu) = \text{OT}_\varepsilon(\mu, \nu) - \frac{1}{2} [\text{OT}_\varepsilon(\mu, \mu) + \text{OT}_\varepsilon(\nu, \nu)].$$

This formulation restores the key metric properties: $S_\varepsilon(\mu, \nu) = 0$ if and only if $\mu = \nu$, while retaining the geometry of the Wasserstein space. In practice, the Sinkhorn divergence can be efficiently computed through iterative matrix-scaling algorithms such as the Sinkhorn–Knopp procedure.

In this work, we employ the Sinkhorn divergence with a squared Euclidean cost and uniform weights over all samples. Computations were performed using a custom implementation in Julia, included in the `OptimalTransport.jl` package, with the

regularization parameter set to $\varepsilon = 0.05$. This metric provides a sensitive, geometry-aware measure of the similarity between the empirical PC-space distributions of natural and generated sequences.

C.5 Principal components beyond the second

Traditional autoregressive and energy-based models such as arDCA and bmDCA are trained without explicit access to global, low-dimensional representations of the sequence distribution. Although these models successfully reproduce local statistics and capture the dominant coevolutionary modes, they generally lack the ability to represent higher-order variability present in the data. In practice, this limitation implies that while arDCA and bmDCA can approximate the major axes of variation, typically the first and second principal components, they fail to recover finer patterns distributed along intermediate and higher components. These components often correspond to subtle structural or functional distinctions within a protein family, and thus carry biologically relevant information that remains inaccessible to unconditioned models.

FeatureDCA addresses this limitation by introducing a controllable conditioning mechanism based on the principal components of the training alignment. By including an adjustable number of components as conditioning variables, the model explicitly learns to represent and generate sequences along increasingly complex directions of evolutionary variability. This flexibility allows FeatureDCA to match the natural sequence distribution not only in its most dominant modes but also across intermediate and higher components, depending on the dimensionality d used during training.

Figure C.1 illustrates this effect for the beta-lactamase family PF13354. The figure compares the projections of natural and generated sequences across the first ten principal components, displayed as pairwise plots (PC1 vs PC2, PC3 vs PC4, and so on up to PC9 vs PC10). Sequences generated by arDCA and bmDCA align well with the natural data only along the first projection, whereas they diverge markedly in higher component pairs. In contrast, FeatureDCA, when trained with an increasing number of conditioning components ($d = 2, 4, 8, 16$), progressively improves its agreement with the natural data across all projections. This behavior demon-

strates that the model captures structured variability at multiple scales, enabling it to reproduce both global and fine-grained features of sequence space.

C.6 In-silico Deep Mutational Scanning

The ability of a generative model to predict the effect of point mutations provides an important measure of its capacity to capture local evolutionary and functional constraints. To test this property, we performed an *in-silico* deep mutational scanning (DMS) experiment using FeatureDCA. For each mutation at position i , replacing residue a with residue b , the predicted mutational effect is computed as the change in log-likelihood between the mutant and the wild-type sequence:

$$\Delta E(a_i \rightarrow b_i) = -\log P(\mathbf{a}^{wt}, \mathbf{y}) + \log P(\mathbf{a}^{mut}, \tilde{\mathbf{y}}) = -\log \frac{P(a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_L, \tilde{\mathbf{y}})}{P(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_L, \mathbf{y})},$$

where $\mathbf{a}^{wt}$ is the wild-type sequence and $\mathbf{a}^{mut}$ is the same sequence with residue a_i replaced by b_i . The vectors $\mathbf{y}$ and $\tilde{\mathbf{y}}$ denote the principal-component projections of the wild-type and mutant sequences, respectively. The probabilities $P(\mathbf{a}, \mathbf{y})$ are obtained from the model's autoregressive factorization.

Experimental DMS data were taken from the publicly available supplementary material of Riesselman *et al.*, specifically from the `blat_ecoli` dataset corresponding to the TEM-1 beta-lactamase experiment by Ostermeier and collaborators. To ensure consistency between model predictions and experimental values, the dataset was preprocessed as follows. All amino-acid substitutions at each site were alphabetically ordered (A, C, D, E, ..., Y) for reproducibility. Missing single-site mutations were included explicitly as entries with a placeholder score of `inf`, representing the absence of experimental data. A multiple sequence alignment (MSA) for the beta-lactamase family (PF13354) was then constructed with a fixed length of 214 positions, including seven alignment gaps, ensuring that the wild-type sequence used in the Ostermeier experiments was present in the alignment. The resulting dataset is available at <https://github.com/francescocaredda/FeatureDCADData>.

