Vision Transformer Neural Architecture Search for Out-of-Distribution Generalization: Benchmark and Insights

Sy-Tuyen Ho* **Tuan Van Vo*** **Somayeh Ebrahimkhani** * **Ngai-Man Cheung†**
Singapore University of Technology and Design (SUTD)
sytuyen_ho@mymail.sutd.edu.sg,
{vovan_tuan,somayeh_ebrahimkhani,ngaiman_cheung}@sutd.edu.sg

Abstract

While Vision Transformer (ViT) have achieved success across various machine learning tasks, deploying them in real-world scenarios faces a critical challenge: generalizing under Out-of-Distribution (OoD) shifts. A crucial research gap remains in understanding how to design ViT architectures – both manually and automatically – to excel in OoD generalization. **To address this gap**, we introduce OoD-ViT-NAS, the first systematic benchmark for ViT Neural Architecture Search (NAS) focused on OoD generalization. This comprehensive benchmark includes $3,000$ ViT architectures of varying model computational budgets evaluated on 8 common large-scale OoD datasets. With this comprehensive benchmark at hand, we analyze the factors that contribute to the OoD generalization of ViT architecture. Our analysis uncovers several key insights. Firstly, we show that ViT architecture designs have a considerable impact on OoD generalization. Secondly, we observe that In-Distribution (ID) accuracy might not be a very good indicator of OoD accuracy. This underscores the risk that ViT architectures optimized for ID accuracy might not perform well under OoD shifts. Thirdly, we conduct the first study to explore NAS for ViT's OoD robustness. Specifically, we study 9 Training-free NAS for their OoD generalization performance on our benchmark. We observe that existing Training-free NAS are largely ineffective in predicting OoD accuracy despite their effectiveness at predicting ID accuracy. Moreover, simple proxies like #Param or #Flop surprisingly outperform more complex Training-free NAS in predicting ViTs OoD accuracy. Finally, we study how ViT architectural attributes impact OoD generalization. We discover that increasing embedding dimensions of a ViT architecture generally can improve the OoD generalization. We show that ViT architectures in our benchmark exhibit a wide range of OoD accuracy, with up to 11.85% for some OoD shift, prompting the importance to study ViT architecture design for OoD. We firmly believe that our OoD-ViT-NAS benchmark and our analysis can catalyze and streamline important research on understanding how ViT architecture designs influence OoD generalization. **Our OoD-NAS-ViT benchmark and code are available at https://hosytuyen.github.io/projects/OoD-ViT-NAS**

1 Introduction

Vision Transformers (ViT) [1] have recently achieved impressive results and become a major area of research in computer vision, with significant efforts towards understanding how ViT works. These efforts have led to the proposal of both manually designed architectures [1, 2, 3, 4, 5] or automated-searched architectures [6, 7, 8, 9, 10, 11, 12] to advance ViT architectures.

