A CNN-ViT hybrid architecture search benchmark on a large-scale dataset

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 November 2025, accepted 3 December 2025, date of publication 10 December 2025,
date of current version 16 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3642734

RESEARCH ARTICLE

A CNN-ViT Hybrid Architecture Search Benchmark on a Large-Scale Dataset

LUCA ROBBIANO, FRANCESCA PISTILLI, AND GIUSEPPE AVERTA
Department of Control and Computer Engineering, Polytechnic University of Turin, 10129 Turin, Italy

Corresponding author: Luca Robbiano (luca.robbiano@polito.it)

This work was supported in part by the Future Artificial Intelligence Research (FAIR); and in part by European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR)—MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3—D.D. 1555 11/10/2022) under Grant PE00000013.

ABSTRACT In recent years, Neural Architecture Search (NAS) has emerged as a promising methodology to automate the design of deep neural networks, enabling the discovery of high-performing architectures across a wide range of tasks. Due to the high computational cost associated with NAS, several benchmarks have been introduced to support the development and evaluation of NAS methods. However, existing benchmarks are often limited in scope, typically relying on small-scale datasets or narrow search spaces, mostly based on Convolutional Neural Networks (CNNs) only. To address these limitations we introduce HyViTas-Bench, a novel NAS benchmark specifically tailored for hybrid CNN-Vision Transformer (ViT) architectures. HyViTas-Bench contains 6,561 unique models trained three times on a reduced, yet large scale, version of ImageNet-1k, offering an evaluation setting that better reflects realistic data. Each architecture is evaluated on 19 hardware platforms (CPU, GPU, and edge devices) for latency measurements, while robustness is validated through repeated training. We also provide an analysis of Out-of-Distribution (OoD) generalization using three external datasets. HyViTas-Bench enables a multifaceted assessment of NAS methods in terms of accuracy, latency, generalization capability, and model size. As such, it represents a valuable resource for advancing research on hybrid architectures and for facilitating the design and comparison of NAS strategies under more realistic and diverse evaluation criteria.

INDEX TERMS Computer vision, deep learning, neural architecture search, automl, benchmark.

I. INTRODUCTION

Deep Neural Networks have significantly advanced the field of computer vision, achieving state-of-the-art performance across various tasks, including image classification, object detection, and segmentation. Traditional Convolutional Neural Networks (CNNs) have long dominated computer vision due to their hierarchical feature extraction capabilities. However, with the advent of Vision Transformers (ViTs) [1], there has been a paradigm shift towards leveraging self-attention [2] mechanisms to capture long-range dependencies in images, often leading to superior performance in large-scale vision tasks. Despite their effectiveness, designing optimal neural network architectures remains a long and heuristic-driven process, motivating the rise of Neural Architecture Search (NAS) [3].

NAS automates the discovery of neural network architectures by exploring a predefined search space using various optimization techniques, including reinforcement learning, evolutionary algorithms, and differentiable search strategies. NAS has been instrumental in discovering highly efficient architectures tailored for specific datasets and hardware constraints [4], [5]. However, it is typically computationally expensive, requiring extensive model training and evaluation to assess candidate architectures [6], despite recent efforts to reduce this cost through training-free performance proxies [7], [8], [9], [10], [11]. Furthermore, the reproducibility and fair evaluation of NAS algorithms remain significantly challenging due to differences in experimental settings, search spaces, and evaluation methodologies [6], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Ye Liu.

To address these challenges, NAS benchmarks have emerged as a crucial resource for the research community [6], [13]. These benchmarks provide precomputed evaluations of architectures in a fixed search space, enabling rapid and fair comparisons of NAS algorithms without the need for exhaustive training. Notable NAS benchmarks, such as NAS-Bench-101 [14], NATS-Bench [15], [16], and NAS-Bench-360 [17], have facilitated research progress by reducing computational costs and enabling standardized comparisons. However, existing benchmarks predominantly focus on traditional CNN-based architectures [13], with limited exploration of more recent convolutional designs [18], [19] or attention-based search spaces [20] architectures, which are increasingly relevant for modern vision applications [21], [22], [23].

In this work, we introduce a novel NAS benchmark specifically designed for hybrid CNN-ViT architectures. It systematically evaluates models that integrate convolutional and transformer-based blocks on a large-scale dataset, enabling a comprehensive and realistic assessment of their performance.

The main contributions of our work can be summarized as follows:

- we propose HyViTas-Bench, the first NAS benchmark combining a hybrid convolutional-ViT search space with models trained on a large-scale dataset, comprising 6,561 architectures each trained three times;
- we show that our reduced ImageNet-based training set (RedImageNet) provides a more reliable proxy for large-scale performance than CIFAR-10/100 and ImageNet-16-120 [24], especially near the search-space optimum;
- we evaluate Out-of-Distribution (OoD) generalization on three additional datasets derived from standard OoD benchmarks ImageNet-Sketch [25], Stylized-ImageNet [26] and ImageNet-C [27], enabling robustness-aware NAS research;
- we provide pre-computed latency metrics on 19 heterogeneous hardware platforms, together with six training-free proxies and full accuracy logs, supporting both efficiency-aware and proxy-based NAS.

II. RELATED WORKS

A. HYBRID VIT

Vision Transformers (ViTs) [1] have achieved remarkable success in computer vision but lack CNNs inductive biases [28], requiring more data and compute to match CNN performance. Additionally, their quadratic attention complexity poses challenges for deployment in resource-constrained environments, especially when working with high-resolution features.

To address these limitations, researchers have improved attention efficiency (e.g., Swin Transformer [29]) and tried to leverage the advantages of both architectures [30] by exploring CNN-ViT hybrids [31], which integrate convolutional biases into Transformers. CoAtNet [32] exemplifies this approach, combining Inverted Bottleneck (IBN) [18] blocks in early stages with Transformers blocks later in the network, achieving superior performance over both pure CNNs and ViT models. Similarly, LocalViT [33] alternates convolutional and attention layers while replacing MLPs with IBNs to enhance locality.

For deployment in low-power and mobile settings, models like MobileFormer [34] and MobileViT [35] combine CNN efficiency with Transformer-based global context modeling, achieving strong performance under tight computational constraints. These architectures demonstrate that careful integration of convolution and attention mechanisms enables improved speed-accuracy trade-offs.

While numerous hybrid design strategies have been proposed, the space of architectural configurations within such models remains underexplored in a systematic and scalable way. To address this gap, we introduce the first benchmark specifically designed to evaluate hybrid CNN-ViT architectures, enabling controlled exploration of key architectural parameters such as convolutional kernel size, inverted bottleneck expansion ratio, and block output dimensions.

B. NAS BENCHMARKS

Neural Architecture Search (NAS) has significantly advanced the field of deep learning by automating the design of network architectures [4], [5]. However, NAS algorithms typically demand considerable computational resources due to the extensive training and evaluation of candidate architectures. To address these challenges, NAS benchmarks have been proposed to support reproducible research and fair comparisons by providing precomputed performance metrics across standardized search spaces [13].

One of the pioneering NAS benchmarks is NAS-Bench-101 [14], comprising approximately 423,000 unique architectures evaluated on CIFAR-10 [41]. Its cell-based search space consists of Directed Acyclic Graphs (DAGs) with a fixed number of nodes and edges. Despite its foundational contribution, NAS-Bench-101 is limited by its exclusive focus on a small-scale dataset such as CIFAR-10.

To extend benchmark versatility, NAS-Bench-201 [15] introduced a search space with 15,625 architectures, evaluated across multiple datasets (CIFAR-10, CIFAR-100, and ImageNet-16-120 [24]). Notably, although NAS-Bench-201 comprises 15,625 architectures, the number of unique architectures is effectively reduced to 6,466 due to isomorphic duplicates [42]. Subsequently, NATS-Bench [16] expanded the NAS-Bench-201 search space by varying the channel numbers, thus increasing the number of architectures to more than 32,000.

In parallel, the demand for hardware-aware NAS benchmarks led to the development of HW-NAS-Bench [36], which extends NAS-Bench-201 by incorporating hardware efficiency metrics, such as latency and energy consumption, across various computing platforms. LatBench [37], another significant contribution, specifically addresses

TABLE 1. Comparison of existing NAS benchmarks. HyViTas-Bench is the first to target hybrid CNN-ViT architectures on large-scale data (RedImageNet), while also supporting out-of-distribution (OoD) accuracy, hardware metrics, and precomputed training-free proxies in a unified setting. See [13] for an in-depth survey.

Benchmark	# Arch.	Dataset(s)	Family	ID Accuracy	OoD Accuracy	HW metrics	Proxy metrics	Multiple runs
Nas-Bench-101 [14]	423k	CIFAR-10	CNN	✓	✗	✗	✗	✓
NAS-Bench-201 [15]	6.5k	CIFAR, ImageNet-16	CNN	✓	✗	✗	✗	✓
NATS-Bench [16]	6.5k + 32k	CIFAR, ImageNet-16	CNN	✓	✗	✓	✗	✓
HW-NAS-Bench [36]	6.5k	CIFAR, ImageNet-1k	CNN	✗	✗	✓	✗	✗
LatBench [37]	15k	CIFAR, ImageNet-1k	CNN	✗	✗	✓	✗	✗
NAS-Bench-MR [38]	2.5k	ImageNet-1k, KITTI [39], Cityscapes [40]	CNN	✓	✗	✓	✗	✗
OoD-ViT-NAS [20]	3k	ImageNet-1k	ViT	✓	✓	✓	✓	✗
HyViTas-Bench (ours)	**6.6k**	**RedImageNet**	**CNN / ViT**	✓	✓	✓	✓	✓

runtime latency, evaluating models across multiple devices, including embedded platforms and mobile GPUs. NAS-Bench-MR [38] further expanded the benchmarking scope by enabling multi-resolution search for diverse computer vision tasks, including image segmentation, video recognition, and 3D detection, although each architecture is trained only once per task, without repeated runs to assess variance.

