POLITECNICO DI TORINO
Repository ISTITUZIONALE

Toward manifest relationality in transformers via symmetry reduction

Original
Toward manifest relationality in transformers via symmetry reduction / François, J., Ravera, L.. - ELETTRONICO. - 1:(2026), pp. 1-20. [10.1103/v57m-nhhq]

Availability:
This version is available at: 11583/3012397 since: 2026-06-29T17:47:50Z

Publisher:
American Physical Society - APS

Published
DOI:10.1103/v57m-nhhq

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

15 October 2026

Toward manifest relationality in transformers via symmetry reduction

J. François [*]
Department of Philosphy, University of Graz (Uni Graz), Heinrichstraße 26/5, 8010 Graz, Austria; Department of Mathematics, Masaryk University (MUNI), Kotlářská 267/2, Veveří, Brno, Czech Republic; and Department of Physics, Mons University (UMONS), 20 Place du Parc, 7000 Mons, Belgium

L. Ravera [†]
Politecnico di Torino (PoliTo), Corso Duca degli Abruzzi 24, 10129 Torino, Italy; Istituto Nazionale di Fisica Nucleare (INFN), Section of Torino, Via P. Giuria 1, 10125 Torino, Italy; and Grupo de Investigación en Física Teórica, Universidad Católica De La Santísima Concepción, Alonso de Ribera 2850, Concepción, Chile

(Received 24 February 2026; accepted 1 May 2026; published 2 June 2026)

Transformer models contain substantial internal redundancy arising from coordinate-dependent representations and continuous symmetries, in model space and in head space, respectively. While recent approaches address this by explicitly breaking symmetry, we propose a complementary framework based on symmetry reduction. We reformulate representations, attention mechanisms, and optimization dynamics in terms of invariant relational quantities, eliminating redundant degrees of freedom by construction. This perspective yields architectures that operate directly on relational structures, providing a principled geometric framework for reducing parameter redundancy and analyzing optimization.

DOI: 10.1103/v57m-nhhq

I. INTRODUCTION

Transformer architectures have become the dominant foundation of modern Machine Learning (ML) systems for language, vision, and multimodal reasoning [1–4]. Since their introduction in Ref. [5], transformers have replaced recurrent and convolutional models in most large-scale sequence modeling tasks.

At a high level, a transformer processes sequences by representing each token as a vector in a shared hidden-feature space, $x_i \in \mathbb{R}^d$, where d is the model (hidden) dimension (all layers act on and update these vectors within the same ambient space $\mathbb{R}^d$), and repeatedly refining these vectors through *attention mechanisms* and *feedforward layers*. Attention allows each token representation to update itself by referencing all other tokens (unless causal masking is applied) in the sequence, enabling flexible modeling of long-range dependencies and contextual relationships. Stacking such layers yields deep models capable of learning complex linguistic and semantic structures.

[*]Contact author: jordan.francois@uni-graz.at
[†]Contact author: lucrezia.ravera@polito.it

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Despite their empirical success, transformers remain poorly understood from a theoretical standpoint [6–8]. Recent research has begun to analyze symmetry and equivariance in deep learning [9,10], as well as optimization dynamics and representational geometry [11,12]. Here, let us remark that, in transformers, equivariance means that if you apply a transformation g to the input (or to an internal representation) and there is a corresponding transformation $\rho(g)$ on the output (or next-layer representation), then the layer F (here, F is simply the learned function implemented by the layer) commutes with that action: $F(g \cdot X) = \rho(g) \cdot F(X)$. It is an architectural symmetry constraint: The computation is designed (or happens) to respect a group action (e.g., token permutations in set/graph transformers, translations/rotations in vision, or internal head-space basis changes in attention submodules). Invariance is the special case $\rho(g) = \mathrm{id}$; i.e., the output does not change.

In particular, our fundamental focus in this work is the following fact: Transformers feature large continuous symmetries in their linear attention submodules, so that many parameters and representation configurations correspond to identical model behavior [13–15]: This large redundancy may result in nonoptimal use of computational resources and/or hinder convergence.

Notably, recent works have identified continuous reparametrization symmetries in attention that go beyond the classical permutation symmetries of Multilayer Perceptrons (MLPs). More specifically, rotation-type symmetries in self-attention have been analyzed and exploited for model fusion [14], and quotient-geometric perspectives have been used to define sharpness and geometry on symmetry-reduced manifolds, including the $GL(d_h, \mathbb{R})$ reparametrization symmetry of the value-output sector in attention heads [16] (d_h being the dimension of the head-space vector space). On the dynamical side, symmetry can induce conserved quantities in idealized learning dynamics [17], and very recent work shows how such conserved quantities can obstruct certain efficient training schemes for transformers, motivating explicit symmetry-breaking interventions [15]. While effective, such breaking introduces preferred directions into internal representation spaces and may obscure deeper relational structure in learned representations [18–22]. Our contribution is to outline a complementary symmetry-reduction program: Rather than breaking the symmetry, we formulate states, attention, and optimization directly in terms of invariant relational variables.

In this, we take inspiration from developments in the modern theoretical physics of gauge field theory that aims, rather than breaking symmetry (e.g., by selecting arbitrary coordinate frames), to eliminate redundant degrees of freedom (d.o.f.) by working directly with *manifestly invariant* variables, e.g., constraint Hamiltonian dynamics [23], symplectic reduction [24], or covariant phase-space approaches [25]. One tool, developed in the past decade, systematically implementing such an idea is the *Dressing Field Method* (DFM) [26–37], which allows to extract the *relational* [38–41] invariant content of physical theories.

We note here that gauge ideas already appear, e.g., in geometric Deep Learning when modeling data on manifolds or non-Euclidean domains (graphs, spheres, tori, etc.)—see, e.g., Refs. [42–48]. In this context, "gauge" often refers to local coordinate freedom/choice of basis in tangent spaces and gauge-equivariant convolutions that respect this freedom (for instance, no preferred frame). These works use the geometric language of connections on (principal) fiber bundles.

In this work, we propose to adapt the DFM philosophy to *transformer architectures*: Instead of modifying networks by inserting preferred coordinate directions, we consider how transformer computations can be reformulated in terms of *relational*, manifestly invariant structures, removing internal coordinate redundancies while preserving model expressivity. We focus on components where internal redundancies appear and propose invariant reformulations:

(1) Dressing internal representation frames by replacing coordinate-dependent vectors with relational invariants, hence reformulating attention mechanisms in invariant relational form.

(2) Considering optimization dynamics in reduced, symmetry-free parameter spaces.

Our immediate goal is not yet to replace existing architectures, but to establish a conceptual and mathematical framework in which learning proceeds directly on meaningful relational d.o.f. rather than on arbitrary coordinate representations. Such a formulation may reduce optimization degeneracies, clarify internal mechanisms, and open directions for invariant and interpretable architectures. In this context, let us observe that our proposal is orthogonal to the encoder-only

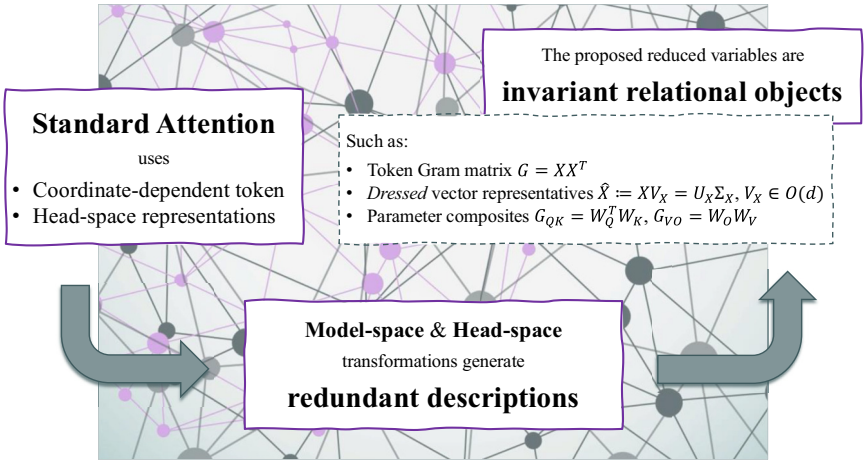

FIG. 1. Schematic summary of the main content of the work.

versus decoder-only architectural distinction: The symmetry-reduction viewpoint concerns internal representational and parameter redundancies and, in principle, may be studied in encoder, decoder, or encoder-decoder attention blocks alike. Accordingly, we do not claim superiority over masked autoregressive decoder-only architectures in the present work.