This processed alignment was used to train FeatureDCA and to compute the *in-silico* mutational scores. When comparing predictions to experimental DMS measurements, we excluded mutations corresponding to missing data (entries labeled

as $\inf$) as well as trivial identity mutations ($a \rightarrow a$), which would otherwise inflate the correlation artificially.

Figure C.2 reports the Spearman rank correlation between predicted and experimental mutational effects as a function of the number of principal components d used for conditioning. For small values of d , FeatureDCA performs comparably to the unconditioned autoregressive baseline (arDCA). As d increases, a non-monotonic trend emerges: the correlation initially decreases, indicating that excessive conditioning may obscure local mutational signals, but recovers at higher dimensionalities, suggesting that the model progressively stabilizes as it incorporates additional global variability. This pattern points to a balance between representational capacity and overfitting, characteristic of conditioning in limited data regimes.

C.7 Supplementary Figures

The figures presented in this section complement the analyses reported in Chapter 6, providing additional results on the conditioned generativity of FeatureDCA. While the main text focuses on the beta-lactamase family PF13354, the supplementary figures below extend the same set of analyses to four additional protein families: PF00014, PF00072, PF00076, and PF00595. Together, these results illustrate the generality of the conclusions drawn from the main study, demonstrating that the behaviors observed for PF13354 hold consistently across diverse protein families.

The figures include comparisons of principal component (PC) projections, pairwise correlation statistics, effective sequence diversity, and structural similarity between natural and generated sequences under different conditioning schemes:

- **Figure C.3** reports the Pearson correlation between the connected pairwise correlations of natural and generated alignments, as well as the effective sequence diversity (M_{eff}) of the generated MSAs. Both quantities are shown as a function of the number of principal components used during training and generation. These plots complement Figure 6.1C–D of the main text.
- **Figure C.4** shows the Wasserstein distances between the PCA-space distributions of natural and generated sequences for different models, computed using the Sinkhorn divergence introduced in Appendix C.4. This analysis parallels the PCA matching results presented in Figure 6.1B of the main text.

- **Figure C.5** provides the two-dimensional projections of natural and generated sequences onto the first two principal components for all four protein families, extending the analysis shown in Figure 6.1A of the main text.
- **Figure C.6** presents the sequence-level analysis of conditional sampling, including Hamming and PCA-space distances between generated sequences and their corresponding wild-type targets. This figure generalizes the analysis of sequence diversity discussed in Figure 6.2 of the main text.
- **Figure C.7** reports the structural validation of generated sequences. The RMSDs between predicted and experimental wild-type structures are shown for each family as a function of the conditioning dimensionality, complementing Figure 6.3 of the main text.