Despite the considerable contributions of these benchmarks, most are predominantly oriented towards CNN architectures, thus lacking evaluations on modern architectural paradigms. With ViTs and hybrid CNN-ViT architectures gaining prominence, a notable gap remains in systematically benchmarking NAS methods targeting these emerging architectures. Recently, OoD-ViT-NAS [20] provided a benchmark specifically thought for Vision Transformers, focusing on the analysis of their generalization capabilities under out-of-distribution (OoD) conditions. Their study also examines training-free performance predictors, also known as zero-cost proxies, and finds that such metrics struggle to correlate with OoD accuracy, with model size emerging as the most reliable indicator. However, similarly to NAS-Bench-MR, OoD-ViT-NAS does not perform multiple independent training runs for each architecture, thus limiting the assessment of robustness to different initializations. In the case of OoD-ViT-NAS, this is due to one-shot supernet weight sharing, an approach that, while greatly reducing computational cost, is known to suffer from weight entanglement, as the shared parameters of sub-networks can introduce bias and lead to inaccurate performance estimation [43], [44].

Our proposed benchmark addresses these limitations by introducing, for the first time, a comprehensive evaluation of hybrid CNN-ViT architectures trained on a large-scale dataset derived from ImageNet-1k [45]. Unlike most prior benchmarks, constrained to small-scale datasets, ours includes 6,561 unique hybrid architectures, each independently trained three times on a large-scale dataset to ensure statistical robustness. This setup enables consistent comparisons and better reflects the challenges of large-scale visual tasks. A summary of how our benchmark relates to existing ones can be found in Table 1, and we refer the reader to [13] for a broader survey on NAS benchmarks.

III. THE HYVITAS-BENCH BENCHMARK

A. SEARCH SPACE

The search space of our proposed benchmark HyViTas-Bench is structured following a four-stage hybrid architecture (Fig. 1), based on the sequential integration of convolutional and transformer blocks [32], [47], [48]. The architecture comprises a fixed Stem module based on a non-overlapping convolution with kernel size 4, inspired by EdgeNeXt [46], and a fixed fully convolutional classification head. Only the four intermediate stages are searchable, divided into two convolutional and two transformer-based stages. The first two searchable stages are purely convolutional, composed of three modern ConvNeXt [19] blocks each. These blocks were proposed as an extension to the residual convolutional paradigm introduced by ResNet [49], integrating large-kernel depth-wise convolutions and Layer Normalization techniques [50] to incorporate design principles from Vision Transformers and utilizing GELU [51] as the activation function. In our implementation we use standard Batch Normalization [52] in place of Layer Normalization to improve efficiency at inference time. The last two searchable stages employ transformer blocks [2], each configured with a fixed number of 8 attention heads. Inspired by [33], each transformer block is complemented by a ConvNeXt block (Fig. 2), effectively combining convolutional inductive biases with the global receptive fields offered by attention mechanisms. This configuration defines a well-structured yet practical search space for hybrid CNN-Transformer models. By design, it balances diversity with tractability: the choice of blocks across stages prevents a combinatorial explosion that would render exhaustive exploration of the search space impractical, while still enabling controlled architectural variations across the hybrid CNN-Transformer design. It facilitates exploration of different capacity allocations and receptive field sizes, preserving consistency between stages. The specific values considered for kernel sizes, expansion factors, and

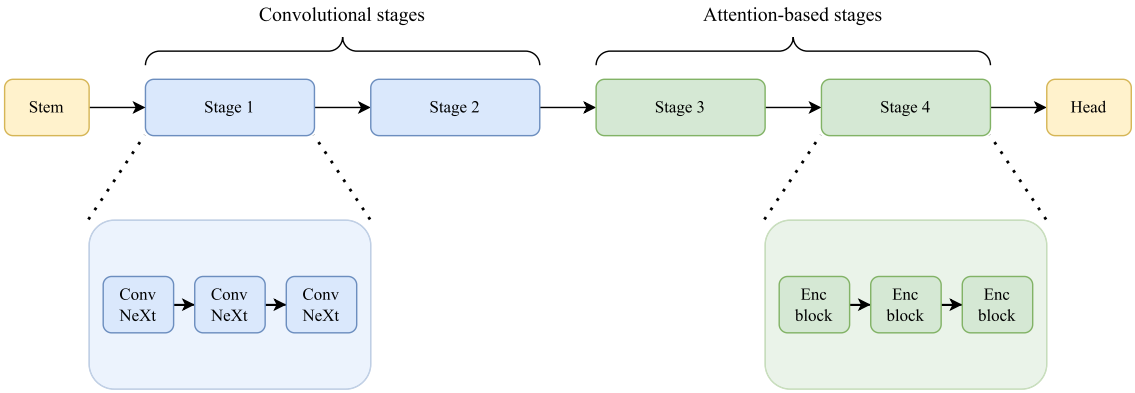

FIGURE 1. Overview of the four-stage hybrid CNN-ViT architecture defining the search space of HyViTas-Bench. It comprises a fixed Stem module inspired by EdgeNeXt [46], two searchable convolutional stages with ConvNeXt [19] blocks, two searchable transformer-based stages inspired by LocalViT [33], and a fixed classification head. Searchable parameters include kernel size, expansion factor (shared within each stage type), and output depth (independently searched for each stage). The architecture of the encoder blocks is detailed in Fig. 2.

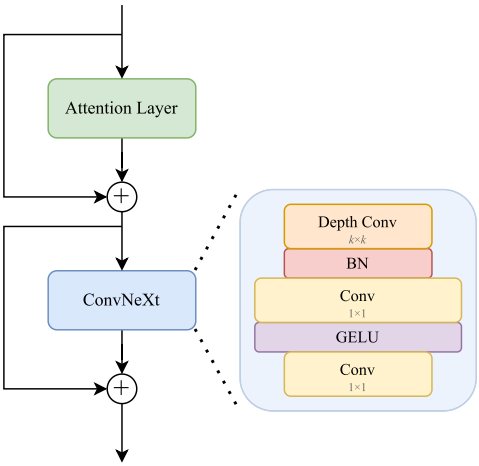

FIGURE 2. Architecture of a transformer-based encoder block from the attention-driven stages. The block comprises an attention layer followed by a ConvNeXt [19] block, each equipped with residual connections. The right inset illustrates the internal structure of the ConvNeXt block, featuring the inverted bottleneck design and depthwise convolution with searchable kernel size.

output depths are reported in Table 2. Full implementation details will be released alongside the benchmark to support reproducibility.

B. DATASETS AND PREPROCESSING

The high computational cost of training a vast number of models on large-scale datasets has historically constrained tabular Neural Architecture Search (NAS) benchmarks to small-scale datasets such as CIFAR-10 or CIFAR-100. Notably, to mitigate this computational burden, NAS-Bench-201 and NATS-Bench employ a significantly reduced version of ImageNet-1k [45], ImageNet-16-120 [24], obtained by downscaling the images to a resolution of 16×16 pixels and by reducing the dataset to 120 classes. However, reliance on these smaller datasets introduces a fundamental limitation, as they often fail to adequately represent the complexity required for reliable predictions of real-world

TABLE 2. Searchable dimensions in HyViTas-Bench search space. Kernel size and expansion factor are searched independently for the first pair and last pair of stages, resulting in a total search space size of $3^8 = 6,561$.

Dimension	Allowed values
Kernel size	[3, 5, 7]
Expansion factor	[1, 2, 4]
Output depth (stage 1)	[16, 32, 48]
Output depth (stage 2)	[64, 80, 96]
Output depth (stage 3)	[112, 128, 144]
Output depth (stage 4)	[192, 224, 256]

generalization [13]. Given the increasing complexity of modern neural architectures, evaluating NAS methods on more realistic datasets that are able to better represent full-scale data, has become increasingly important.

To address this gap, our novel benchmark evaluates architectures on a large-scale dataset which, following the best practice suggested by [53], consists of a reduced version of ImageNet-1k comprising 500 randomly selected classes, with all images downscaled to a maximum resolution of 192×192 pixels. This dataset achieves a balance between computational feasibility and representativeness, offering a diverse range of image statistics while remaining practical for large-scale NAS experimentation. The dataset contains 523,901 training images and 25,000 validation images, with an overall storage footprint of less than 20 GB, making it significantly smaller than the full ImageNet-1k dataset (146 GB) in terms of storage requirements, but still reasonably large-scale in terms of number of samples and resolution if compared to CIFAR or ImageNet-16-120.

By training all architectures on this dataset, our benchmark supports a more realistic assessment of generalization performance, narrowing the gap between existing NAS benchmarks and practical deployment conditions. This setting enables more meaningful comparisons and supports the development of NAS methods suited for large-scale, real-world applications.