For clarity, we distinguish three distinct (but compatible) reductions: (1) *state/representation reduction*, which replaces coordinate-dependent hidden states $X \in \mathbb{R}^{n\times d}$ (X is the hidden-state matrix of one transformer layer, n is the sequence length—number of tokens, each row is one token—and d is the model/hidden dimension, or size of each token vector; so, row i of X is $x_i \in \mathbb{R}^d$) by invariant relational objects [e.g., $XX^\top$ or $XAX^\top$, where $XX^\top \in \mathbb{R}^{n\times n}$ is the *Gram matrix of tokens*—entry (i, j) is $x_i^\top x_j$—which encodes pairwise relations between tokens and is invariant under any global rotation of the hidden space, while $XAX^\top$, with $A \in \mathbb{R}^{d\times d}$, denotes a learned matrix—entry (i, j) is $x_i^\top A x_j$—or learned relational kernel, a more flexible invariant similarity than plain dot products]; (2) *parameter reduction*, which replaces redundant parametrizations by invariant composites (e.g., $W_Q^\top W_K$, with W_Q the *query* projection matrix and W_K the *key* projection matrix, both $\in \mathbb{R}^{d_h\times d}$—they are both learned linear maps; they take a token vector $x_i \in \mathbb{R}^d$ and produce $q_i = W_Q x_i$, $k_i = W_K x_i$, where q_i, $k_i \in \mathbb{R}^{d_h}$ live in the head space; then, attention scores are $q_i^\top k_j$, which is just the dot product between query and key) that determine the forward computation; and (3) *dynamical/optimization reduction*, which modifies training so that updates have no component along symmetry orbits (quotient or projected dynamics).

Sections III and IV focus on (1), while Sec. V is devoted to (2) and (3). Figure 1 provides a schematic overview of our symmetry-reduction perspective. A complementary summary of the redundancies and the corresponding invariant variables is given in Table I.

Throughout this paper, we distinguish two independent sources of redundancy in transformer architectures and address them separately:

TABLE I. Redundancies and corresponding invariants.

Redundancy type	Invariant variable
Model space frame	$G = XX^\top, XAX^\top, \hat{X} = Xu[X]^{-1}$
QK head-space symmetry	$G_{QK} = W_Q^\top W_K$
VO head-space symmetry	$G_{VO} = W_O W_V$

(1) *Model space (representation-frame) redundancy*, corresponding to arbitrary orthogonal changes of basis $X \mapsto XU^\top$, $U \in O(d)$, in the model feature space $\mathbb{R}^d$. This motivates a reformulation of hidden states and attention weights in terms of invariant relational quantities (Secs. III and IV).

Part I. Relational Representation and Model Space Symmetry Reduction in $\mathbb{R}^d$ of the article is devoted to this.

(2) *Parameter space reparametrization redundancy* inside attention heads, corresponding to exact internal symmetries $O(d_h)$ (query-key sector) and $GL(d_h, \mathbb{R})$ (value-output sector). These symmetries act on parameters while leaving the computed function unchanged and are treated via symmetry-reduced optimization and invariant composites (Sec. V).

Part II. Attention-Head reparametrization Symmetry and Optimization Dynamics in $\mathbb{R}^{d_h}$ of the paper concerns this.

The first reduction concerns *representations*; the second concerns *parameters*. They are logically independent but complementary.

The remainder of this paper is structured as follows. In Sec. II, we briefly review the DFM in its field-theoretic setting, emphasizing its role as conceptual motivation for symmetry reduction via relational variables. Section III discusses internal representation frames in transformer models. In Sec. IV, we reformulate token representations in model space in relational terms via "dressing." Section V addresses optimization dynamics on reduced, symmetry-free parameter spaces, hence in head space. There, we also provide a concrete differentiable realization and initialization sketch for a representative case, that is, the value-output sector of a single attention head. Section VI concludes the paper with directions for future work, including a concrete evaluation road map. In the Appendix, we provide an illustrative algorithmic sketch of symmetry-reduced optimization for a representative attention submodule.

Let us remark that this paper develops a conceptual and mathematical framework. We (1) identify precise reparametrization symmetries for idealized transformer blocks, (2) show how invariant composites encode the functionally relevant d.o.f. in specific submodules (e.g., query-key—Q/K—bilinear forms), and (3) suggest symmetry-reduced architectural and optimization realizations. We do not claim that every practical transformer with LayerNorm (*layer normalization*, which normalizes each token vector independently across its feature dimension) and standard implementation details enjoys the same full symmetry group; rather, we treat such details as reducing the symmetry and motivating approximate or partial reductions. Empirical evaluation and engineering trade-offs are left for future work.

a. Scope of the present work: The purpose of this paper is conceptual and mathematical. We develop a symmetry-reduction framework for transformer representations and optimization, identify the relevant invariant composites in idealized attention submodules, and formulate corresponding relational variables. We also provide concrete differentiable realizations for representative sectors (see Sec. V D), illustrating how invariant variables, gradients, and initialization may be handled in practice.

We do not, however, present a benchmarked implementation or claim empirical speedups, accuracy gains, or reduced training cost on standard datasets. Such questions require a dedicated engineering and experimental study, which lies beyond the scope of the present article. To make the empirical implications of the framework precise, we provide in Sec. VI A a concrete evaluation road map for future work.

II. THE DRESSING FIELD METHOD IN FIELD THEORY: CONCEPTUAL MOTIVATION

The purpose of this section is conceptual rather than technical. We do not claim that transformer models possess gauge symmetries in the field-theoretic sense, nor that the Dressing Field Method can be directly applied to ML architectures. Instead, we briefly review the DFM in its native theoretical physics context as a mature framework whose central insight—the elimination

of redundant d.o.f. through invariant, relational variables—serves as methodological *inspiration* for the symmetry-reduction program suggested here.

In Gauge Field Theory, the DFM provides a systematic procedure for replacing gauge-dependent fields by relational, gauge-invariant quantities extracted from the original field content. Our goal is to abstract this philosophy and adapt it to transformer architectures, where internal representations exhibit analogous redundancies arising from coordinate-dependent parametrizations rather than gauge symmetries.

We therefore present a concise overview of the DFM in the case of internal gauge symmetries, in order to motivate the relational reformulations introduced in subsequent sections.

In its natural theoretical physics framework, namely, general-relativistic Gauge Field Theory, the DFM [26–37] is used to construct gauge-invariant, relational variables [41] from the field space $\Phi = \{\phi\}$ of a gauge theory, ϕ being a collection of fields on the "spacetime" manifold M. We briefly recall the dressing procedure for internal gauge symmetry group $\mathcal{H}$, whose elements are smooth maps $\gamma : M \to H$ and H is a (finite-dimensional) Lie group.

A. Invariant dressed fields

Consider a gauge field theory with field content $\phi = \{A, \varphi\}$, where A denotes the gauge potential [e.g., the electromagnetic potential, when $H = U(1)$] and φ collectively denotes matter fields. These transform under the gauge group $\mathcal{H}$ as

$$A^\gamma := \gamma^{-1} A \gamma + \gamma^{-1} d\gamma, \quad \varphi^\gamma := \gamma^{-1} \varphi. \tag{1}$$

A dressing field is defined as a smooth map

$$u : M \to H, \quad \text{such that} \quad u^\gamma = \gamma^{-1} u, \tag{2}$$

with the crucial property that u is extracted from the original field content, i.e., $u = u[\phi]$. Consequently, the dressing field is an equivariant functional of the fields,

$$u^\gamma := u[\phi^\gamma] = \gamma^{-1} u[\phi]. \tag{3}$$

Given such a dressing field, one defines the dressed fields

$$A^u := u^{-1} A u + u^{-1} du, \quad \varphi^u := u^{-1} \varphi. \tag{4}$$

By construction, these are strictly $\mathcal{H}$ invariant.

When the dressing field is field dependent, the resulting dressed variables admit a natural relational interpretation: The dressed fields $\{A^{u[A,\varphi]}, \varphi^{u[A,\varphi]}\}$ encode gauge-invariant relations among the original d.o.f. in $\phi = \{A, \varphi\}$. In this sense, the DFM replaces gauge-dependent variables with relational physical quantities.