*Equal Contribution †Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

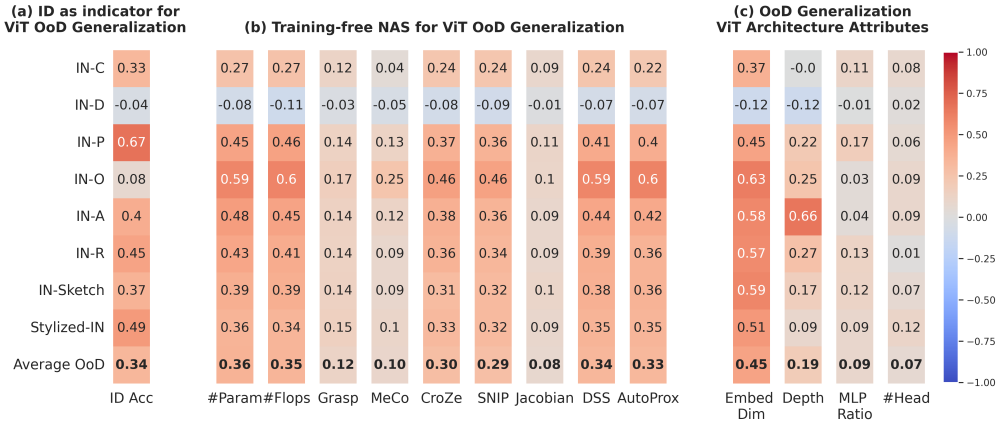

Figure 1: **We propose, OoD-ViT-NAS, the first comprehensive benchmark for NAS on OoD generalization of ViT architectures.** Then, we comprehensively investigate OoD generalization for ViT. The detailed of 8 OoD datasets in our investigation can be found in Tab. 1. In this figure, we show the Kendall τ ranking correlation between OoD accuracy of different datasets on the left and different quantities at the bottom. Our analysis uncovers several key insights. **(a) ID as an indicator for ViT OoD Generalization (Sec. 4.2)** We show that the correlation between ID accuracy and OoD accuracy is not very high. This suggests that current architectural insights based on ID accuracy might not translate well to OoD generalization. **(b) Training-free NAS for ViT OoD Generalization. (Sec. 4.3)** We conduct the first study of NAS for ViT's OoD generalization, showing that their effectiveness significantly weakens in predicting OoD accuracy. **(c) OoD Generalization ViT Architectural Attributes. (Sec. 4.4)** Our first study on the impact of ViT architectural attributes on OoD generalization shows that the embedding dimension generally has the highest correlation with OoD accuracy among ViT architectural attributes. Additional results can be found in the Appx.

Research Gap. Existing research on ViT architectures focuses on maximizing In-Distribution (ID) accuracy, while studies on the impact of ViT architectures on Out-of-Distribution (OoD) generalization are limited. Initial works [13], [14], and [15] evaluate sets of 3, 10 and 22 human-designed ViT architectures under OoD settings, respectively, and provide coarse insights into which models exhibit better OoD generalization. However, with very limited ViT architectures studied in previous works, the influence of ViT structural attributes (e.g., embedding dimension, number of heads, MLP ratio, number of layers) on OoD generalization remains unclear. Besides, in the context of ViT Neural Architecture Search (NAS), while there are various ViT NAS for ID accuracy [6, 7, 16, 8, 10, 11, 17, 18, 19, 20, 9], there is no study on ViT NAS for OoD generalization.

In this paper, we address existing research gaps by introducing OoD-ViT-NAS, the first comprehensive benchmark specifically designed for ViT's OoD generalization. Building NAS benchmarks is notoriously time-consuming and expensive due to the need to train and evaluate every candidate architecture. This challenge is particularly acute for ViT, known for its high computational demands and memory usage [21, 22, 23]. To overcome this bottleneck, we propose leveraging One-Shot NAS, specifically AutoFormer [6], a widely used ViT search space.We sample a diverse set of sub-architectures (models) to populate our benchmark. Importantly, these subnets inherit the weights from the pre-trained supernets, and *their performance has been shown to be comparable to, or even superior to, that of architectures trained alone* [20, 6] This approach enables us to efficiently acquire a large pool of ViT architectures for OoD generalization analysis. Using OoD-ViT-NAS, we conduct extensive OoD generalization analysis and gain several key insights. Additionally, with our benchmark, our work is the first to explore (training-free) NAS for ViT's OoD generalization. Our contributions are summarized below:

- We introduce OoD-ViT-NAS, the first comprehensive benchmark designed for NAS research on ViT's OOD generalization. This benchmark includes $3,000$ diverse ViT architectures sampled from the widely used ViT search space [6]. These architectures span a wide range of computational budgets. To thoroughly benchmark OoD generalization, these architectures are evaluated on the 8 most common and state-of-the-art (SOTA) OoD datasets: ImageNet-C [24], ImageNet-A [25], ImageNet-O [25], ImageNet-P [24], ImageNet-D [26], ImageNet-R [27], ImageNet-Sketch [28], and Stylized ImageNet [29] (Sec. 3)

- Our analysis demonstrates the significant influence of ViT architectural designs on OoD accuracy. This observation encourages future research to focus more on ViT architecture research for OoD generalization (Sec. 4.1)
- We show that high In-Distribution (ID) accuracy is not a very good indicator of OoD accuracy. This suggests that current architectural insights based on ID accuracy might not translate well to OoD generalization (Sec. 4.2)
- We conduct the first study to explore NAS for ViT's OoD generalization. We study 9 Training-free NAS for their OOD generalization performance on our benchmark. We observe that despite their prediction accuracy for ID, their effectiveness significantly weakens when predicting OoD accuracy. Furthermore, simple proxies such as the number of parameters (#Param) or the number of floating point operations (#Flop) surprisingly outperform more complex Training-free NAS in predicting ViTs OoD accuracy (Sec. 4.3)
- We study the impact of ViT architecture design on OoD generalization and demonstrate that careful design of ViT architectures can significantly improve OoD generalization. Specifically, increasing the embedding dimensions of a ViT architecture generally can improve its OoD generalization. (Sec. 4.4) We show that architectures with comparable ID accuracy (within an averaging range of 1.39%) exhibit a wider range of OoD accuracy, averaging 3.80% and reaching highs of 11.85%, being comparable or even outperforming state-of-the-art (SOTA) training OoD generalization method, such as those based on domain invariant representation learning [30, 31]. For example, under the same OoD setting, the SOTA method [30] shows an improvement of 1.9% OoD accuracy.