In order to assess the performance of each architecture under out-of-distribution (OoD) conditions, we further evaluate all models on three additional datasets derived from established OoD benchmarks by applying the same reduction strategy as in the in-distribution training set, selecting the same 500 classes and downscaling all images to a maximum resolution of 192 × 192 pixels:

- **ImageNet-Sketch** [25]: a domain generalization benchmark composed of sketch-style black-and-white illustrations of ImageNet-1k classes. Unlike natural photographs, these images lack texture and color, placing emphasis on the model ability to extract semantic information from shape and contour. This dataset is particularly challenging due to its abstract and texture-free representations.
- **ImageNet-C** [27]: a corruption robustness benchmark in which 19 types of algorithmic corruptions (15 ''core'' and 4 ''extra''), ranging from noise and blur to weather and digital artifacts, are applied to the ImageNet-1k validation set. Each corruption type is applied at five severity levels. For our benchmark, we evaluate each model under levels from 1 (minimal) to 5 (severe) for each corruption type to assess robustness across different perturbation intensities. Unless otherwise specified, RedImageNet-C results in tables and figures are averaged over the 15 core corruption types; full per-corruption and per-severity results are included in the benchmark data.
- **Stylized-ImageNet** [26]: a dataset generated by replacing the original textures of ImageNet-1k images with the style of randomly sampled paintings via AdaIN [54] style transfer, while broadly preserving object shape and global image structure. This induces a strong shape-bias, making it the most challenging dataset in our evaluation for assessing robustness.

In the remainder of this work we denote the reduced, computationally-lighter variants of the ImageNet-Sketch, ImageNet-C, and Stylized-ImageNet test datasets by adding the *Red-* prefix (e.g. RedImageNet-C). For compactness in tables, we abbreviate OoD dataset names to RedIN-Sketch, RedIN-C and RedStylized-IN. This additional evaluation enables a comprehensive analysis of architecture robustness and generalization beyond the training distribution, complementing the in-distribution performance assessment and supporting the study of domain shift resilience within NAS. While HyViTas-Bench focuses on natural images, it could be extended to other domains (e.g. medical or satellite imagery) that lie outside the scope of the present work. Given the high cost of exhaustively training the full search space, this could be done by training a small stratified subset of architectures on the target dataset and fitting a cross-dataset surrogate to estimate the remaining entries by employing ground truth provided by HyViTas-Bench.

C. TRAINING AND EVALUATION PROTOCOL

All architectures in the benchmark are trained on the RedImageNet dataset described in Section III-B. Each model is trained for 200 epochs using the AdamW [55] optimizer with a batch size of 1024. We do not include low-fidelity trainings with fewer epochs in this release, as such approximations can distort the relative ranking of architectures; our goal is to provide a faithful and stable ground truth. Nevertheless, the released full-fidelity benchmark can be used to explore acceleration strategies, for example to calibrate short evaluations when probing larger spaces or additional datasets. The learning rate follows a cosine annealing schedule [56], starting from an initial value of 0.001, preceded by 5 warmup epochs with a linear ramp-up from an initial value of 5×10^{-5}. All training is performed using mixed-precision computation to improve throughput and memory efficiency. The data augmentation pipeline follows the RandAugment [57] policy, applied after a random resized crop to 160×160 pixels. Horizontal flipping is applied with a probability of 0.5. Additional training hyperparameters and configuration files are available alongside the released code for full reproducibility. For each architecture, training is repeated across three independent runs using different random seeds to account for variance due to stochastic optimization and different initializations, resulting in a total of 19,683 trained models. Evaluation is conducted on the validation set using both top-1 and top-5 accuracy metrics. All models are trained using PyTorch 2.2 on a distributed cluster equipped with NVIDIA A100 accelerators. Overall, generating the complete benchmark requires approximately 338,000 GPU-hours. More details about the computational cost can be found in Appendix B. In addition to accuracy-based evaluation, we provide inference latency for each architecture. Since inference time is inherently hardware-dependent, we measure latency across 19 heterogeneous hardware platforms, spanning server-grade and consumer GPUs, edge devices, and general-purpose CPUs based on both x86 and ARM architectures. For each device we run 52 forward passes using batches of a single image. The first 2 runs are discarded to eliminate initialization effects such as kernel loading, clock ramp-up, and memory allocation overhead. Latency is then computed as the average over the remaining 50 steps. All latency measurements are conducted using PyTorch 2.6 to ensure consistency across devices. The complete list of tested devices can be found in Appendix A.

IV. BENCHMARK ANALYSIS AND EXPERIMENTAL EVALUATION

A. ARCHITECTURE DIVERSITY

To be effective, a NAS benchmark must contain a diverse range of architectures, including both high-performing and suboptimal models. While a positive correlation between model size and final accuracy is expected, the search space should be designed to discourage trivial optimization strategies such as simple parameter maximization from consistently yielding the best-performing architectures. A well-structured space ensures that performance improvements arise from meaningful architectural choices rather than brute-force scaling. To illustrate this, Fig. 3 shows the

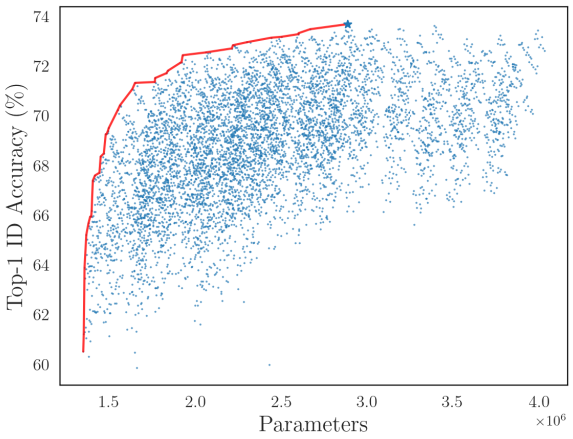

FIGURE 3. Correlation between model size and their final validation accuracy, averaged over three runs. Outliers achieving a performance below 50% are not shown in the plot, being less than 1% of the models. The red line highlights the Pareto front, and a star (★) indicates the optimal architecture in the search space.

relationship between model size and the average validation accuracy across all architectures in our benchmark, with Pareto-optimal models highlighted. Although the largest architecture contains 4M parameters, peak performance is achieved by a model with only 2.9M parameters. Fig. 4 complements this analysis by visualizing the distribution of validation accuracies and training stability. The average top-1 accuracy is 68.60, and most architectures achieve accuracies within the range of 60%-74%, with the top-performing model reaching 73.66%. A small subset performs significantly worse, clustering between 40% and 47%. We observe that these architectures share specific structural traits, namely a first stage with output dimension 16, an expansion factor of 1, and a kernel size of 3, suggesting limited representational capacity. Regarding stability, measured as the standard deviation across three independent training runs, most models show low variance ($\sigma \approx 0.2$), indicating a high degree of consistency and robustness to different initializations. However, a minority exhibits substantially higher variance, suggesting sensitivity to initial conditions. Notably, some architectures proved to be particularly difficult to train, frequently experiencing divergence or NaN values, indicating poor trainability and highlighting a strong sensitivity to initialization conditions. We find that these unstable models often share the same structural pattern found in low-performing ones, with the combination of minimal output dimension, kernel size, and expansion factor in the initial stage likely acting as a bottleneck during training. To assess stability and trainability, our benchmark provides queryable information on the number of training attempts that have been required to obtain three successfully converged models, offering insights into the trainability of different architectures.

B. QUALITY OF PROXY DATASET

To demonstrate that an architecture achieving high performance on a small-scale dataset such as CIFAR-10 does not necessarily generalize optimally to a larger dataset, particularly a large-scale benchmark such as ImageNet-1k, we evaluate the consistency of architectural rankings across different datasets. Specifically, we randomly select 20 architectures from our benchmark and train them on CIFAR-10, CIFAR-100, ImageNet-16-120, and the full ImageNet-1k dataset, following the same training protocol described in Section III-C, with the number of training epochs adjusted to match dataset size: 100 epochs for the three small-scale datasets and 300 epochs for ImageNet-1k. The results, presented in Table 3, indicate that the positive correlation between ImageNet-1k and our proxy dataset described in Section III-B consistently surpasses the correlations obtained with smaller-scale datasets such as CIFAR-10, CIFAR-100, and ImageNet-16-120. To further assess the robustness of our proposed benchmark dataset RedImageNet as a proxy for performance on large-scale datasets, we introduce a more challenging setting by randomly sampling 20 architectures from the top 15% best-performing models in our benchmark. This scenario is particularly challenging, as top-performing architectures tend to have similar performance levels, making performance ranking more sensitive to noise. Notably, even in this more challenging scenario where architectures are drawn exclusively from the top-performing subset, our proxy dataset maintains a statistically significant positive correlation with ImageNet-1k, whereas the correlations for other small-scale datasets deteriorate and fail to remain positive. While prior studies such as [53] have shown that individual architectures found on small-scale datasets such as CIFAR can yield competitive performance on large-scale datasets like ImageNet-1k or even ImageNet-22k [58], our analysis emphasizes that such small-scale datasets fail to preserve relative performance rankings. These findings suggest that our benchmark constitutes a more reliable and scalable proxy for ImageNet-1k performance, providing a practical alternative for evaluating architectures in large-scale settings while significantly reducing the computational cost associated with full ImageNet-1k training.

TABLE 3. Kendall-τ and Spearman-ρ correlations between architecture rankings on CIFAR-10, CIFAR-100, ImageNet-16-120, the proxy underlying our benchmark RedImageNet, and the full ImageNet-1k.

Dataset	Random		Random (top 15%)	
	τ	ρ	τ	ρ
CIFAR-10	0.495	0.662	-0.221	-0.253
CIFAR-100	0.411	0.571	-0.147	-0.230
ImageNet-16-120	0.421	0.592	-0.105	-0.174
RedImageNet	**0.758**	**0.898**	**0.332**	**0.503**

C. ACCURACY-DRIVEN BASELINES

To demonstrate the utility of our benchmark for evaluating NAS algorithms, we include six standard baselines: Random Search, Regularized Evolution [59], REINFORCE [60], BANANAS [61], DARTS [62] and SPOS [63].