The dynamics of a gauge field theory is determined by a Lagrangian $L(\phi)$, typically required to be quasiinvariant under $\mathcal{H}$, i.e.,

$$L(\phi^\gamma) = L(\phi) + db(\gamma; \phi). \tag{5}$$

This ensures the covariance of the field equation, obtained from $L(\phi)$ via the variational principle, under $\mathcal{H}$. Given a dressing field u, one defines the dressed Lagrangian

$$L(\phi^u) := L(\phi) + db(u; \phi), \tag{6}$$

which is strictly $\mathcal{H}$ invariant. The dressed field equations possess the same functional form as the original ones, but now govern gauge-invariant relational variables with a well-posed deterministic evolution.

B. Residual transformations

If $\mathcal{H}$ is only partially reduced, dressed fields may still transform under residual symmetries. Additionally, possible ambiguities in the choice of dressing field are encoded in *transformations of the second kind*, often interpretable as physical reference-frame covariance [32,35].

We emphasize that this brief review is presented solely to illustrate the general mechanism of symmetry reduction through invariant relational variables. In the remainder of this work, we apply this idea abstractly to transformer architectures, replacing gauge symmetry by internal representational redundancy and dressed fields/variables by relational reformulation of neural representations.

PART I. RELATIONAL REPRESENTATION AND MODEL SPACE SYMMETRY REDUCTION IN $\mathbb{R}^d$

III. INTERNAL REPRESENTATION FRAMES IN MACHINE LEARNING

A central observation motivating our proposed approach to ML is that internal representations in transformer models are not uniquely defined: They depend on arbitrary coordinate choices in hidden-state space. While such choices do not affect model outputs, they introduce redundant internal d.o.f. that influence optimization dynamics and obscure the relational structure encoded in representations. In this section, we analyze this coordinate redundancy and propose a reformulation in which computations depend only on invariant, relational quantities rather than arbitrary coordinate frames.

A. Coordinate redundancy in transformer representations

In transformer architectures, hidden states at a given layer are represented as vectors

$$x_i \in \mathbb{R}^d, \quad i = 1, \ldots, n, \tag{7}$$

where n is the sequence length and d the hidden dimension. Attention and feedforward layers act on these vectors through learned linear maps.

For simplicity of exposition, we shall suppress positional encoding mechanisms in the main formulas. In practical transformers, positional information may be added either additively or through structured modifications of the query-key interaction, as in rotary position embeddings. Such mechanisms generally constrain/modify the symmetry structure discussed here.

Each token representation x_i is linearly projected into three vectors called the query, key, and value representations:

$$q_i = W_Q x_i, \quad k_i = W_K x_i, \quad v_i = W_V x_i, \tag{8}$$

where the matrices

$$W_Q, W_K, W_V \in \mathbb{R}^{d_h \times d} \tag{9}$$

are learned parameters and d_h is the dimension of each attention head. Attention weights are computed from query-key similarities,

$$\begin{aligned} s_{ij} &= q_i^\top k_j = (W_Q x_i)^\top (W_K x_j), \\ \mathrm{Attn}(i, j) = \alpha_{ij} &:= \mathrm{softmax}_j\left(\frac{s_{ij}}{\sqrt{d_h}}\right), \\ y_i &= \sum_j \alpha_{ij} v_j, \end{aligned} \tag{10}$$

and are then used to aggregate value vectors across tokens, producing context-dependent updates of token representations. We recall that the "softmax" function applied to a vector of raw scores (logits) $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$ is defined as $\mathrm{softmax}(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^n \exp(z_j)}$, $i = 1, \ldots, n$. It maps the

scores to a probability distribution (non-negative entries summing to 1), with larger values receiving exponentially higher weight. Finally, we have $W_O \in \mathbb{R}^{d\times d_h}$, which projects head-space outputs in $\mathbb{R}^{d_h}$ back to the model space $\mathbb{R}^d$. Notice that the softmax does not break the shared $O(d_h)$ query-key symmetry because it depends only on the invariant logits $q_i^\top k_j$, but nonlinear and normalization layers outside this idealized submodule generally reduce the full symmetry.

Although the mechanism in Eq. (10) enables flexible contextual interactions, it remains fundamentally coordinate based: Queries, keys, and values are vectors in an internal latent space whose basis is arbitrary.

This means that transformer blocks contain *reparametrization redundancies* in intermediate subspaces.

First, on the one hand, we have a transformation acting on tokens in the *model space*,

$$x_i \mapsto U x_i, \qquad \text{with } U \in O(d), \tag{11}$$

which is a change of basis in *model space*. To preserve the attention computation under this change of basis, the weight matrices must transform as

$$\begin{aligned} W_Q &\mapsto W_Q U^{-1}, \\ W_K &\mapsto W_K U^{-1}, \\ W_V &\mapsto W_V U^{-1}, \\ W_O &\mapsto U W_O. \end{aligned} \tag{12}$$

This is *representation covariance*, not a parameter redundancy.

On the other hand, we have continuous symmetries arising in the *linear attention submodules* as invertible changes of basis in each *head space*. Concretely, for a single head with $W_Q, W_K, W_V \in \mathbb{R}^{d_h\times d}$ and $W_O \in \mathbb{R}^{d\times d_h}$, the forward computation is invariant under

$$(W_Q, W_K) \mapsto (S W_Q,\ S W_K), \qquad S \in O(d_h), \tag{13}$$

$$(W_V, W_O) \mapsto (S^{-1} W_V,\ W_O S), \qquad S \in GL(d_h, \mathbb{R}). \tag{14}$$

We restrict to $O(d_h)$ since the score uses the standard Euclidean inner product in head space; a general $GL(d_h, \mathbb{R})$ transformation would not preserve, e.g., $q_i^\top k_j$ without a compensating change of metric. Note that (1) query-key scores depend only on head-space inner products and are preserved by the shared $O(d_h)$ rotation in Eq. (13), and (2) the value-output contribution depends only on the composite $W_O W_V$ and is therefore invariant under the refactorization symmetry (14).

Consequently, attention operates in a representation space containing redundant d.o.f. So, multiple internal parameter configurations correspond to identical model functions. Learning dynamics may therefore evolve along directions that do not affect outputs, producing optimization degeneracies analogous to gauge redundancies in physical systems. This motivates our proposal: reformulating attention directly in terms of relational invariants.

Notice, however, that a full transformer block with LayerNorm, biases, and elementwise MLP nonlinearities is not invariant under arbitrary changes of basis in the model dimension d; our relational scoring isolates an $O(d)$-invariant subcomputation inside this otherwise symmetry-breaking architecture (see Sec. III A 1).

1. What symmetry actually holds (architecture dependent)?

Before proceeding with our approach, to avoid overclaiming, let us fix here an explicit class of blocks and state the precise reparametrization symmetry.

a. What is symmetric?. In standard transformers, most nonlinear components (MLP with elementwise activation, LayerNorm, biases) are defined with respect to a chosen coordinate system in $\mathbb{R}^d$ and therefore are *not* equivariant under an arbitrary $GL(d, \mathbb{R})$ change of basis in the hidden-feature dimension. Exact continuous symmetries instead appear in specific *linear submodules*

(notably, *attention heads*), where reparametrizations of intermediate head spaces leave the computed function unchanged (see Sec. V). Accordingly, in this paper, *internal symmetry* refers to (1) exact symmetries of such linear submodules and (2) approximate symmetries of more complete blocks when nonlinearities act as symmetry-breaking perturbations.

b. Effect of LayerNorm and biases. In practical transformers, LayerNorm and bias terms restrict this symmetry. For example, standard LayerNorm is not equivariant under arbitrary $GL(d, \mathbb{R})$ basis changes. Hence, throughout this paper, any claim of continuous internal symmetry should be interpreted as (1) exact for the minimal block above, and (2) approximate or reduced to a smaller group for common architectural choices (LayerNorm, biases, parameter tying), which we treat as symmetry-breaking perturbations of the idealized symmetry.

This point is especially important in practice, since mainstream architectures in language and vision (e.g., BERT/GPT-type models, ViT-type models) systematically include normalization layers and nonlinear MLP blocks. Accordingly, the exact continuous symmetries discussed here should be understood as properties of idealized linear attention submodules embedded in larger architectures where these symmetries are partially or approximately broken. This is why we view the present work as a framework for identifying the symmetry-reduced core structure, rather than as a claim about exact full-network symmetry in production models.

IV. DRESSING AND RELATIONAL INVARIANTS IN MODEL SPACE

We start by providing our idea for a relational scoring formulation in which attention *weights* are computed from the Gram matrix

$$G := XX^\top \in \mathbb{R}^{n\times n}, \quad n = \text{sequence length}, \tag{15}$$

which is strictly invariant under $X \mapsto XU^\top$, $U \in O(d)$. Token states are represented relationally by the Gram data, e.g., by the ith row $(G_{ij})_{j=1}^n$ that encodes the relations of token i to all other tokens in the current sequence, rather than by coordinate-dependent vectors $x_i \in \mathbb{R}^d$. When convenient, we still use X as an $O(d)$-equivariant representative to transport vector-valued features, while the invariant content used for scoring is carried by G.