2 Related Work

Out-of-Distribution (OoD) Generalization. Addressing Out-of-distribution (OoD) generalization is a challenge, particularly in computer vision. Various approaches have been proposed to tackle this issue. A common strategy focuses on learning features that remain consistent across different domains, thereby promoting generalizability [31, 32, 33, 34, 35, 36, 37]. Other directions explore distributional robustness [38, 39], model ensembles [40, 41], test-time adaptation [42, 43], data augmentation techniques [44, 45, 46, 47, 48, 49, 50], and meta learning [51, 52] for OoD generalization. From an architectural perspective, a few attempts investigate the impact of network architecture on OoD generalization. Early work [53] shows that over-parameterized networks can hinder OoD performance due to overfitting. This raises an intriguing question: can sub-networks within such architectures achieve better OoD performance? Inspired by the Lottery Ticket Hypothesis (LTH) [54], the Functional LTH has been explored and shown that over-parameterized networks harbor sub-networks with better OoD performance. Techniques like Modular Risk Minimization [55] and Debiased Contrastive Weight Pruning [56] aim to identify these winning tickets. Another direction [57, 14] leverages Neural Architecture Search (NAS) to analyze the OoD robust architectures. However, these studies primarily focus on CNNs. While ViTs have achieved success in various visual recognition, investigations into their OoD generalization are limited. Initial works [13], [14], and [15] evaluate sets of 3, 10, and 22 human-designed ViT architectures respectively, under OoD settings. Their results provide coarse insights into which models exhibit better OoD performance. *However, the influence of ViT architectural attributes on OoD robustness remains unclear.*

Neural Architecture Search (NAS). NAS is a promising approach that has achieved remarkable success in automatically searching efficient and effective architectures for ID performance [58, 20, 59, 60]. Recently, NAS has been explored in the context of adversarial robustness for CNNs as well [61, 62, 63]. With the rise of Vision Transformers (ViTs), several NAS approaches have been applied to improve ViT architectures, including Autoformer [6], S3 [16], ViTAS [8], ElasticViT [9], DSS [10], Auto-Prox [17] and GLiT [64]. Additionally, hybrid CNN-ViT architectures like HR-NAS [18], UniNet [19], and NASViT [12] have also been explored. These efforts have shown promising results in terms of ID accuracy. *However, there has not been any work on NAS for ViT architectures specifically for OoD generalization.*

3 OoD-ViT-NAS: NAS Benchmark for ViT's OoD Generalization

In this section, we describe the construction of our OoD-ViT-NAS benchmark with details on the search spaces, datasets, evaluation metrics, and protocol. Our comprehensive benchmark includes