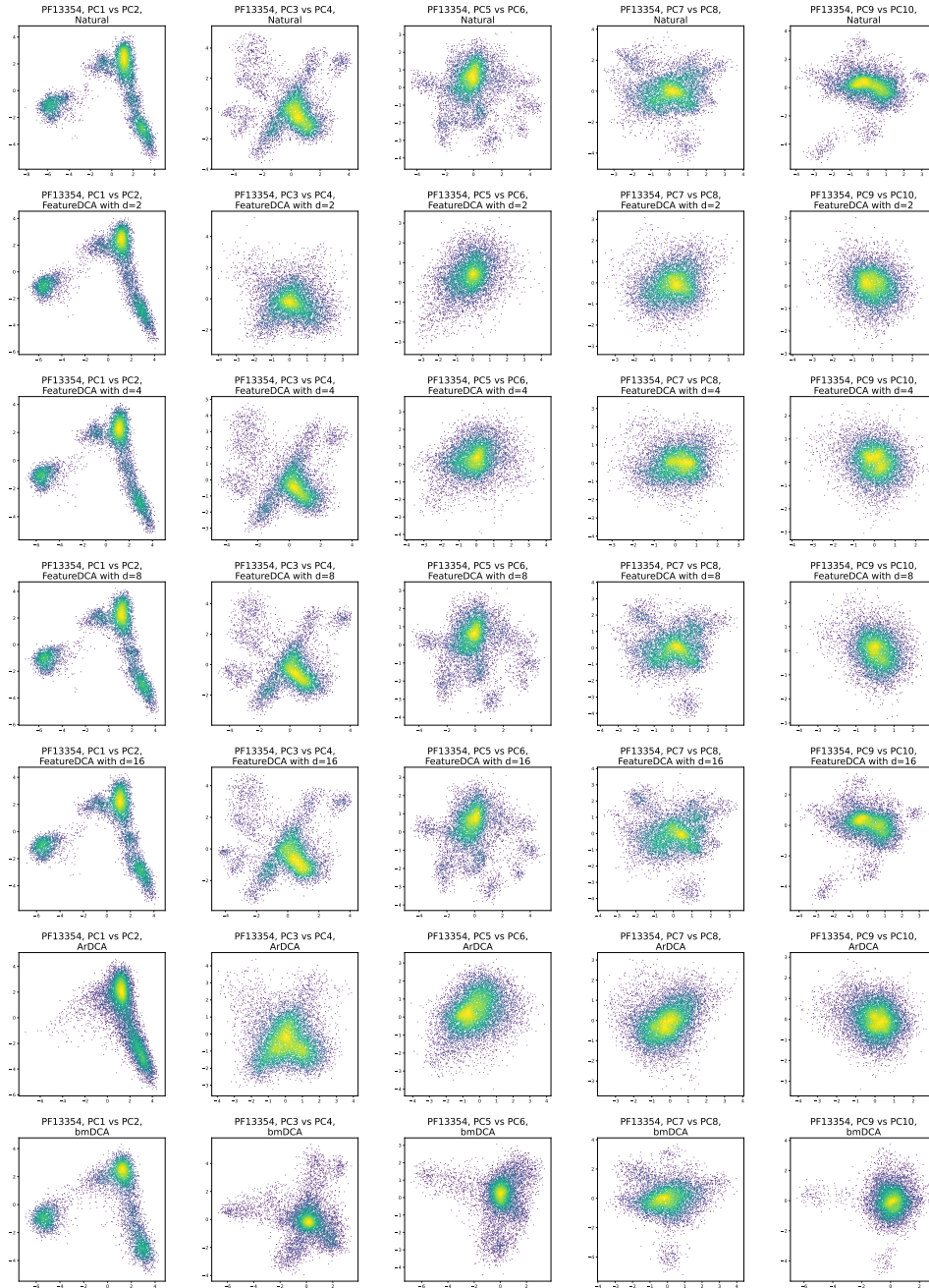


Fig. C.1 Principal component projections of natural and generated sequences for PF13354. Each column shows projections onto successive pairs of principal components: PC1 vs PC2, PC3 vs PC4, and so on up to PC9 vs PC10. The first row corresponds to the natural MSA, while subsequent rows show sequences generated by FeatureDCA trained with $d = 2, 4, 8, 16$ principal components, and by arDCA and bmDCA. FeatureDCA reproduces the natural sequence distribution within the first d principal components used during training, achieving progressively improved correspondence as d increases. Source [43]