We present this $O(d)$-equivariant Gram-based attention as an idealized, symmetry-manifest alternative; incorporating generic learned feature maps (e.g., W_V, W_O) would generally break $O(d)$ unless explicitly constrained. Accordingly, we view the following construction as a conceptual prototype of relational attention. We may therefore think of an attention mechanism in which attention weights depend only on $O(d)$-invariant quantities between token representations [hence, $O(d)$-frame-invariant attention weights]. Let $X \in \mathbb{R}^{n\times d}$ denote the matrix of token states (row i is $x_i^\top$), and let the model feature-space transform as Eq. (11). Under this action, the Gram matrix $G := XX^\top$, $G_{ij} = x_i^\top x_j$, is strictly invariant.

We therefore define attention scores directly from Gram invariants as

$$s_{ij} = f(R_{ij}) = f(G_{ij}, G_{ii}, G_{jj}), \tag{16}$$

where $f : \mathbb{R}^3 \to \mathbb{R}$ is a learned scalar function (for instance, a small multilayer perceptron applied pointwise) and $G_{ii} = \|x_i\|^2$ allows the scoring function to access token norms. We remark that, when R_{ij} is taken to be an inner product (or a learned bilinear form) and f is scalar, the resulting attention weights are a form of *kernelized attention*. Our contribution here is not to claim novelty of pairwise-kernel attention *per se*, but to place such constructions in a symmetry-reduction program: (1) choose relational invariants as primary variables, and (2) parametrize and optimize only the d.o.f. that affect these invariants. We stress that this construction is intended as an idealized, illustrative prototype rather than a drop-in replacement for standard dot-product attention, meant to make the symmetry-reduced structure explicit.

The attention weights are then given by

$$\mathrm{Attn}(i, j) = \mathrm{softmax}_j\Big(\frac{s_{ij}}{\tau}\Big), \tag{17}$$

where $\tau > 0$ is a temperature parameter (analogous to the conventional score normalization in dot-product attention). By analogy with dot-product attention, one may set $\tau = \sqrt{d_h}$, but here τ is simply a temperature hyperparameter for the scalar logits s_{ij}. We restrict here to $\tau > 0$, for which the softmax is smooth. The singular limit $\tau \to 0^+$ corresponds formally to a hard argmax selection and would require a separate nonsmooth treatment, beyond the scope of the present discussion.

Since the scores depend only on Gram invariants, the attention weights are strictly $O(d)$ invariant. To propagate vector-valued information while preserving symmetry, we take values to be the token states themselves and define the output as

$$y_i = \sum_j \mathrm{Attn}(i, j)\, x_j, \tag{18}$$

or in matrix form $Y = AX$, where $A_{ij} = \mathrm{Attn}(i, j)$. This update is $O(d)$ equivariant: If $X \mapsto XU^\top$, then $Y \mapsto YU^\top$. If one wishes a *fully* invariant state update, one may propagate G itself:

$$G^+ := YY^\top = (AX)(AX)^\top = AGA^\top, \tag{19}$$

which is strictly $O(d)$ invariant as A depends only on G.

In this formulation, internal coordinate frames are eliminated by construction: Attention operates purely on relational invariants and vector features are transported equivariantly. This realizes a genuine dressed attention mechanism in which semantic interaction depends only on relations between tokens rather than on arbitrary feature-space coordinates.

A. Dressing of vector representatives

Besides fully relational state propagation, one may realize symmetry reduction by introducing a *dressing field* that selects a canonical representative of each $O(d)$ orbit while retaining vector-valued carriers. Let $X \in \mathbb{R}^{n\times d}$ denote the token matrix, transforming as

$$X \mapsto XU^\top, \quad U \in O(d). \tag{20}$$

A "dressing field" is a data-dependent map

$$u[X] \in O(d) \tag{21}$$

satisfying the equivariance property

$$u[XU^\top] = u[X]\, U^\top. \tag{22}$$

One then defines the dressed representative

$$\hat{X} := X\, u[X]^{-1}. \tag{23}$$

By construction,

$$X \mapsto XU^\top \;\Rightarrow\; \hat{X} \mapsto \hat{X}, \tag{24}$$

so $\hat{X}$ is strictly $O(d)$ invariant.

A concrete choice is obtained from the singular value decomposition (SVD, which factorizes X into orthogonal matrices U_X, V_X, and a diagonal matrix Σ_X, enabling a canonical extraction of the right singular vectors V_X)

$$X = U_X\, \Sigma_X\, V_X^\top, \quad V_X \in O(d), \tag{25}$$

and defining

$$u[X] := V_X^\top. \tag{26}$$

Under $X \mapsto XU^\top$, one has (for a consistent choice of SVD) $V_{XU^\top} = U\, V_X$; hence,

$$u[XU^\top] = V_{XU^\top}^\top = V_X^\top U^\top = u[X]\, U^\top, \tag{27}$$

as required. The dressed representative becomes

$$\hat{X} := XV_X = U_X\,\Sigma_X. \tag{28}$$

All subsequent computations may then be performed using $\hat{X}$, which is invariant under model space frame rotations.

a. Residual ambiguity. If Σ_X has degeneracies or X is rank deficient, V_X is not unique, leading to a residual $O(m)$ freedom within each m-dimensional degenerate singular subspace, together with discrete sign and permutation ambiguities. This corresponds to the standard *transformations of the second kind*, familiar from DFM constructions.

Note that this realization implements symmetry reduction without eliminating vector carriers: Vectors are retained for computational convenience, but internal frame redundancy is removed by dressing rather than broken.

b. Well-definedness of the dressing map. The definition $u[X] := V_X^\top$ presupposes a deterministic convention for choosing the SVD. Indeed, the right singular factor $V_X \in O(d)$ is not uniquely defined: Even when Σ_X has *distinct* nonzero singular values, each singular vector is only defined up to an overall sign, and when Σ_X has degeneracies [or $\mathrm{rank}(X) < d$] there is a continuous freedom $V_X \mapsto V_X R$, with $R \in O(m)$ acting in each degenerate singular subspace. Accordingly, the equivariance identity $u[XU^\top] = u[X]\,U^\top$ holds on the open set where a chosen convention fixes these ambiguities (e.g., by fixing the sign of each singular vector via a deterministic rule), and otherwise holds up to the corresponding residual transformation. In practice, one may (1) restrict the discussion to matrices X with simple singular spectrum, where the map can be made locally smooth by such conventions, or (2) treat the remaining freedom as a residual symmetry (of the second kind) and accept that $\hat{X}$ is defined only up to the corresponding residual action.

Our relational formulation eliminates dependence on model space coordinate frames $[O(d)]$ at the level of attention weights. Even after eliminating model space frame dependence, standard attention heads still possess exact internal reparametrization symmetries acting on intermediate head spaces. These do not affect representations directly but generate flat directions in parameter space. We now turn to this second, independent reduction.

In the next section, we examine how optimization dynamics themselves can be reformulated in reduced relational parameter spaces. In fact, head-space reparametrization symmetries $[O(d_h), GL(d_h)]$ act in parameter space and can be removed by quotient-based optimization. Optimization dynamics therefore evolve in reduced parameter spaces, eliminating directions that do not affect model behavior. Note that this symmetry reduction parallels gauge reduction in physical systems (this is an analogy, not a literal gauge symmetry): Rather than introducing preferred directions, redundant d.o.f. are eliminated by construction. Consequently, learning trajectories are constrained to meaningful relational manifolds, potentially improving optimization efficiency and stability.

PART II. ATTENTION-HEAD REPARAMETRIZATION SYMMETRY AND OPTIMIZATION DYNAMICS IN $\mathbb{R}^{d_h}$

V. OPTIMIZATION DYNAMICS ON REDUCED, SYMMETRY-FREE PARAMETER SPACES

Standard transformer parametrizations contain large internal redundancies: Many distinct parameter settings realize the same input-output function due to continuous symmetries (e.g., internal rotations of representation subspaces) and other reparametrization freedoms in head space. From an *optimization* perspective, such redundancies give rise to flat directions and may induce dynamics that expend update budget moving along directions that do not change the predictive behavior of the model.