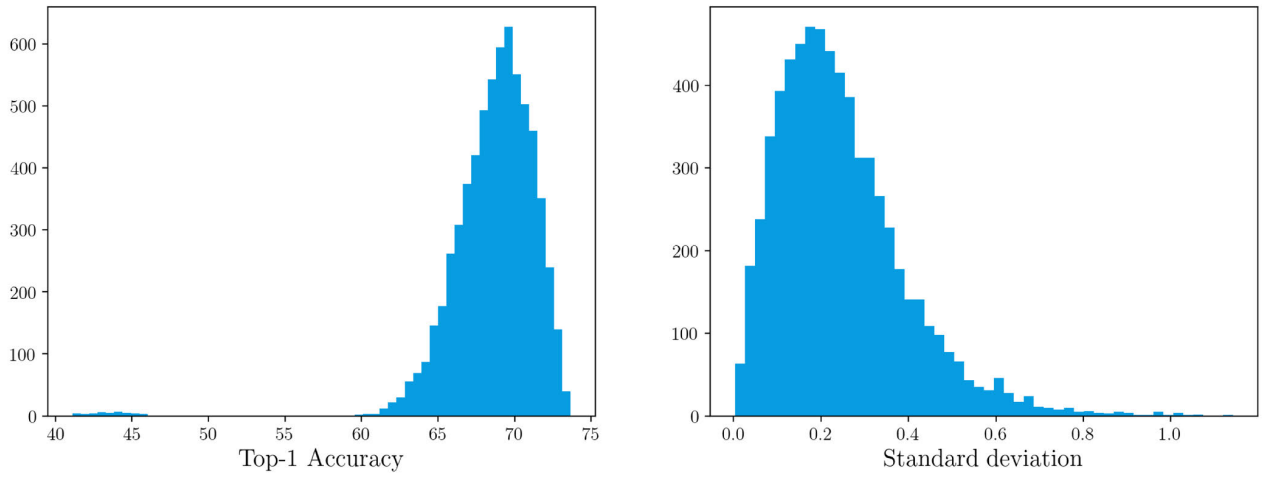

FIGURE 4. Distribution of average final validation top-1 accuracies (left) and standard deviations over three runs (right) across architectures in the benchmark. Standard deviation reflects stability with respect to random initialization: lower values indicate more stable architectures. A small number of outliers are excluded for visualization clarity.

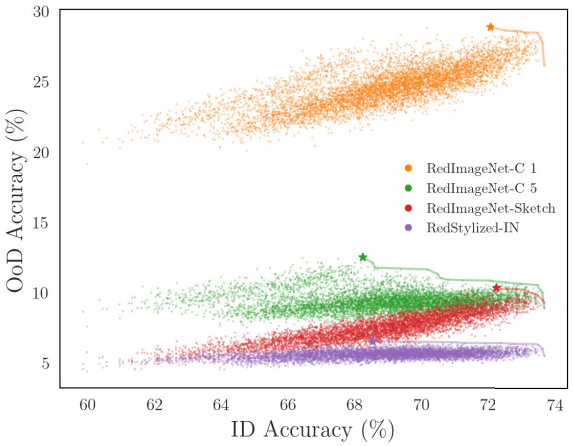

FIGURE 5. Scatter plot of in-distribution (ID) accuracy on RedImageNet versus out-of-distribution (OoD) accuracy on the reduced versions of ImageNet-Sketch, Stylized-ImageNet and ImageNet-C (corruption levels 1 and 5). For RedImageNet-C, accuracies are averaged over 15 corruption types. Stars (★) indicate the best architectures for each OoD dataset, which do not coincide with the top-ID ones. The Pareto front is highlighted for each OoD dataset. Kendall-τ values are 0.68 (RedImageNet-Sketch), 0.25 (RedStylized-ImageNet), 0.57 (RedImageNet-C 1), and 0.09 (RedImageNet-C 5).

These methods are widely adopted in NAS literature due to their ease of implementation and competitive performance in several benchmarks such as NAS-Bench-101 and NAS-Bench-201.

Random Search (RS) serves as a minimal baseline, where architectures are sampled uniformly at random from the search space. Despite its simplicity, it has been shown to be a surprisingly strong competitor in constrained NAS settings. Given the total cardinality of our search space (6,561 architectures), we sample 100 unique architectures, evaluate their validation accuracy and report the best performance.

Regularized Evolution with Aging (REA) implements a population-based search algorithm where new architectures are generated by mutating existing ones, and the population is periodically refreshed by removing the least promising candidates. This method has demonstrated robustness and consistency across various NAS benchmarks and requires no gradient or model-based guidance. In our setup we use a population of 50 architectures and a tournament size of 10 candidates to select parent architectures to mutate. We enforce a limit of 100 architecture evaluations.

REINFORCE is a policy-gradient algorithm that maintains a parametric distribution over architectural choices and updates it to increase the likelihood of high-performing architectures. At each iteration, an architecture is sampled from the current policy, evaluated, and used to compute a gradient update based on the reward signal. We apply REINFORCE with a fixed budget of 100 architecture evaluations, using the same evaluation protocol as for other baselines.

BANANAS performs Bayesian optimization over architectures by training an ensemble of lightweight predictors on an encoded representation of each architecture. The predictors are then used to select promising candidates. At each iteration we generate a candidate pool by mutating the most promising architectures, score all the candidates with the regressor and evaluate the $k = 10$ ones with the best acquisition values. We initialize with 10 random architectures and use a pool of 512 candidates per round. We evaluate BANANAS only when the objective requires training-based evaluation (ID or ID+OoD accuracy). We do not apply BANANAS to training-free proxies for zero-shot search, as its model-based surrogate would approximate a quantity that is both inexpensive to query directly and already a proxy approximating the true target.

DARTS is a differentiable NAS method that relaxes discrete architectural choices into continuous mixtures,

TABLE 4. Top-1 ID and OoD accuracies (mean ± standard deviation) for Random Search, Regularized Evolution (REA), REINFORCE and BANANAS across 500 trials, using different optimization targets. Accuracy-based methods optimize either ID accuracy or a combined objective ("ID+OoD"), defined as the average of ID accuracy and mean accuracy across four OoD datasets: RedImageNet-C level 1, RedImageNet-C level 5, RedImageNet-Sketch and RedStylized-ImageNet. Zero-shot methods use standard training-free proxies as optimization targets. Additionally, we report results with one-shot methods DARTS and SPOS across five runs targeting ID accuracy. The "Optimum" row indicates the best result observed in the benchmark for each dataset, while "Search space mean" reports the average performance and variability across all the architectures in the search space.