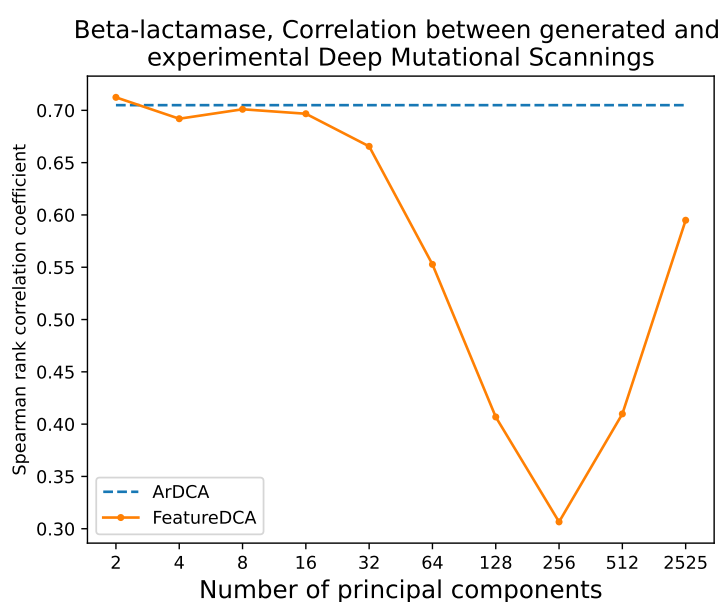


Fig. C.2 Predictive performance of FeatureDCA and arDCA on experimental deep mutational scanning data for beta-lactamase (PF13354). The Spearman rank correlation coefficient between predicted and experimental mutational effects is shown as a function of the number of principal components used during training and generation. arDCA corresponds to the unconditioned autoregressive baseline. FeatureDCA matches the performance of arDCA for small d , while exhibiting a non-monotonic dependence on the conditioning dimensionality: correlations first decline as more components are introduced, then increase again at larger d , indicating a trade-off between dimensionality, signal strength, and generalization. Source [43]

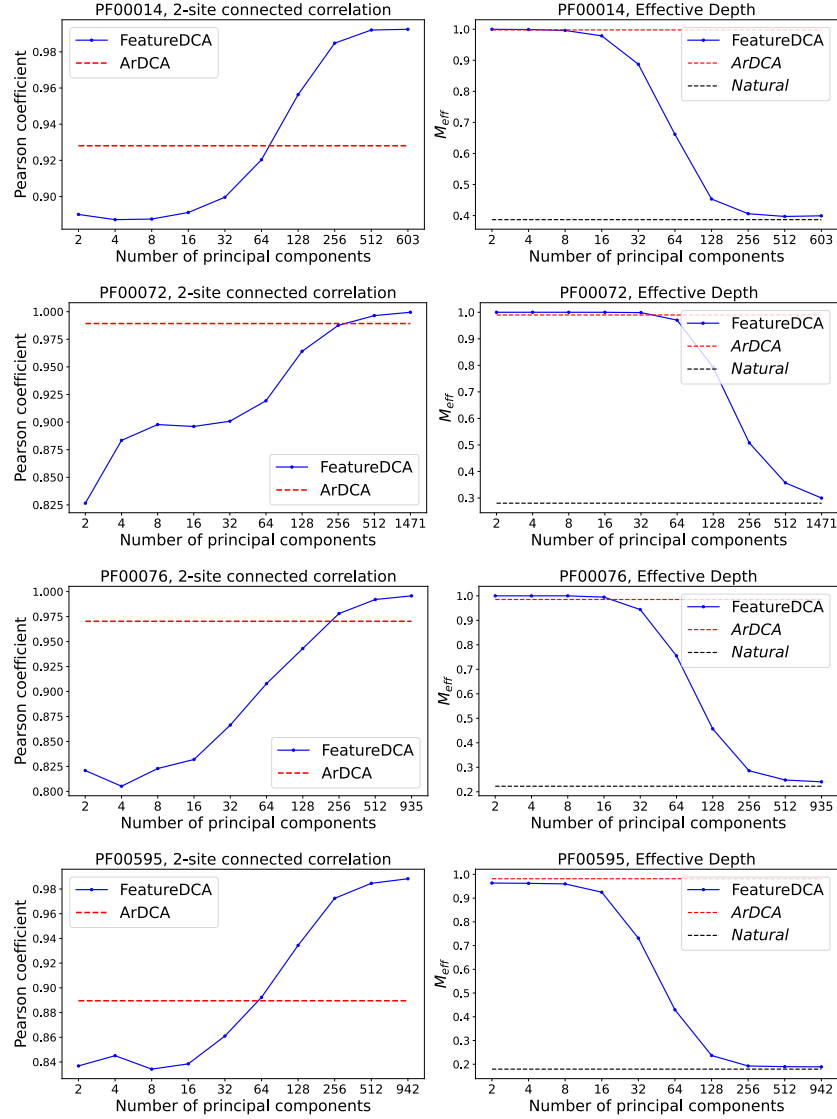


Fig. C.3 Comparison of statistical properties across protein families as a function of PCA feature dimensionality. Each row corresponds to a different protein family (PF00014, PF00072, PF00076, PF00595). **Left:** Pearson correlation between the pairwise connected correlations of natural and generated sequences, plotted as a function of the number of principal components used during training and generation (blue line: FeatureDCA; red dotted line: arDCA baseline). **Right:** Effective sequence depth (M_{eff}) of the generated datasets (blue line: FeatureDCA; red dotted line: arDCA; black dotted line: natural MSA). Source [43]