In fact, when training is viewed dynamically, internal symmetries can be particularly consequential. Symmetry directions can create (approximately) conserved quantities or long-lived modes that constrain exploration and slow descent in functionally relevant directions. This phenomenon has been highlighted recently in Hamiltonian analyses of learning dynamics in attention models,

where continuous symmetries yield conserved currents in idealized dynamics and corresponding optimization pathologies in practice [15].

The objective of this section is to formalize a symmetry-reduction viewpoint for optimization: Rather than breaking symmetries by introducing preferred directions, we aim to *quotient out* redundant d.o.f. and study learning directly on a reduced, symmetry-free space of functionally distinct models.

A. Symmetry groups and equivalence classes in parameter space

Let $\theta \in \Theta$ denote the full parameter vector of a transformer model (all trainable weights and biases, i.e., in a transformer: embedding tables, all projection matrices W_Q, W_K, W_V, W_O, MLP weights, LayerNorm parameters, etc.; Θ is the parameter space) and let

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \ell(f_\theta(x), y) \tag{29}$$

be the population loss, that is, the expected loss over the true data distribution [which is the ideal objective one would minimize knowing $\mathcal{D}$ exactly, with x the input random variable, y the target/label random variable associated with x, (x, y) a data point, and $\mathcal{D}$ the data distribution over pairs (x, y)—writing $(x, y) \sim \mathcal{D}$ means "sample a random training example from the underlying population distribution"; $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\cdot]$ is the expectation with respect to that distribution and ℓ denotes the per-example loss function or pointwise loss – it compares the model's prediction to the true target y].

Now, suppose there exists a (Lie) group G acting on parameters,

$$\theta \mapsto g \cdot \theta, \quad g \in G, \tag{30}$$

such that the model function is unchanged:

$$f_{g\cdot\theta}(x) = f_\theta(x) \quad \forall x,\ \forall g \in G. \tag{31}$$

Then, $\mathcal{L}(\theta)$ is constant along *group orbits*:

$$\mathcal{L}(g \cdot \theta) = \mathcal{L}(\theta). \tag{32}$$

Hence, physically (or functionally) distinct models correspond not to points in Θ but to equivalence classes

$$[\theta] \ := \ \{g \cdot \theta \,:\, g \in G\}. \tag{33}$$

The *reduced parameter space* is the quotient

$$\Theta_{\text{red}} := \Theta/G, \tag{34}$$

whose points correspond to distinct functions. Optimization should take place on Θ_{red} rather than on Θ, because motion tangent to the orbit $[\theta]$ does not change the learned function.

B. Decomposition of updates: tangent versus transverse directions

At a given θ, the orbit direction is spanned by the infinitesimal generators of G. Let $\mathfrak{g}$ denote the Lie algebra of G and let $\xi \in \mathfrak{g}$ generate an infinitesimal action $\delta_\xi\theta$. Then, tangent directions (spanning the tangent space to a group orbit) have the form

$$\delta\theta_\parallel \in T_\theta([\theta]) = \{\delta_\xi\theta : \xi \in \mathfrak{g}\}. \tag{35}$$

We may decompose any parameter update $\Delta\theta$ (i.e., a finite optimization update, e.g., gradient step) into tangent and transverse components,

$$\Delta\theta = \Delta\theta_\parallel + \Delta\theta_\perp, \tag{36}$$

where $\Delta\theta_\perp$ moves in functionally meaningful directions (changing $[\theta]$) and $\Delta\theta_\parallel$ moves along a redundant orbit. Here, $\Delta\theta_\parallel \in \text{span}\{\delta_\xi\theta : \xi \in \mathfrak{g}\}$, i.e., $\Delta\theta_\parallel = \sum_a c_a\, \delta_{\xi_a}\theta$, for some basis $\{\xi_a\}$ of $\mathfrak{g}$ and coefficients $c_a \in \mathbb{R}$.

In ideal learning dynamics, we would enforce

$$\Delta\theta_{\parallel} = 0, \tag{37}$$

so that learning updates are purely transverse. Note that this is similar, morally, to what is done in gauge theory in physics, when *fixing a gauge*. *Quotient-based* learning dynamics would avoid this by evolving directly in Θ_{red}. This is the spirit of the DFM.

a. A DFM-inspired dressing map for optimization. In physics, in gauge theory, symmetry reduction can be achieved by constructing dressed variables via a dressing field that removes the redundant group action. Translating this logic to optimization suggests introducing a projection (surjective) map

$$\begin{aligned} \mathcal{D} : \Theta &\to \Theta_{\text{red}}, \\ \theta &\mapsto \tilde{\theta} := \mathcal{D}(\theta) = \mathcal{D}(g \cdot \theta), \end{aligned} \tag{38}$$

which assigns to each θ a representative in the reduced space, or equivalently a set of invariant coordinates. Optimization then proceeds by updating $\tilde{\theta}$ and, when needed, reconstructing a compatible θ (a representative) for forward computation. Geometrically, we have the *pushforward map* $\mathcal{D}_* : T_\theta\Theta \to \Theta_{\text{red}}$ acting as

$$\mathcal{D}_*(\Delta\theta_{\parallel}) = 0, \quad \text{so} \quad \mathcal{D}_*(\Delta\theta) = \mathcal{D}_*(\Delta\theta_{\perp}) \in T_{\tilde{\theta}}\Theta_{\text{red}}. \tag{39}$$

Thus, the optimal update is $\Delta\tilde{\theta} := \mathcal{D}_*(\Delta\theta_{\perp})$.

Observe that this differs from explicit symmetry breaking: No preferred direction is introduced; rather, the redundant d.o.f. never appear as independent optimization variables.

1. Reduced variables as invariants: learning on Θ_{red}

A practical approach to quotient optimization is to parametrize the model using *invariants* under the group G. In the attention setting, an example of such invariant combinations arises from the observation that dot-product scores depend on W_Q and W_K only through the bilinear form

$$G_{QK} := W_Q^\top W_K, \tag{40}$$

since

$$(W_Q x_i)^\top (W_K x_j) = x_i^\top (W_Q^\top W_K)\, x_j = x_i^\top G_{QK}\, x_j. \tag{41}$$

Under a *left* action by an orthogonal matrix $S \in O(d_h)$,

$$W_Q \mapsto SW_Q, \quad W_K \mapsto SW_K, \tag{42}$$

the composite

$$G_{QK} := W_Q^\top W_K \in \mathbb{R}^{d\times d} \tag{43}$$

is unchanged since $(SW_Q)^\top (SW_K) = W_Q^\top W_K$. Hence, (W_Q, W_K) are identifiable only up to a shared $O(d_h)$ rotation of the head space. In physics, this would correspond to the alternative move of *dressing*, rather than *gauge fixing* [28,49].

Note that, if $d_h < d$, then $\operatorname{rank}(G_{QK}) \leqslant d_h$, so not every $d \times d$ matrix is representable as $W_Q^\top W_K$. Thus, optimizing directly in G_{QK} either requires enforcing $\operatorname{rank}(G_{QK}) \leqslant d_h$ (a constrained problem) or else allowing arbitrary G_{QK}, which changes the model class.

a. Value-output sector. Let $W_V \in \mathbb{R}^{d_h\times d}$ and $W_O \in \mathbb{R}^{d\times d_h}$ denote the value and output projections of a single head (up to architectural conventions). The contribution of this head to the hidden update is

$$x_i \mapsto W_O W_V x_i. \tag{44}$$

Define the *composite* (which can be seen as a dressed quantity)

$$G_{VO} := W_O W_V \in \mathbb{R}^{d\times d}. \tag{45}$$

Under a right action by any invertible matrix $S \in GL(d_h, \mathbb{R})$,

$$W_V \mapsto S^{-1} W_V, \quad W_O \mapsto W_O S, \tag{46}$$

the composite G_{VO} is unchanged:

$$(W_O S)(S^{-1} W_V) = W_O W_V. \tag{47}$$

Hence, (W_O, W_V) are identifiable only up to this internal $GL(d_h, \mathbb{R})$ reparametrization.

Since G_{VO} factors through a d_h-dimensional latent space, it necessarily satisfies

$$\operatorname{rank}(G_{VO}) \leqslant d_h. \tag{48}$$

Consequently, optimizing directly over G_{VO} either requires enforcing the rank constraint or, if arbitrary $d \times d$ matrices are allowed, amounts to optimizing over a different model family.

A simple projected gradient step in the value-output sector is sketched in the Appendix.

b. Multihead symmetry and head permutations. For H attention heads, the forward computation is invariant under arbitrary permutations of the heads. Let $\sigma \in S_H$ act by permuting head indices simultaneously in $(W_Q^{(h)}, W_K^{(h)}, W_V^{(h)}, W_O^{(h)})$. Then, the summed output over heads is unchanged. Thus, in addition to the continuous reparametrizations described above, there is a discrete symmetry group S_H acting on parameters.