Method	Optimization target	ID Accuracy	OoD Accuracy			
			RedIN-C 1	RedIN-C 5	RedIN-Sketch	RedStylized-IN
Accuracy-based						
Random Search	ID	73.03 ± 0.40	27.22 ± 0.70	9.70 ± 0.39	9.20 ± 0.42	5.80 ± 0.21
REA [59]		73.36 ± 0.32	27.35 ± 0.67	9.69 ± 0.44	$\mathbf{9.33 \pm 0.32}$	5.82 ± 0.21
REINFORCE [60]		73.00 ± 0.39	27.18 ± 0.69	9.68 ± 0.41	9.21 ± 0.42	5.79 ± 0.22
BANANAS [61]		$\mathbf{73.44 \pm 0.29}$	27.29 ± 0.69	9.64 ± 0.46	9.32 ± 0.30	5.80 ± 0.21
DARTS [62]		72.89 ± 0.24	27.42 ± 0.34	9.94 ± 0.34	8.86 ± 0.27	5.85 ± 0.21
SPOS [63]		71.92 ± 1.00	$\mathbf{27.48 \pm 0.67}$	$\mathbf{10.08 \pm 0.67}$	9.03 ± 0.32	$\mathbf{6.00 \pm 0.21}$
Random Search	ID + OoD	72.97 ± 0.42	27.59 ± 0.59	9.93 ± 0.38	9.33 ± 0.39	5.90 ± 0.20
REA		73.26 ± 0.32	28.03 ± 0.42	10.15 ± 0.32	9.51 ± 0.36	6.02 ± 0.18
REINFORCE		72.97 ± 0.41	27.57 ± 0.59	9.92 ± 0.37	9.32 ± 0.38	5.90 ± 0.21
BANANAS		$\mathbf{73.32 \pm 0.31}$	$\mathbf{28.07 \pm 0.38}$	$\mathbf{10.18 \pm 0.29}$	$\mathbf{9.55 \pm 0.34}$	$\mathbf{6.04 \pm 0.18}$
Zero-shot						
Random Search	NASWOT [7]	72.98 ± 0.42	27.30 ± 0.70	9.72 ± 0.39	9.11 ± 0.36	5.74 ± 0.21
REA		$\mathbf{73.18 \pm 0.38}$	$\mathbf{27.80 \pm 0.59}$	9.88 ± 0.32	$\mathbf{9.22 \pm 0.29}$	5.81 ± 0.16
REINFORCE		72.96 ± 0.42	27.21 ± 0.72	9.68 ± 0.39	9.12 ± 0.41	5.71 ± 0.22
Random Search	Synflow [64]	71.78 ± 1.05	26.75 ± 0.83	9.75 ± 0.44	8.53 ± 0.54	5.70 ± 0.25
REA		72.46 ± 0.56	27.34 ± 0.64	9.90 ± 0.35	8.76 ± 0.30	5.62 ± 0.25
REINFORCE		71.84 ± 0.99	26.77 ± 0.81	9.75 ± 0.43	8.58 ± 0.48	5.70 ± 0.24
Random Search	LogSynflow [65]	72.02 ± 0.89	26.83 ± 0.82	9.75 ± 0.40	8.62 ± 0.44	5.68 ± 0.23
REA		72.46 ± 0.52	27.28 ± 0.71	9.85 ± 0.39	8.69 ± 0.31	5.59 ± 0.24
REINFORCE		72.04 ± 0.86	26.85 ± 0.84	9.76 ± 0.40	8.63 ± 0.44	5.69 ± 0.24
Random Search	DSS [66]	71.67 ± 1.02	26.69 ± 0.85	9.75 ± 0.43	8.52 ± 0.50	5.72 ± 0.25
REA		72.33 ± 0.66	27.40 ± 0.60	9.96 ± 0.35	8.73 ± 0.30	5.67 ± 0.25
REINFORCE		71.65 ± 1.05	26.71 ± 0.83	9.76 ± 0.43	8.48 ± 0.50	5.73 ± 0.24
Random Search	Jacobian [67]	70.01 ± 2.02	25.48 ± 1.16	9.66 ± 0.41	7.83 ± 0.93	5.71 ± 0.20
REA		70.34 ± 1.43	25.58 ± 0.92	9.61 ± 0.33	8.08 ± 0.63	5.78 ± 0.22
REINFORCE		69.91 ± 2.08	25.49 ± 1.13	9.68 ± 0.42	7.83 ± 0.93	5.71 ± 0.20
Random Search	SNIP [68]	72.33 ± 0.68	27.09 ± 0.79	9.83 ± 0.41	8.84 ± 0.44	5.76 ± 0.23
REA		72.64 ± 0.45	27.56 ± 0.53	9.97 ± 0.30	8.78 ± 0.28	5.67 ± 0.24
REINFORCE		72.40 ± 0.68	27.11 ± 0.79	9.83 ± 0.41	8.84 ± 0.42	5.75 ± 0.25
Random Search	# Parameters	71.35 ± 1.32	26.97 ± 0.78	$\mathbf{10.03 \pm 0.61}$	8.33 ± 0.70	$\mathbf{5.83 \pm 0.23}$
REA		72.40 ± 0.82	27.37 ± 0.58	9.88 ± 0.39	8.82 ± 0.47	5.67 ± 0.23
REINFORCE		71.30 ± 1.26	26.92 ± 0.87	10.01 ± 0.55	8.32 ± 0.71	5.81 ± 0.24
Search space mean	-	68.60 ± 3.26	24.52 ± 1.64	9.39 ± 0.72	7.48 ± 1.00	5.59 ± 0.32
Optimum	-	73.66	28.86	12.49	10.33	6.52

enabling gradient-based optimization. It adopts a bi-level scheme in which network weights w are updated to minimize $\mathcal{L}_t(w, \alpha)$, while architecture parameters α are updated on a held-out validation data split to minimize $\mathcal{L}_v(w, \alpha)$. To evaluate DARTS on HyViTas-Bench we instantiate DARTS in a layer-wise fashion: learnable logits α parametrize each of the eight design decisions described in Table 2. We alternate updates of w and α and discretize at the end. The resulting architecture is then evaluated by querying the benchmark table for its accuracy.

SPOS trains a weight-sharing supernet by activating one candidate path per batch, sampled uniformly from the search space. At each iteration, only weights on the selected path are updated. After training, candidate architectures inherit the supernet weights and are evaluated after a short recalibration of normalization layers. The final architecture is selected via a simple evolutionary search.

We perform experiments under two evaluation modes:

- *ID*: architectures are evaluated only on the validation set of RedImageNet;
- *ID+OoD*: both in-distribution and out-of-distribution performance are taken into account. The objective function maximized during search is the sum of the ID accuracy and the mean OoD accuracy, computed across RedImageNet-Sketch, RedStylized-ImageNet and two severity levels of RedImageNet-C. Although

our benchmark provides per-corruption metrics for all RedImageNet-C variants, in these experiments we aggregate them by averaging across the 15 main corruption types at each severity level.

We repeat every experiment 500 times and report average results with standard deviation in the upper part of Table 4, except for gradient-based methods DARTS and SPOS, which were repeated only five times in ID mode given the need for full supernet training for each run. We observe that multi-shot methods (i.e. Random Search, REA, REINFORCE and BANANAS) all achieve competitive performance in terms of ID accuracy, reaching values close to the best architectures in the search space. It must be noted that without HyViTas-Bench these search experiments would require a substantial amount of computational resources, with a simulated cost of approximately 1,500 GPU-hours per run, needed to train and evaluate up to 100 candidate architectures. Interestingly, REINFORCE does not exhibit a clear advantage over Random Search. This result is consistent with prior findings in the NAS literature [16], where policy-gradient methods often struggle to outperform random baselines under tight evaluation budgets due to noisy gradient estimates and limited opportunities for policy refinement.

In contrast, the gap in out-of-distribution (OoD) performance remains considerably wider, revealing a notable decoupling between ID accuracy and robustness (Fig. 5). While it is well established that ID accuracy is not a reliable indicator for OoD generalization, our results emphasize this discrepancy within the specific context of hybrid CNN-ViT architectures. Interestingly, despite being outperformed by multi-shot methods in terms of ID performance, one-shot gradient-based methods achieve better results in most OoD scenarios, suggesting the ability to find more robust architectures.

To our knowledge, HyViTas-Bench is the first benchmark to support controlled and reproducible evaluation of NAS strategies with respect to both ID and OoD performance in hybrid models. This makes it a valuable testbed for developing robustness-aware NAS approaches that move beyond traditional accuracy-centric evaluation.

D. ZERO-SHOT METHODS

Due to the considerable computational cost associated with traditional training-based NAS methods, recent research has increasingly focused on alternative, cost-effective approaches for estimating the potential performance of neural architectures without explicit training [7], [8], [9], [10], [11]. These zero-shot approaches evaluate intrinsic architectural properties to predict future accuracy, enabling rapid exploration of extensive search spaces at minimal computational expense. Motivated by these advantages, we integrate several established training-free proxy metrics into our benchmark, precomputing them for each architecture to facilitate efficient experimentation on hybrid CNN-ViT search spaces.

While most research efforts on training-free proxies have concentrated primarily on predicting the performance of neural networks in ID settings, considerably less attention has been devoted to their ability to predict architectures OoD generalization capability. Recent investigations, particularly [20], began addressing this limitation explicitly for ViT-based architectures. Their findings highlighted that, in the proposed ViT-based search space, common training-free metrics were outperformed by trivial architectural attributes, such as the number of parameters or floating-point operations (FLOPs), when predicting generalization performance.

We extend this line of inquiry by performing both correlation analyses and full NAS experiments within the context of hybrid CNN-ViT architectures included in our proposed benchmark HyViTas-Bench, reporting the Kendall-τ and Spearman-ρ correlations of various training-free metrics with OoD accuracy (see Table 5) and running three baseline search strategies 500 times for each metric (lower part of Table 4). We exclude BANANAS from these experiments since training-free metrics can be evaluated at negligible cost and therefore do not justify an additional surrogate layer. Consistent with the observations reported by [20], our findings confirm that trivial measures such as model size still exhibit higher correlation with accuracy on heavily corrupted datasets such as RedImageNet-C and RedStylized-ImageNet. Nevertheless, we also observe notable deviations from this trend on RedImageNet-Sketch, a dataset characterized by high-level semantic shifts rather than low-level corruptions. Specifically, activation-based proxies such as NASWOT [7] and gradient-based metrics like SNIP [68] and Synflow [64] demonstrate stronger correlations with OoD accuracy on RedImageNet-Sketch compared to simpler proxies based solely on model size. This suggests that training-free metrics capturing functional and optimization-related architectural properties can offer clear advantages under certain types of domain shift, pointing to a promising direction for future research into more context-aware and effective proxy designs for robust NAS.

It is worth noting that despite its simplicity, the number of parameters remains a strong predictor of OoD accuracy across our evaluations, surpassing more complex proxies in several settings in terms of correlation. Interestingly, however, in the NAS experiments conducted using training-free metrics as search objectives, this proxy was outperformed by NASWOT and SNIP, indicating that higher global correlation with accuracy does not necessarily translate into superior search performance.

The Jacobian-based proxy [67] shows consistently low rank correlation with hybrid CNN-ViT performance. This aligns with previous findings [20] that, although originally devised for assessing adversarial robustness in CNNs, this metric does not transfer effectively to transformer-based architectures. These observations further reinforce that the effectiveness of NAS methods and training-free metrics should not be assumed to hold across different architectural search spaces and evaluation scenarios. Our results highlight

TABLE 5. Kendall-τ and Spearman-ρ correlations between various training-free proxies and final top-1 accuracy on in-distribution (RedImageNet val) and out-of-distribution settings: RedImageNet-C (all corruption levels), RedImageNet-Sketch and RedStylized-ImageNet. While NASWOT shows the highest correlation with ID performance, it deteriorates under severe corruptions (RedImageNet-C 5) but remains effective on domain-shifted data like RedImageNet-Sketch.