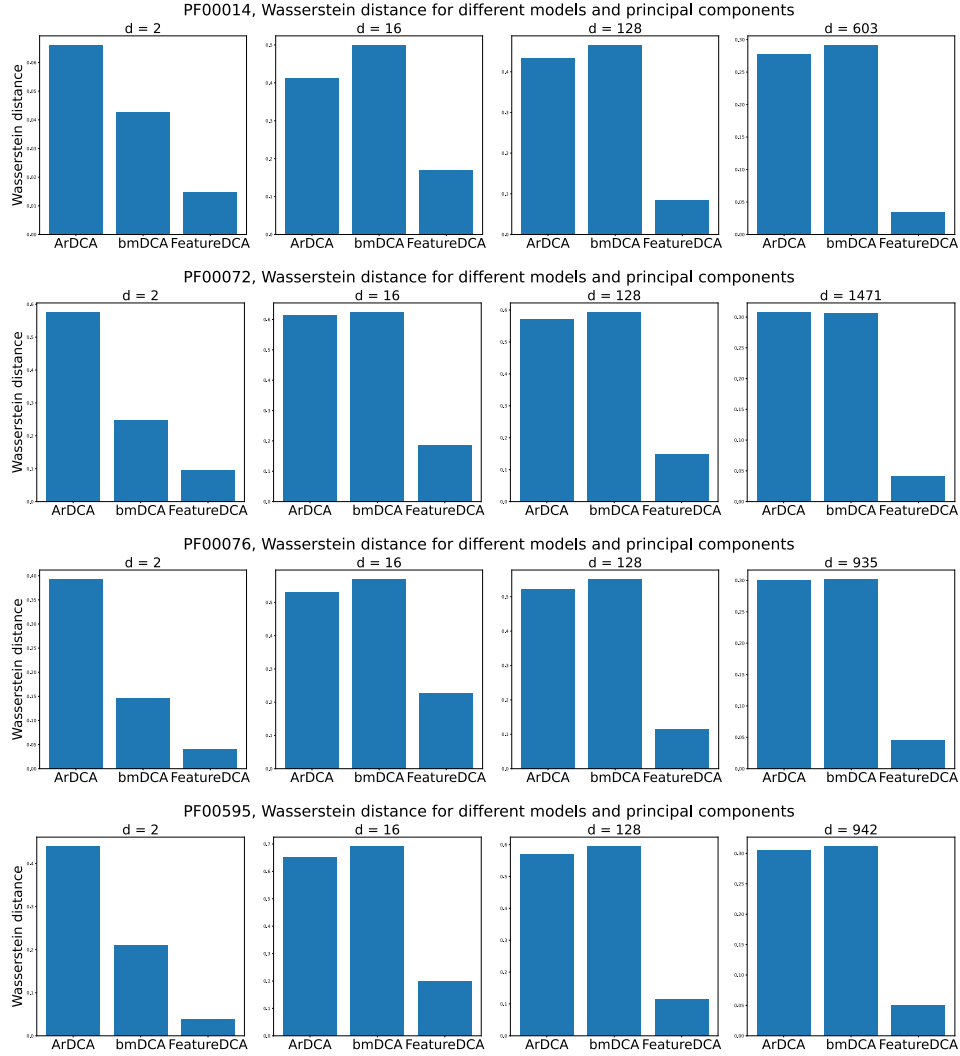


Fig. C.4 Wasserstein distance between the PCA-space distributions of natural and generated MSAs for different models. Each row corresponds to a different protein family (PF00014, PF00072, PF00076, PF00595). FeatureDCA was trained with the corresponding number of principal components used to compute the d -dimensional Wasserstein distance. Source [43]