Functionally distinct models therefore live on the quotient space

$$\Theta_{\text{red}} = \Theta / \big(O(d_h)^H_{\text{QK}} \times GL(d_h, \mathbb{R})^H_{\text{VO}} \times S_H \big), \tag{49}$$

up to architectural details.

Note that the discrete permutation symmetry S_H differs conceptually from the continuous reparametrizations discussed above. While the groups $O(d_h)^H$ and $GL(d_h, \mathbb{R})^H$ generate infinitesimal tangent directions in parameter space and are responsible for genuine optimization degeneracies, the action of S_H is purely *discrete* and corresponds only to a relabeling of identical attention heads. S_H induces no flat directions, conserved quantities, or continuous orbit structure. As such, one may regard it as a *residual reference-frame freedom*.

c. Projected gradient flow. Let us mention that a complementary approach is to keep the original parametrization but project updates to the transverse subspace. Let $\nabla\mathcal{L}(\theta)$ denote the gradient in Θ equipped with an inner product $\langle \cdot, \cdot \rangle$. Let $P_\perp(\theta)$ be the orthogonal projector onto the transverse complement of $T_\theta([\theta])$. Here, orthogonality is defined with respect to a chosen Riemannian metric on parameter space. This choice is not canonical and affects the reduced dynamics. In practice, exact projection is expensive, so symmetry reduction is typically implemented in structured approximations (e.g., per-head reductions where tangent directions are known analytically).

Then, the projected gradient flow is

$$\dot{\theta} = -P_\perp(\theta)\, \nabla\mathcal{L}(\theta). \tag{50}$$

Discretizations of this flow yield reduced versions of stochastic gradient descent or momentum methods. This quotient-based projection viewpoint is conceptually related to projection-based gradient modification methods such as PCGrad [50], although the objective here is different: We project out symmetry-tangent directions associated with internal reparametrization redundancies, rather than resolving conflicts among gradients coming from multiple tasks.

From this perspective, standard optimizers can be interpreted as evolving on Θ with implicit (often uncontrolled) components in both tangent and transverse directions, whereas a projected optimizer explicitly removes the tangent component. Notice the difference with Eq. (39).

Note that, as in other projection-based optimization schemes, removing selected update components may improve conditioning in some regimes but may also increase per-step cost or alter

beneficial stochastic effects. Hence, whether quotient-based projection is advantageous is task and implementation dependent, and should be assessed empirically.

d. Reduced Hamiltonian dynamics and conserved quantities. When optimization is formulated dynamically—for instance, in Hamiltonian or momentum-based forms—symmetries can induce conserved quantities. Consider a generic Hamiltonian-like system on (θ, π):

$$\dot{\theta} = \frac{\partial H}{\partial \pi}, \quad \dot{\pi} = -\frac{\partial H}{\partial \theta}, \tag{51}$$

where π denotes momenta. If H is invariant under a continuous group action, then the associated Noether charges are conserved in idealized dynamics, constraining exploration in phase space.

A symmetry-reduced formulation replaces the full phase space by a reduced phase space:

$$(\Theta \times \Pi)/G, \tag{52}$$

in which the Noether charges are absent as independent d.o.f.; equivalently, the reduced system evolves only in the functionally relevant directions, eliminating conserved motions along redundant symmetry orbits.

This makes clear why symmetry reduction is a principled alternative to symmetry breaking: Rather than introducing stochastic perturbations or preferred directions to shake the system out of conserved motions, we can remove the conserved directions by construction.

C. Practical schemes for symmetry-free optimization

Let us therefore outline three practical schemes for optimizing in reduced spaces:

(1) *Dressed representatives.* Maintain a representative θ for forward computation while regularly applying a dressing map $\mathcal{D}$ that removes accumulated motion along the orbit.

(2) *Invariant reparametrization.* Replace redundant parameters by invariant composites (e.g., G_{QK}, G_{VO}) and train directly in these variables. This eliminates redundant motions but may change computational cost and requires ensuring sufficient expressivity and numerical stability.

(3) *Projection-based quotient updates.* Maintain standard parameters but project gradients (or momentum) orthogonally to symmetry orbits. This requires computing tangent directions and implementing projectors, but keeps the forward pass unchanged.

Each scheme implements the same principle: learning updates should occur only in the reduced space of functionally distinct models.

This section formalized optimization as a dynamical process on a parameter space with symmetry-induced redundancies. We proposed a symmetry-reduced viewpoint in which training takes place on a quotient space Θ/G, either by reparametrizing in invariant variables or by explicitly projecting updates to remove tangent components along symmetry orbits.

In summary, attention layers exhibit continuous internal symmetries $O(d_h)$ (query-key), $GL(d_h, \mathbb{R})$ (value-output), and a discrete head permutation symmetry S_H. The functionally meaningful d.o.f. are therefore encoded in invariant composites such as G_{QK} and G_{VO} subject to rank constraints, together with their arrangement across heads. Symmetry-reduced optimization amounts to learning directly on this quotient space rather than on redundant parametrizations.

For clarity, we now provide a minimal concrete differentiable realization and initialization strategy for a representative reduced sector.

D. Concrete differentiable realization and initialization sketch

To clarify how symmetry-reduced variables may be handled in practice, we describe a minimal differentiable realization for a representative case, namely, the value-output sector of a single attention head. This sector contributes to the hidden update through the composite operator $G_{VO} := W_O W_V \in \mathbb{R}^{d\times d}$, which is invariant under the internal reparametrization $(W_V, W_O) \mapsto (S^{-1}W_V, W_O S)$, with $S \in GL(d_h, \mathbb{R})$.

A convenient realization of this invariant is obtained by introducing matrices $A \in \mathbb{R}^{d\times d_h}$, $B \in \mathbb{R}^{d_h\times d}$, and defining $G_{VO} = AB$, which enforces $\mathrm{rank}(G_{VO}) \leqslant d_h$ by construction. The pair (A, B) provides a representation of the invariant operator G_{VO}, and it is therefore not unique. In particular, it admits a residual redundancy $(A, B) \mapsto (AS, S^{-1}B)$, with $S \in GL(d_h, \mathbb{R})$, which leaves the composite G_{VO} unchanged and pertains only to the chosen representation of G_{VO}.

Now, as far as differentiable implementation is concerned, we observe that the map $(A, B) \mapsto AB$ is smooth, so gradients of the loss with respect to A and B can be computed by standard automatic differentiation through the matrix product. No modification of backpropagation algorithms is required.

A natural initialization is then obtained by sampling A and B using standard variance-preserving random initializations (e.g., Xavier/Glorot-type schemes adapted to their dimensions), and setting $G_{VO}^{(0)} = A_0 B_0$.

Alternatively, one may treat G_{VO} itself as the optimization variable in $\mathbb{R}^{d\times d}$, perform gradient updates directly on G_{VO}, and enforce the rank constraint $\mathrm{rank}(G_{VO}) \leqslant d_h$ by projection (for instance via truncated SVD), as described in the Appendix. An initial $G_{VO}^{(0)}$ may again be obtained from a low-rank factorization.

This construction provides a concrete differentiable realization for a representative symmetry-reduced sector. It illustrates how invariant variables, gradients, and initialization can be handled within standard automatic-differentiation-based machine learning frameworks, yet without specifying a complete training procedure for full transformer architectures.

VI. CONCLUSIONS

We outline a conceptual framework for relational, symmetry-reduced transformer architectures, in which

(1) internal representation frames are dressed by replacing coordinate-dependent vectors with relational invariants, and attention mechanisms are rewritten in invariant relational form;

(2) optimization dynamics are studied on reduced parameter spaces, eliminating motion along redundant symmetry orbits.

Together, these would yield a framework in which representations, attention, optimization, structure are all formulated in manifestly relational way (with invariant weights and equivariant vector carriers).

Of course, both standard dot-product attention and explicit relational matrices involve pairwise token interactions and therefore retain an $O(n^2)$ interaction structure in sequence length. Standard implementations propagate per-token vectors and compute pairwise scores on the fly, whereas a fully relational formulation may choose to store relational objects across layers. Storing relational objects across depth can increase memory unless one uses low-rank/sparse parametrizations or recomputation strategies.