Proxy metric	ID		RedIN-C 1		RedIN-C 2		RedIN-C 3		RedIN-C 4		RedIN-C 5		RedIN-Sketch		RedStylized-IN	
	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ
NASWOT [7]	**0.65**	**0.83**	0.41	0.56	0.33	0.46	0.27	0.37	0.15	0.20	0.01	-0.01	**0.52**	0.70	0.18	0.26
Synflow [64]	0.44	0.62	0.41	0.57	0.36	0.51	0.32	0.45	0.24	0.34	0.14	0.20	0.33	0.49	0.21	0.30
LogSynflow [65]	0.50	0.69	0.47	0.65	0.42	0.58	0.38	0.52	0.29	0.40	0.18	0.24	0.38	0.55	0.25	0.37
DSS [66]	0.44	0.63	0.37	0.53	0.33	0.46	0.28	0.40	0.20	0.27	0.09	0.11	0.38	0.55	0.21	0.31
Jacobian [67]	-0.01	-0.01	0.06	0.09	0.09	0.13	0.09	0.14	0.10	0.15	0.11	0.16	0.02	0.03	0.12	0.19
SNIP [68]	0.54	0.73	0.43	0.60	0.38	0.53	0.34	0.46	0.24	0.31	0.11	0.12	**0.52**	**0.71**	0.25	0.36
# Parameters	0.37	0.54	**0.55**	**0.74**	**0.52**	**0.71**	**0.49**	**0.68**	**0.44**	**0.62**	**0.37**	**0.53**	0.22	0.32	**0.33**	**0.47**

the need to assess such approaches on diverse architectures and benchmarks to obtain a more reliable picture of their generalization capabilities.

V. CONCLUSION

In this work we introduced HyViTas-Bench, the first Neural Architecture Search benchmark specifically designed for hybrid CNN-ViT architectures. To the best of our knowledge, HyViTas-Bench is also the first NAS benchmark to provide models trained independently multiple times on a large-scale dataset of size comparable to ImageNet-1k, enabling statistically robust performance comparisons. The benchmark includes 6,561 unique architectures, each evaluated not only in terms of in-distribution validation accuracy, but also on reduced versions of ImageNet-Sketch, Stylized-ImageNet and the five corruption levels of ImageNet-C as out-of-distribution (OoD) datasets. To support hardware-aware NAS research, we also provide inference latency measurements across 19 heterogeneous hardware devices, covering a broad spectrum of real-world deployment scenarios.

To demonstrate the utility of HyViTas-Bench, we evaluated standard NAS baselines under both ID and OoD conditions. The baselines were tested in two distinct evaluation paradigms: (i) multi-shot, using our precomputed tabular results to simulate full model training, and (ii) zero-shot, relying on six different training-free performance proxies. Additionally, we performed one-shot experiments on the ID setting with supernet-based methods. Our results show that high ID accuracy does not necessarily translate to robust OoD performance, highlighting the importance of generalization-aware evaluation criteria. Furthermore, we confirm that in OoD scenarios, common training-free proxies are often outperformed by a simple architectural attribute such as the number of parameters, although we find this trend does not hold uniformly. In the case of ImageNet-Sketch, which involves a high-level domain shift, activation or gradient-based proxies demonstrate higher predictive power.

While our benchmark provides a comprehensive testbed for hybrid architecture search, building large-scale NAS benchmarks generally involves non-trivial computational effort, especially when multiple training runs per architecture are required. Future extensions could explore integrating surrogate accuracy predictors, low-fidelity evaluations, or task-conditioned zero-cost proxies to enable enlargement of the search space.

We hope that HyViTas-Bench will facilitate reproducible, large-scale NAS experiments and foster future research on hybrid architectures, robustness, and hardware efficiency in neural architecture design. In addition to benchmark data, we release weights for all trained models, which may serve as a valuable resource for further research into NAS, optimization dynamics and generalization under distribution shift. Benchmark data, code and instructions to access the training dataset are available at https://github.com/lr94/hyvitasbench.

APPENDIX A
HARDWARE PLATFORMS

Table 6 reports the inference latency of our models on 19 hardware platforms, spanning edge devices, consumer CPUs, and both consumer and server-grade GPUs. For each device, we include the mean, minimum, maximum, and standard deviation of the per-image inference time (batch size = 1) across the architectures in the benchmark, measured as detailed in Section III-C. Fig. 8 illustrates the distribution of latency versus accuracy across the full architecture set on a subset of platforms, with the Pareto front highlighted in red. Interestingly, high-end consumer GPUs such as the RTX 5090 and 4090 achieve lower inference latency than server-grade accelerators like the A100 and H100. This behavior is consistent with known architectural differences: consumer GPUs are typically optimized for low-latency workloads and operate at higher clock frequencies, whereas data-center GPUs prioritize throughput and scalability, often running at lower frequencies under thermal or power constraints. Additionally, the relatively compact size of our models does not saturate the parallel compute capacity of large accelerators with single-image batches, further narrowing the expected performance gap. Nevertheless, including all four hardware types provides a broader view of real-world deployment scenarios, from edge inference to large-scale datacenter environments. Since power efficiency is a primary concern for edge deployments, we additionally measured energy per inference on an NVIDIA Jetson Orin

TABLE 6. Inference time (in milliseconds) measured with batch size 1. For each device, we report the mean, minimum, maximum, and standard deviation across all the architectures in the search space.

Type	Device	Inference time (ms)			
		Mean	Std.	Min	Max
Edge	Raspberry Pi Zero 2W	236.153	43.283	171.493	1489.639
	Raspberry Pi 4	125.037	15.992	89.899	218.476
	NVIDIA Jetson Orin Nano	18.806	0.716	17.349	20.342
	NVIDIA Jetson Orin NX 8 GB	20.144	0.828	18.216	22.768
	NVIDIA Jetson AGX Orin	13.549	0.536	12.413	14.768
Consumer CPUs	Intel i7 9750H	19.987	2.895	12.411	36.059
	Intel i9 10920X	9.590	0.723	7.524	12.366
	AMD Threadripper 5995WX	9.895	0.894	8.054	12.943
Consumer GPUs	NVIDIA GTX 1070	5.990	0.122	5.879	9.119
	NVIDIA GTX 1650 Max-Q	5.858	0.318	4.998	7.566
	NVIDIA RTX 2080 Ti	4.015	0.081	3.819	5.468
	NVIDIA RTX 3080	4.302	0.156	3.950	4.975
	NVIDIA RTX 3090	4.432	0.176	4.005	4.810
	NVIDIA RTX 4090	2.298	0.067	1.961	2.531
	NVIDIA RTX 5090	1.508	0.079	1.335	1.677
Server GPUs	NVIDIA RTX A5000	4.407	0.183	3.929	4.810
	NVIDIA V100	9.342	0.887	8.201	11.081
	NVIDIA A100	4.509	0.158	4.189	4.739
	NVIDIA H100	3.034	0.155	2.697	3.313

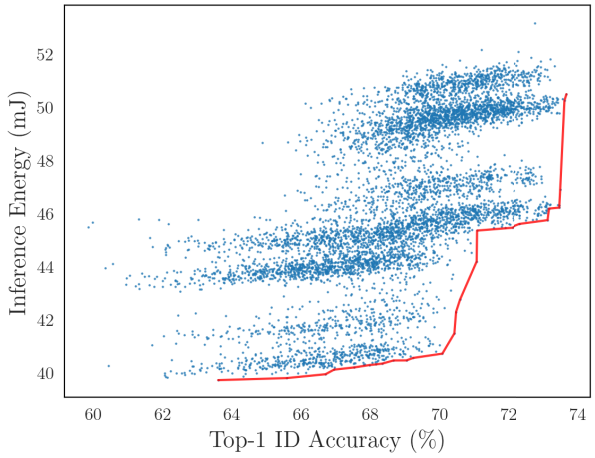

FIGURE 6. Energy consumed per inference step on the NVIDIA Jetson Orin platform for all architectures in HyViTas-Bench. The red curve denotes the Pareto front with respect to top-1 ID accuracy.

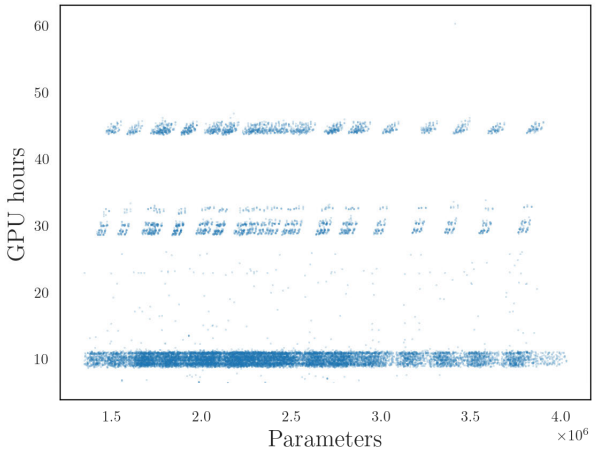

FIGURE 7. Per-model training time versus model size. Training time is not strictly correlated with model size and instead forms three distinct clusters: one group of architectures requiring approximately 10 hours for full training, another around 30 hours, and a smaller set of the slowest models requiring up to 47 hours.

Developer Kit as a representative edge GPU. Following [36] we employed the on-board INA3221 power monitor to obtain readings under the same single-image inference setting used for latency measurements, averaging power consumption for 300 inference steps for each architecture. Fig. 6 shows the distribution of the energy consumed per inference step versus model accuracy.