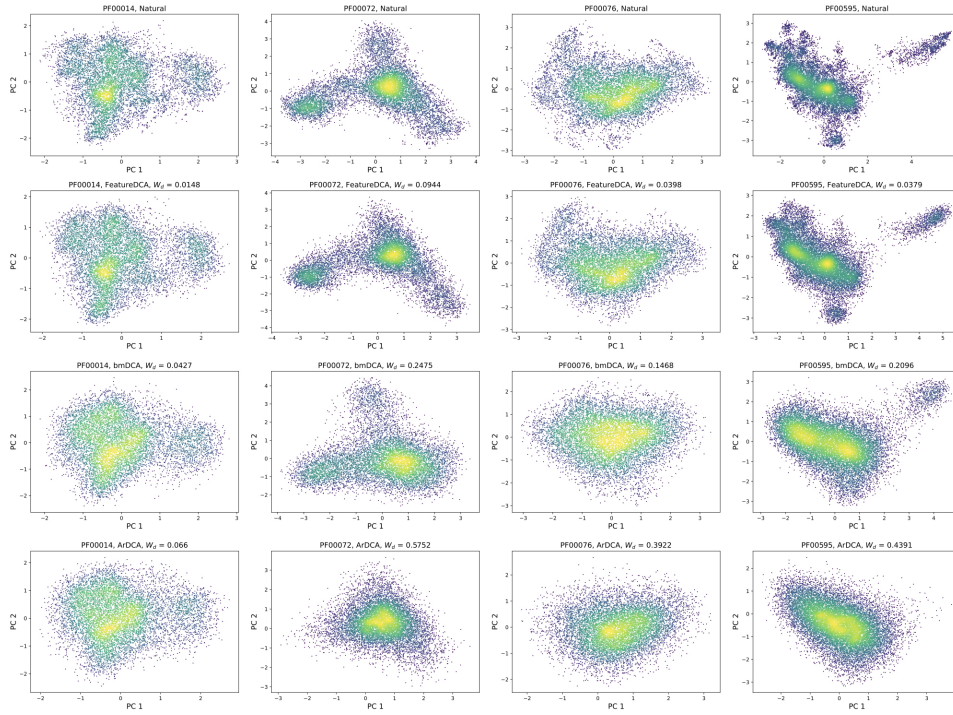


Fig. C.5 Projection onto the first two principal components of natural and generated MSAs for different protein families. Each column corresponds to a distinct family (PF00014, PF00072, PF00076, PF00595). The first row shows the PCA projection of natural sequences; subsequent rows display projections of sequences generated by FeatureDCA, bmDCA, and arDCA, respectively. For each case, the Wasserstein distance W_d quantifies the discrepancy between the natural and generated distributions along the first two principal components. Source [43]