Empirical evaluation of these ideas is left for future work (possibly collaborative). In particular, whether symmetry reduction yields measurable gains in wall-clock training time, memory consumption, or convergence speed relative to standard transformer implementations is an open implementation-dependent question that requires dedicated benchmarking on shared tasks and matched model budgets. The present paper provides the conceptual and mathematical framework needed to formulate such comparisons precisely, but does not itself report experimental results.

For future practical implementations, beyond predictive accuracy one should also track resource and optimization metrics such as

(1) training loss curves (does the invariant version converge faster or more stably?);

(2) sensitivity to random initialization (variance across 20–50 seeds);

(3) effective rank of internal representations or parameter matrices (does symmetry reduction lead to lower effective dimensionality?);

(4) qualitative attention patterns (are they more interpretable in the relational case?).

Future directions include developing scalable implementations of invariant relational attention and designing optimizers that operate directly on reduced parameter spaces. We also note that introducing architectural inductive biases—for instance, through locality, hierarchy, or constrained attention patterns—is complementary in spirit to the present proposal. Such biases may improve optimization and generalization for reasons different from symmetry reduction, and studying their interplay with the reduced relational variables proposed here would be an interesting direction for future work.

More broadly, symmetry reduction provides a unifying lens for understanding representation and learning dynamics in modern ML. By removing redundant d.o.f., we move toward models whose internal structure directly reflects the relational organization of the data they process. We hope this framework opens avenues for both theoretical analysis and practical architecture design in deep learning.

A. Empirical scope and evaluation road map

A natural question is whether the symmetry-reduced formulations proposed here lead, in practice, to improved training efficiency, lower memory usage, or better predictive performance relative to standard transformer baselines. We emphasize that the present paper does not answer this experimentally. Its objective is to isolate the relevant symmetries, formulate invariant relational variables, and identify reduced parametrizations and projected dynamics at the conceptual and mathematical level.

Nevertheless, the framework suggests a concrete experimental program. A minimal empirical study would compare (1) a standard transformer baseline, (2) a relational/invariant scoring variant in model space, and/or (3) a symmetry-reduced optimization variant in head space, on the same task and with matched parameter budgets as closely as possible. Relevant measurements would include the following:

(1) predictive performance (validation/test loss or accuracy);
(2) wall-clock training time per epoch and to target loss;
(3) peak memory consumption;
(4) number of trainable parameters and effective rank of learned operators;
(5) sensitivity to initialization across multiple random seeds;
(6) optimizer statistics indicating motion along reduced versus redundant directions when projection-based schemes are used.

From a computational viewpoint, both standard dot-product attention and the relational scoring constructions discussed in this paper involve pairwise token interactions and therefore share the same $O(n^2)$ interaction structure in sequence length. However, their constant factors, memory access patterns, and implementation overheads may differ substantially. In particular, explicitly storing relational objects such as $G = XX^\top$ across layers may increase memory usage unless low-rank, sparse, kernelized, or recomputation strategies are adopted. Likewise, quotient- or projection-based optimization may reduce motion along redundant parameter directions, but whether this translates into lower wall-clock cost or improved convergence is an empirical question.

Accordingly, the present work should be read as providing the mathematical framework and architectural principles that make such an experimental comparison well posed, rather than as reporting the outcome of that comparison. A full empirical evaluation is left for future work, potentially in collaboration with researchers specializing in large-scale implementation and benchmarking. A possible starting point for such future benchmarking is to follow the experimental paradigm of Ref. [15], adapting it to compare standard transformer baselines with the symmetry-reduced variants proposed here under matched parameter and training budgets.

ACKNOWLEDGMENTS

J.F. was supported by the Austrian Science Fund (FWF), Grant No. P 36542, and by the Czech Science Foundation (GAČR), Grant No. GA24-10887S. L.R. was supported by the

research grant PNRR Young Researchers, funded by MUR, MSCA Seal of Excellence (SoE), CUP E13C24003600006, ID SOE2024_0000103, project GrIFOS, of which this paper is part.

Both authors share equal credit for the work done in this paper: J.F. and L.R. conceived of the presented idea and developed the formalism. Both J.F. and L.R. contributed equally to the writing of the article.

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX: ALGORITHMIC SKETCH: INVARIANT UPDATES FOR THE VALUE-OUTPUT SECTOR

This Appendix sketches an invariant (symmetry-reduced) optimization viewpoint for the value-output parameters ($W_V \in \mathbb{R}^{d_h \times d}$, $W_O \in \mathbb{R}^{d \times d_h}$) of a single attention head. As discussed in Sec. V, the forward contribution of this sector depends on (W_V, W_O) only through the composite

$$G_{VO} := W_O W_V \in \mathbb{R}^{d \times d}, \tag{A1}$$

which is invariant under the internal reparametrization

$$(W_V, W_O) \mapsto (S^{-1} W_V,\ W_O S), \quad S \in GL(d_h, \mathbb{R}). \tag{A2}$$

In the idealized attention submodule (i.e., in the absence of symmetry-breaking components such as LayerNorm or nonlinear MLPs), this composite therefore captures the functionally relevant d.o.f. of the value-output sector.

From the symmetry-reduction perspective advocated in this paper, it is natural to regard G_{VO} as the primary variable and to formulate learning directly in terms of this invariant. Here, let us also remark that, in practice, gradients with respect to the invariant composites can be obtained by standard automatic differentiation once a concrete differentiable realization of the reduced variables is chosen.

1. Invariant gradient and direct update on G_{VO}

Restricting attention to the symmetric submodule, the loss may be viewed as a function of the invariant,

$$\mathcal{L} = \mathcal{L}(G_{VO}), \tag{A3}$$

up to symmetry-breaking components such as LayerNorm, biases, and nonlinear MLPs (cf. Sec. III A 1). One may therefore define the invariant gradient

$$\nabla_{G_{VO}} \mathcal{L} \ \in \ \mathbb{R}^{d \times d} \tag{A4}$$

and perform symmetry-reduced gradient descent directly on G_{VO}:

$$G_{VO} \ \leftarrow \ G_{VO} - \eta \nabla_{G_{VO}} \mathcal{L}, \tag{A5}$$

with learning rate $\eta > 0$ ($\eta \in \mathbb{R}_{>0}$). This update evolves on the reduced parameter space and contains no motion along the internal $GL(d_h, \mathbb{R})$ symmetry directions, since these do not appear as independent variables.

2. Rank constraint and low-rank realizations

Since G_{VO} factors through a d_h-dimensional head space, it necessarily satisfies

$$\operatorname{rank}(G_{VO}) \leqslant d_h. \tag{A6}$$

Consequently, unrestricted updates of the form (A5) may leave the representable set unless this constraint is enforced [if one updated G_{VO} freely, it may become full rank, hence no longer realizable by any (W_O, W_V)]. Two standard strategies may be used to address this.

(1) *Explicit low-rank parametrization*: One may parametrize the invariant directly as

$$G_{VO} = AB, \quad A \in \mathbb{R}^{d\times d_h}, \quad B \in \mathbb{R}^{d_h\times d}, \tag{A7}$$

which enforces $\mathrm{rank}(G_{VO}) \leqslant d_h$ by construction. Gradient-based optimization is then carried out on (A, B), with the model depending on these variables only through their product AB. Note that A and B merely provide a low-rank realization of the invariant operator. The factorization itself retains a residual internal reparametrization freedom $(A, B) \mapsto (AS, S^{-1}B)$, $S \in GL(d_h, \mathbb{R})$, but this redundancy pertains only to the chosen representation of G_{VO}; the learned function depends solely on the invariant composite.

(2) *Projected invariant updates*: Alternatively, one may update G_{VO} in the ambient space via Eq. (A5) and subsequently project back onto the rank-$\leqslant d_h$ manifold. A simple projection is obtained via (rank-$\leqslant d_h$) truncated SVD:

$$G_{VO} = U\,\Sigma V^\top, \quad \Pi_{\leqslant d_h}(G_{VO}) := U_{d_h}\,\Sigma_{d_h}\,V_{d_h}^\top, \tag{A8}$$

where $G_{VO} = U\,\Sigma V^\top$ is an SVD with singular values in descending order, $U_{d_h} \in \mathbb{R}^{d\times d_h}$ and $V_{d_h} \in \mathbb{R}^{d\times d_h}$ denote the matrices formed by the first d_h columns of U and V, respectively, and $\Sigma_{d_h} \in \mathbb{R}^{d_h\times d_h}$ is the diagonal matrix of the largest d_h singular values. The SVD factorizes G_{VO} into orthogonal matrices U, V, and a diagonal matrix Σ of non-negative singular values, providing the optimal rank-$\leqslant d_h$ approximation in Frobenius norm (the truncated SVD gives the lowest possible total squared error when compressing G_{VO} to rank $\leqslant d_h$).