APPENDIX B
COMPUTATIONAL COST

Creating a faithful, low-noise tabular NAS benchmark inevitably requires substantial compute. Generating HyViTas-Bench consumed approximately 320,000 GPU-hours for training, and additional 18,000 GPU-hours for OoD evaluation, for a total of 338,000 GPU-hours. The evaluation cost is dominated by robustness assessment on RedImageNet-C alone: the 95 corruption-severity combinations account for 17,000 GPU-hours. On a cluster equipped with NVIDIA A100 GPUs, each model training run takes on average 16 GPU-hours, but the distribution exhibits substantial variance ($\sigma = 11.8$), with training times clustering into three groups that are not strictly explained by

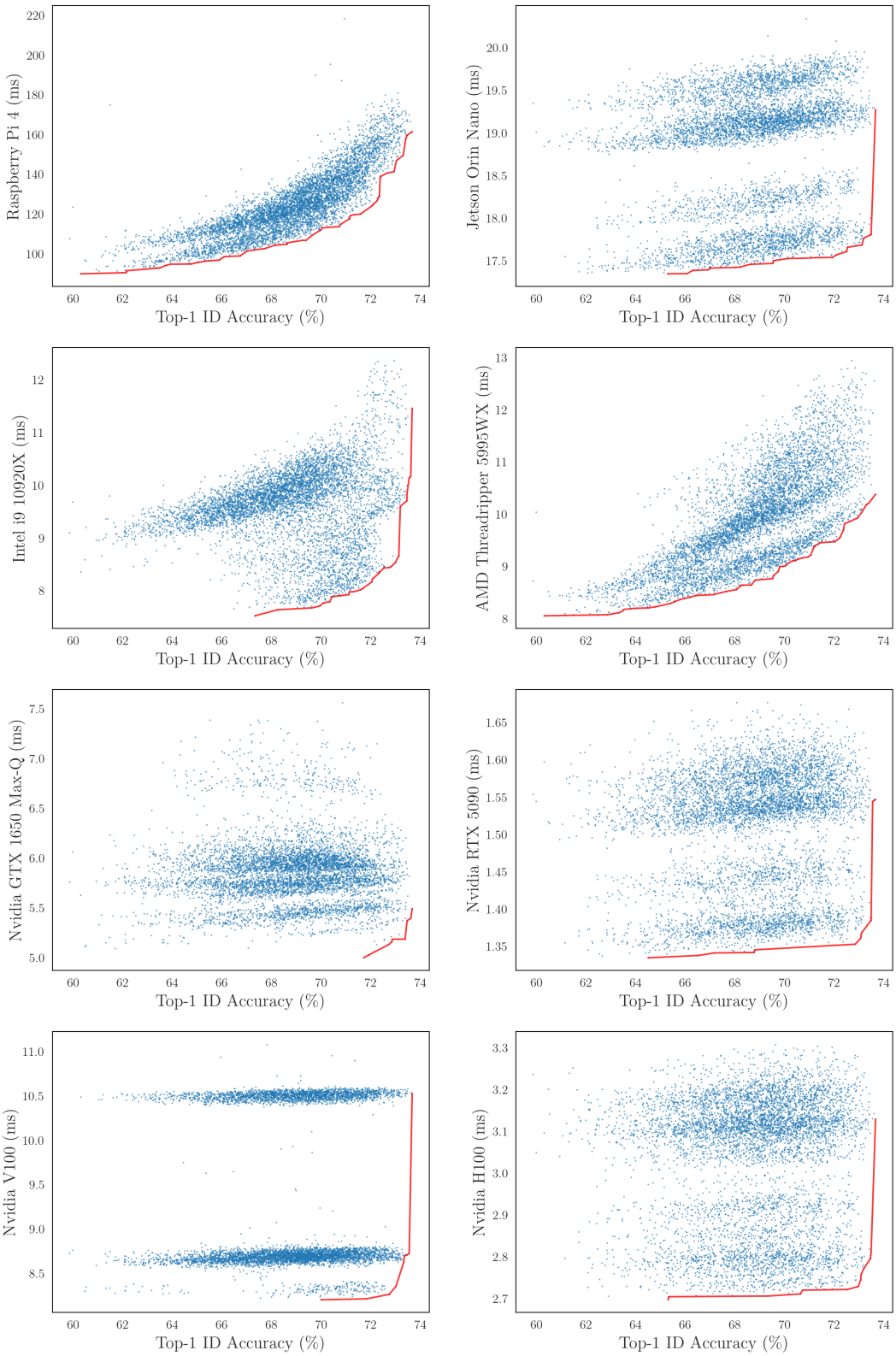

FIGURE 8. ID accuracy versus inference latency for the architectures in our benchmark evaluated on eight hardware platforms (two per type). Each point represents a unique architecture; the red curve denotes the Pareto front, illustrating optimal accuracy-latency trade-offs. A small subset of low-performing outliers has been excluded for clarity of visualization.

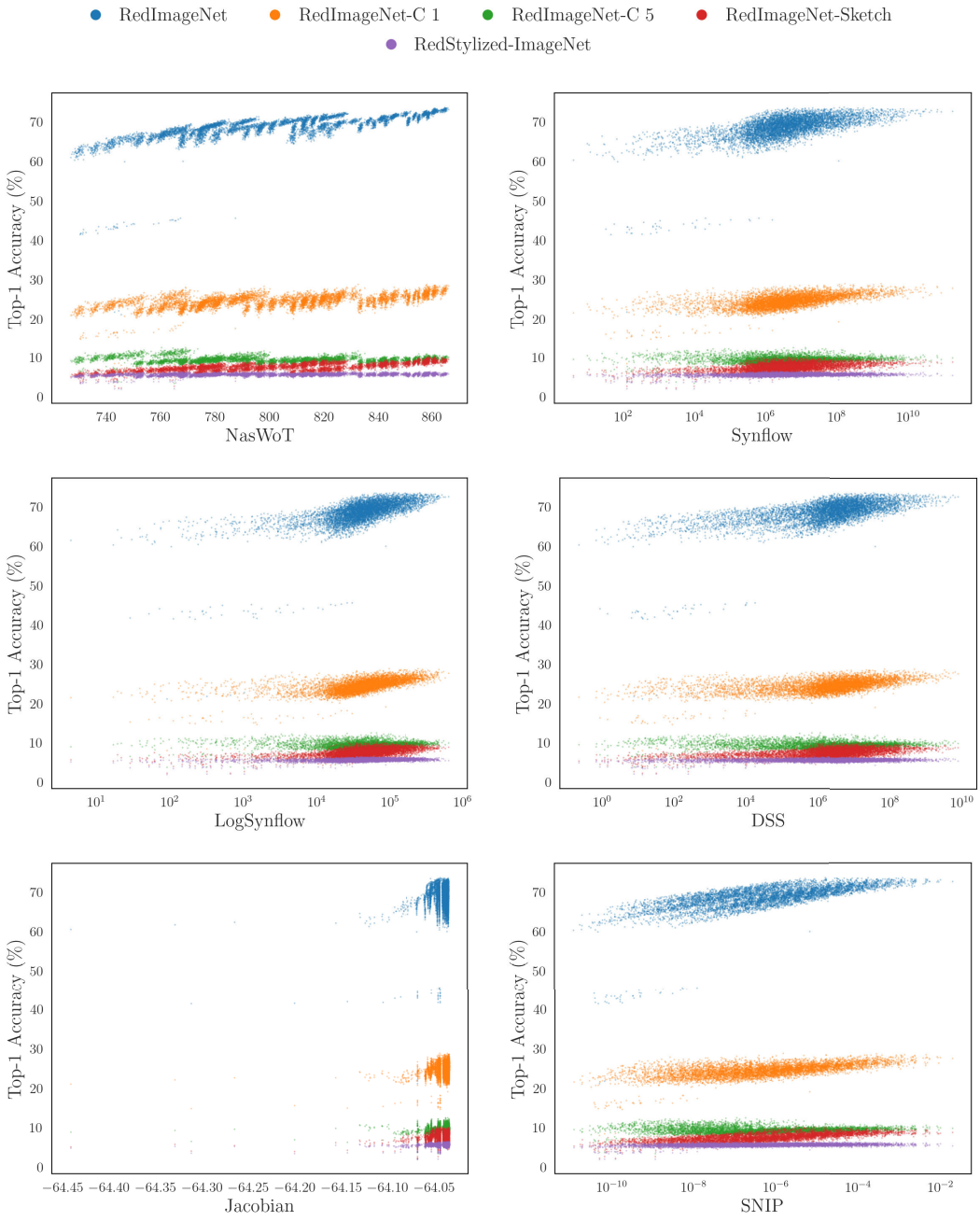

FIGURE 9. Scatter plots showing the relationship between six training-free metrics and both ID and OoD accuracy across the architectures in the benchmark. Each point corresponds to a single architecture evaluated in a specific setting, with color indicating the dataset (ID or one of the OoD benchmarks).

parameter count (Fig. 7). The vast majority of models can be trained in 10 hours, with the slowest taking up to 47 hours, excluding outliers. It must be noted that these figures reflect wall-clock time on a shared HPC system; as such, individual runs are subject to non-deterministic factors (e.g. network and node load). For this reason, per-model training time was not included as part of the benchmark data.

Although this upfront cost is high, consistently with prior tabular NAS benchmark efforts (e.g. [14], [38], [69]), it is incurred once and then amortized across the community. Once released, HyViTas-Bench enables zero-cost experimentation: researchers can fairly and reproducibly evaluate NAS algorithms on a hybrid CNN-ViT search space without retraining any model and compare methods under identical conditions, including OoD robustness.