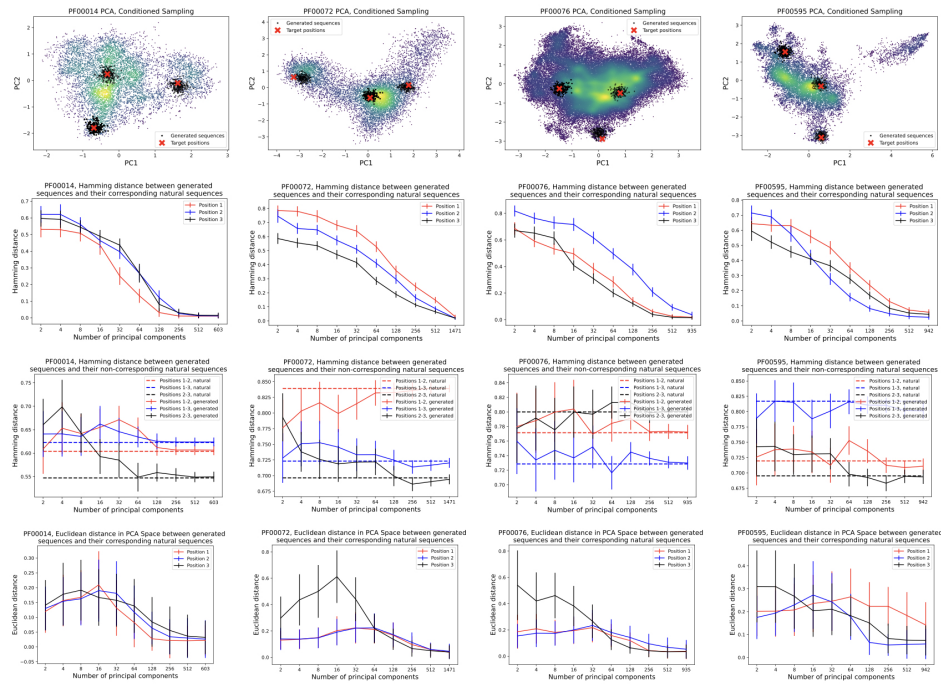


Fig. C.6 Conditional generativity across protein families. Each column corresponds to a protein family (PF00014, PF00072, PF00076, PF00595). **First row:** red crosses mark the conditioning positions chosen in PCA space; black dots show the generated sequences sampled around those targets. **Second row:** Hamming distance between generated sequences and their conditioning wild types. **Third row:** Hamming distance between generated sequences and non-corresponding wild types. **Fourth row:** Euclidean distance in the first two principal components between the conditioning positions and generated sequences. Source [43]

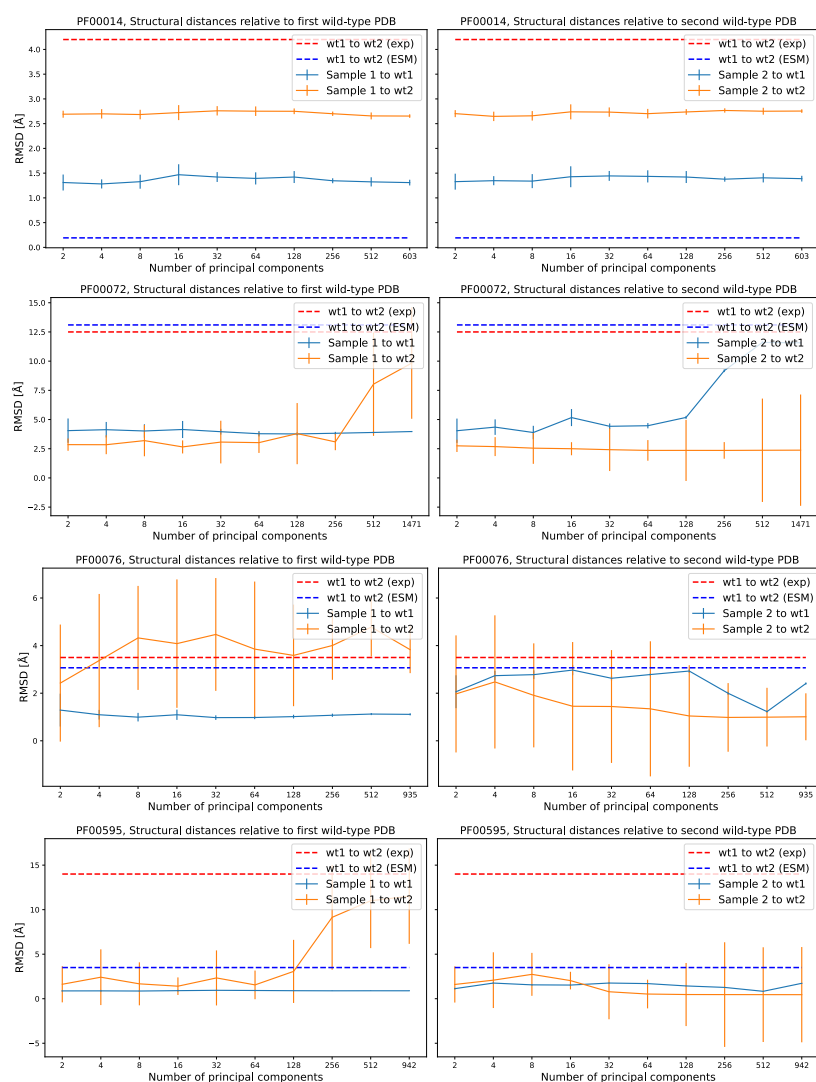


Fig. C.7 Structural similarity between generated and wild-type sequences across different protein families. Each row corresponds to a protein family (PF00014, PF00072, PF00076, PF00595). Each panel shows the RMSD (Å) between ESM-predicted structures of generated sequences and their corresponding wild-type references, plotted against the number of principal components used for training and generation. **Left:** RMSD relative to the first wild-type structure (wt1). **Right:** RMSD relative to the second wild-type (wt2). Dashed lines indicate the RMSD between the two wild types using experimental (red) and predicted (blue) structures; solid curves show average RMSDs of generated samples (blue: wt1, orange: wt2) with standard deviation as error bars. Source [43]