The invariant update then reads

$$G_{VO} \;\leftarrow\; \Pi_{\leqslant d_h}\big(G_{VO} - \eta\nabla_{G_{VO}}\mathcal{L}\big). \tag{A9}$$

In other words, one first performs a gradient step in the ambient space of $d \times d$ matrices and then projects back onto the rank-$\leqslant d_h$ manifold to ensure compatibility with the attention-head structure.

3. Choosing representatives for standard implementations

If one wishes to implement the forward pass in the conventional factorized form W_O, W_V, a representative factorization of the current invariant G_{VO} may be chosen at any stage. For example, from a truncated SVD $G_{VO} = U\,\Sigma V^\top$, with $\Sigma \in \mathbb{R}^{d_h\times d_h}$, one may set

$$W_O := U\,\Sigma^{1/2}, \quad W_V := \Sigma^{1/2}V^\top, \tag{A10}$$

so that $W_O W_V = G_{VO}$. This choice is not unique: Any pair $(W_O S, S^{-1}W_V)$, with $S \in GL(d_h, \mathbb{R})$, yields the same invariant. Of course, such choices correspond merely to different internal coordinate frames for the same relational operator.

In conclusion, in this symmetry-reduced formulation, the value-output sector is naturally described by the invariant composite $G_{VO} = W_O W_V$. Learning may therefore proceed directly on this object, with rank constraints enforced either by low-rank parametrization or by projection. This realizes symmetry reduction by construction: Optimization evolves only in functionally meaningful directions, without introducing preferred internal frames or explicit symmetry breaking.

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805.

[2] T. B. Brown *et al.*, Language models are few-shot learners, arXiv:2005.14165.

[3] A. Dosovitskiy *et al.*, An image is worth 16 × 16 words: Transformers for image recognition at scale, arXiv:2010.11929.

[4] A. Radford *et al.*, Learning transferable visual models from natural language supervision, arXiv:2103.00020.
[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, arXiv:1706.03762.
[6] W. Merrill, G. Weiss, Y. Goldberg, R. Schwartz, N. A. Smith, and E. Yahav, A formal hierarchy of RNNs and transformers, arXiv:2004.08500.
[7] T. Lian, Y. Wang, X. Liu, and X. Qiu, A survey of transformers, arXiv:2106.04554.
[8] N. Elhage *et al.*, A mathematical framework for transformer circuits, 2023, https://transformer-circuits.pub/2021/framework/index.html.
[9] T. S. Cohen and M. Welling, Group equivariant convolutional networks, arXiv:1602.07576.
[10] R. Kondor and S. Trivedi, On the generalization of equivariance and convolution in neural networks to the action of compact groups, arXiv:1802.03690.
[11] D. Kunin, J. Sagastuy-Brena, S. Ganguli, D. L. K. Yamins, and H. Tanaka, Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics, arXiv:2012.04728.
[12] H. Tanaka and D. Kunin, Noether's learning dynamics: Role of symmetry breaking in neural networks, arXiv:2105.02716.
[13] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, Sharp minima can generalize for deep nets, arXiv:1703.04933.
[14] B. Zhang, Z. Zheng, Z. Chen, and J. Li, Beyond the permutation symmetry of transformers: The role of rotation for model fusion, arXiv:2502.00264.
[15] E. Silverstein, D. Kunin, and V. Shyam, Symmetry breaking in transformers for efficient and interpretable training, arXiv:2601.22257.
[16] M. F. da Silva, F. Dangel, and S. Oore, Hide & seek: Transformer symmetries obscure sharpness & Riemannian geometry finds it, arXiv:2505.05409.
[17] B. Zhao, I. Ganev, R. Walters, R. Yu, and N. Dehmamy, Symmetries, flat minima and the conserved quantities of gradient flows, arXiv:2210.17216.
[18] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, A simple neural network module for relational reasoning, arXiv:1706.01427.
[19] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, Deep sets, arXiv:1703.06114.
[20] P. W. Battaglia *et al.*, Relational inductive biases, deep learning, and graph networks, arXiv:1806.01261.
[21] V. P. Dwivedi and X. Bresson, A generalization of transformer networks to graphs, arXiv:2012.09699.
[22] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, Do transformers really perform bad for graph representation? in *Advances in Neural Information Processing Systems* (NeurIPS, 2021).
[23] M. Henneaux and C. Teitelboim, *Quantization of Gauge Systems* (Princeton University Press, Princeton, NJ, 1992).
[24] V. Guillemin and S. Sternberg, *Symplectic Techniques in Physics* (Cambridge University Press, Cambridge, 1990).
[25] F. Gieres, Covariant canonical formulations of classical field theories, SciPost Phys. Lect. Notes, **77** (2023).
[26] J. François, Artificial versus substantial gauge symmetries: A criterion and an application to the electroweak model, Philos. Sci. **86**, 472 (2019).
[27] J. T. Francois André, The dressing field method for diffeomorphisms: A relational framework, J. Phys. A **57**, 305203 (2024).
[28] J. T. François and L. Ravera, Geometric relational framework for general-relativistic gauge field theories, Fortschr. Phys. **73**, 2400149 (2025).
[29] J. François and L. Ravera, Dressing fields for supersymmetry: The cases of the Rarita-Schwinger and gravitino fields, J. High Energy Phys. 07 (2024) 041.
[30] J. François and L. Ravera, Unconventional supersymmetry via the dressing field method, Phys. Rev. D **111**, 125022 (2025).
[31] J. François and L. Ravera, Reassessing the foundations of metric-affine gravity, Eur. Phys. J. C **85**, 902 (2025).

[32] J. François and L. Ravera, Mechanics as a general-relativistic gauge field theory, and relational quantization, arXiv:2510.19845.
[33] J. François and L. Ravera, Off-shell supersymmetry via manifest invariance, Phys. Lett. B **868**, 139633 (2025).
[34] J. François and L. Ravera, Raising galaxy rotation curves via dressing, Phys. Rev. D **112**, L081501 (2025).
[35] J. T. François and L. Ravera, Relational bundle geometric formulation of non-relativistic quantum mechanics, Fortschr. Phys. **73**, e70040 (2025).
[36] J. François and L. Ravera, Spacetime boundaries do not break diffeomorphism and gauge symmetries, Phys. Rev. D **112**, 125029 (2025).
[37] P. Berghofer, J. François, and L. Ravera, What price fiber bundle substantivalism? On how to avoid holes in fibers, arXiv:2505.12876.
[38] C. Rovelli, What is observable in classical and quantum gravity? Class. Quantum Grav. **8**, 297 (1991).
[39] C. Rovelli, Partial observables, Phys. Rev. D **65**, 124013 (2002).
[40] C. Rovelli, Why gauge? Found. Phys. **44**, 91 (2014).
[41] J. François and L. Ravera, On the meaning of local symmetries: Epistemic-ontological dialectics, Found. Phys. **55**, 38 (2025).
[42] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, Gauge equivariant convolutional networks and the icosahedral CNN, arXiv:1902.04615.
[43] E. Theodosis, D. E. Ba, and N. Dehmamy, Incorporating gauge-invariance in equivariant networks, OpenReview, 2024, https://openreview.net/forum?id=YAINolpm8n.
[44] E. Theodosis, D. Ba, and N. Dehmamy, Constructing gauge-invariant neural networks for scientific applications, in ICML 2024 Workshop GRaM, Vienna, Austria, 29th July 2024, https://openreview.net/forum?id=1fwEsyICb3.
[45] Y. Choi and C.-K. Kim, Gauge-equivariant graph networks via self-interference cancellation, arXiv:2511.16062.
[46] L. Huang, O. Balabanov, H. Linander, M. Granath, D. Persson, and J. E. Gerken, Learning Chern numbers of multiband topological insulators with gauge equivariant neural networks, arXiv:2502.15376.
[47] A. Strunk and R. Assam, Gauge flow models, arXiv:2507.13414.
[48] H. Honda, A gauge-theory-based graph neural network, OpenReview, 2026, https://openreview.net/forum?id=QxoyccprRp.
[49] P. Berghofer and J. François, Dressing vs. fixing: On how to extract and interpret gauge-invariant content, Found. Phys. **54**, 72 (2024).
[50] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, Gradient surgery for multi-task learning, in *Advances in Neural Information Processing Systems 33* (NeurIPS, 2020), pp. 5824–5836.