APPENDIX C
TRAINING-FREE METRICS VISUALIZATION

Fig. 9 provides a visual overview of the relationship between the training-free metrics and the final ID and OoD accuracies

of the architectures in the benchmark. Each subplot shows a scatter distribution, where each point corresponds to a single architecture evaluated on a specific ID or OoD dataset, depending on the color. The plots complement the correlation analysis presented in Table 5 and are included here for completeness.

ACKNOWLEDGMENT

The authors acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support. This manuscript reflects only the authors' views and opinions, neither European Union nor European Commission can be considered responsible for them.

REFERENCES

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2025, pp. 5998–6008.

[3] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[4] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[5] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[6] X. Wang and W. Zhu, "Advances in neural architecture search," *Nat. Sci. Rev.*, vol. 11, no. 8, p. 282, Jul. 2024.

[7] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7588–7598.

[8] M. S. Abdelfattah, A. Mehrotra, Ł. Dudziak, and N. D. Lane, "Zero-cost proxies for lightweight NAS," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[9] N. Cavagnero, L. Robbiano, F. Pistilli, B. Caputo, and G. Averta, "Entropic score metric: Decoupling topology and size in training-free NAS," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 1451–1460.

[10] S. Liu, H. Zhang, and Y. Jin, "A survey on computationally efficient neural architecture search," *J. Autom. Intell.*, vol. 1, no. 1, 2022, Art. no. 100002.

[11] G. Li, D. Hoang, K. Bhardwaj, M. Lin, Z. Wang, and R. Marculescu, "Zero-shot neural architecture search: Challenges, solutions, and opportunities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 7618–7635, Dec. 2024.

[12] A. Yang, P. M. Esperança, and F. M. Carlucci, "NAS evaluation is frustratingly hard," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[13] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, "Neural architecture search benchmarks: Insights and survey," *IEEE Access*, vol. 11, pp. 25217–25236, 2023.

[14] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, "NAS-bench-101: Towards reproducible neural architecture search," in *Proc. 36th Int. Conf. Mach. Learn.*, California, Jun. 2019, pp. 7105–7114.

[15] X. Dong and Y. Yang, "NAS-Bench-201: Extending the scope of reproducible neural architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[16] X. Dong, L. Liu, K. Musial, and B. Gabrys, "NATS-bench: Benchmarking NAS algorithms for architecture topology and size," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3634–3646, Jul. 2022.

[17] R. Tu, N. J. Roberts, M. Khodak, J.-H. Shen, F. Sala, and A. Talwalkar, "NAS-bench-360: Benchmarking neural architecture search on diverse tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12380–12394.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[19] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11976–11986.

[20] N.-M. Cheung, S. Ebrahimkhani, S.-T. Ho, and T. Van Vo, "Vision transformer neural architecture search for out-of-distribution generalization: Benchmark and insights," in *Proc. Adv. Neural Inf. Process. Syst. 37*, 2024, pp. 84197–84245.

[21] M. Oquab et al., "Dinov2: Learning robust visual features without supervision," 2023, *arXiv:2304.07193*.

[22] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 1290–1299.

[23] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. K. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "SAM 2: Segment anything in images and videos," in *Proc. Int. Conf. Learn. Represent.*, 2024.

[24] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of ImageNet as an alternative to the CIFAR datasets," 2017, *arXiv:1707.08819*.

[25] H. Wang, S. Ge, E. P. Xing, and Z. C. Lipton, "Learning robust global representations by penalizing local predictive power," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 10506–10518.

[26] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[27] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[28] A. Khan, Z. Rauf, A. Sohail, A. R. Khan, H. Asif, A. Asif, and U. Farooq, "A survey of the vision transformers and their CNN-transformer based variants," *Artif. Intell. Rev.*, vol. 56, no. S3, pp. 2917–2970, Dec. 2023.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[30] S. B. Vijayakumar, K. T. Chitty-Venkata, K. Arya, and A. K. Somani, "ConVision benchmark: A contemporary framework to benchmark CNN and ViT models," *AI*, vol. 5, no. 3, pp. 1132–1171, Jul. 2024.

[31] Y. Haruna, S. Qin, A. H. A. Chukkol, A. A. Yusuf, I. Bello, and A. Lawan, "Exploring the synergies of hybrid convolutional neural network and vision transformer architectures for computer vision: A survey," *Eng. Appl. Artif. Intell.*, vol. 144, Mar. 2025, Art. no. 110057.

[32] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 3965–3977.

[33] Y. Li, K. Zhang, J. Cao, R. Timofte, M. Magno, L. Benini, and L. Van Gool, "LocalViT: Bringing locality to vision transformers," 2021, *arXiv:2104.05707*.

[34] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5260–5269.

[35] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[36] C. Li, Z. Yu, Y. Fu, Y. Zhang, Y. Zhao, H. You, Q. Yu, Y. Wang, and Y. C. Lin, "HW-NAS-bench: Hardware-aware neural architecture search benchmark," 2021, *arXiv:2103.10584*.

[37] Ł. Dudziak, T. Chau, M. S. Abdelfattah, R. Lee, H. Kim, and N. D. Lane, "BRP-NAS: Prediction-based NAS using GCNs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 10480–10490.

[38] M. Ding, Y. Huo, H. Lu, L. Yang, Z. Wang, Z. Lu, J. Wang, and P. Luo, "Learning versatile neural architectures by propagating network codes," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[39] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[40] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
[41] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[42] Y. Mehta, C. White, A. Zela, A. Krishnakumar, G. Zabergja, S. Moradian, M. Safari, K. Yu, and F. Hutter, "NAS-bench-suite: NAS evaluation is (Now) surprisingly easy," in *Proc. Int. Conf. Learn. Represent.*, 2022.
[43] X. Chu, B. Zhang, and R. Xu, "FairNAS: Rethinking evaluation fairness of weight sharing neural architecture search," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12239–12248.
[44] Y. Zhao, L. Wang, Y. Tian, R. Fonseca, and T. Guo, "Few-shot neural architecture search," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2020, pp. 12707–12718.
[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[46] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. S. Khan, "EdgeNeXt: Efficiently amalgamated CNN-transformer architecture for mobile vision applications," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 3–20.
[47] C. Wang, H. Xu, X. Zhang, L. Wang, Z. Zheng, and H. Liu, "Convolutional embedding makes hierarchical vision transformer stronger," in *Proc. ECCV*, 2022, pp. 739–756.
[48] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[50] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[51] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, France, Jul. 2015, pp. 448–456.
[53] R. Panda, M. Merler, M. S. Jaiswal, H. Wu, K. Ramakrishnan, U. Finkler, C.-F. R. Chen, M. Cho, R. Feris, D. Kung, and B. Bhattacharjee, "Nastransfer: Analyzing architecture transferability in large scale neural architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 9294–9302.
[54] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.
[55] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[56] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[57] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 702–703.
[58] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[59] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. aaai Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.
[60] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.
[61] C. White, W. Neiswanger, and Y. Savani, "BANANAS: Bayesian optimization with neural architectures for neural architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 10293–10301.
[62] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[63] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 544–560.
[64] H. Tanaka, D. Kunin, D. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6377–6389.
[65] N. Cavagnero, L. Robbiano, B. Caputo, and G. Averta, "FreeREA: Training-free evolution-based architecture search," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 1493–1502.
[66] M. Ding, X. Lian, L. Yang, P. Wang, X. Jin, Z. Lu, and P. Luo, "HR-NAS: Searching efficient high-resolution neural architectures with lightweight transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2981–2991.
[67] R. Hosseini, X. Yang, and P. Xie, "DSRNA: Differentiable search of robust neural architectures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6192–6201.
[68] N. Lee, T. Ajanthan, and P. H. S. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[69] Y. Duan, X. Chen, H. Xu, Z. Chen, X. Liang, T. Zhang, and Z. Li, "TransNAS-bench-101: Improving transferability and generalizability of cross-task neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5247–5256.

LUCA ROBBIANO received the M.Sc. degree in computer engineering from the Polytechnic University of Turin, Italy, in 2020, where he is currently pursuing the Ph.D. degree with the Visual and Multimodal Applied Learning Laboratory, where his research focuses on neural architecture search and efficient deep learning models. From 2020 to 2021, he was a Research Fellow with Italian Institute of Technology (IIT), working on deep learning methods for computer vision and robotics. His broader research interests include robustness, generalization, and high performance computing for deep learning.

FRANCESCA PISTILLI received the M.Sc. degree in electronic engineering from the Polytechnic University of Turin, in 2019, the M.Sc. degree in electrical and computer engineering from the University of Illinois Chicago, Chicago, IL, USA, in 2020, and the Ph.D. degree from the Image Processing and Learning Group (IPL), Polytechnic University of Turin, in 2023. She is currently an Assistant Professor with the Polytechnic University of Turin. Her current research interests include the intersection between computer vision and robotics.

GIUSEPPE AVERTA received the Ph.D. degree in robotics from the University of Pisa, in 2020. In 2019, he was a Visiting Student with the Eric P. and Evelyn E. Newman Laboratory, Biomechanics and Human Rehabilitation Group, MIT. He is currently an Assistant Professor of robotics and machine learning with the Polytechnic University of Turin. He is also an Italian Institute of Technology Alumnus. His current research interests include the development of a truly embodied intelligence for human–robot cooperation, with research activities in human action recognition, deep learning for egocentric vision, human-inspired design, planning, and control guidelines for autonomous, collaborative, assistive, and prosthetic robots.