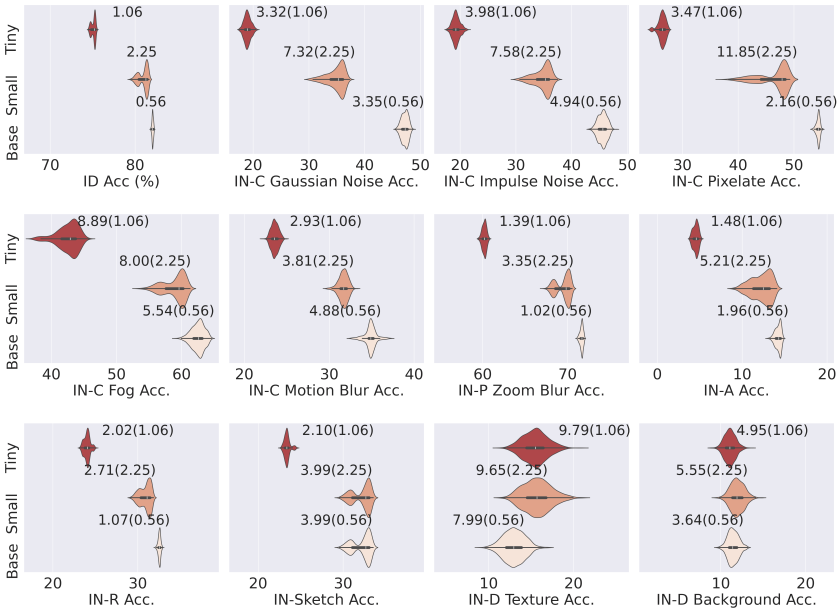

Figure 2: **Our analysis of the OoD accuracy range highlights the significant influence of ViT architectural designs on OoD accuracy. (Sec. 4.1)** The numbers within each violin plot for each sub-figure (e.g., IN-D 9.79 (1.06), 9.65 (2.25), and 7.99 (0.56)) denote the corresponding OoD (ID) accuracy range of architectures sampled from Autoformer-Tiny/Small/Base search space, respectively. See Appx. G for additional plots and results on other OoD shifts. For a fair comparison, we fix the same range for the x-axis across all sub-figures. We include the ID accuracy range in the top-left sub-figure for reference. On average, the OoD accuracy across all shifts is 3.8%/4.86%/2.74% for the search spaces in our OoD-ViT-NAS benchmark. This range is comparable to and even surpasses the current SOTA method based on domain-invariant representation learning [30], which achieved a 1.9% improvement in OoD accuracy under similar settings.

$3,000$ ViT architectures of varying sizes evaluated on 8 widely used large-scale, high-resolution, and SOTA OoD datasets. Our OoD-ViT-NAS benchmark is summarized in the Tab. 1

Search Space. We construct our benchmark based on Autoformer [6] search space. This search space is currently a widely used search space in the ViT NAS community for ID data [7, 65, 60, 10, 11, 17, 66, 67]. Autoformer search space is a large vision transformer search space including five architectural attributes that define the building block. *Embedding Dimension:* This determines the input feature representation size and is typically consistent across layers in ViT architectures. *Q-K-V Dimension:* This specifies the size of the query, key, and value vectors used in the attention mechanism. *Number of Heads:* This defines the number of parallel attention computations performed within a single attention block. *MLP Ratio:* This controls the dimensionality of the feed-forward network within each transformer block. Unlike embedding dimension, in Autoformer search space, Q-K-V Dimension, Number of Heads, and MLP Ratio can be varied across layers. *Network Depth:* This refers to the total number of transformer layers stacked in the architecture. It is important to note that Autoformer maintains a fixed ratio between the Q-K-V dimension and the number of heads in each block. This ensures that the scaling factor in the attention calculation remains constant. This helps stabilize the gradients of different heads during the training [6]. We strictly follow Autoformer search space. The details can be found in the Appx. E.1.

Dataset. Our benchmark consists of the evaluation on large-scale, high-resolution, and most SOTA OoD datasets, including ImageNet-1k [68], ImageNet-C [24], ImageNet-P [24], ImageNet-A [25], ImageNet-O [25], ImageNet-R [27], ImageNet-Sketch [28], Stylized ImageNet [29], and ImageNet-D [26]. These datasets capture a comprehensive range of OoD shifts such as common corruptions (blur, noise, digital, weather), Stable-Diffusion-based OoD shifts, and natural OoD shifts. A detailed description of these datasets can be found in the Appx. E.2.

Metrics. Following the previous OoD generalization methods [57, 30, 31, 15], we employ three metrics to construct our benchmark:

Table 1: **An overview of comprehensive setups to construct our OoD-ViT-NAS benchmark.** We utilize the widely used ViT NAS search space, Autoformer [6], which includes three different search spaces Autoformer-Tiny/Small/Base to cover a broad range of model sizes. We randomly sample $3,000$ **architectures** from these search spaces to populate our benchmark. To ensure comprehensiveness, we evaluate these architectures across 8 of the most common SOTA OoD datasets. Following prior OoD generalization works [57, 30, 31, 15], we employ three metrics for our benchmark: ID Accuracy, OoD Accuracy, and Area Under the Precision-Recall Curve (AUPR).

Search Space	Dataset	#Classes	#Images	#OoD Shifts	OoD Shift Type	Metrics
Autoformer-Tiny [6]/ Autoformer-Small [6]/ Autoformer-Base [6]	ImageNet-1K (IN-1K) [68]	1K	50K	-	-	-
	ImageNet-C (IN-C) [24]	1K	3.75M	15	Algorithmic	ID Acc/ OoD Acc
	ImageNet-P (IN-P) [24]	1K	1M	9		
	ImageNet-D (IN-D) [26]	113	4.8K	3		
	Stylized ImageNet (Stylized IN) [29]	1K	50K	1		
	ImageNet-R (IN-R) [27]	200	30K	1	Natural	
	ImageNet-Sketch (IN-Sketch) [28]	1K	50K	1		
	ImageNet-A (IN-A) [25]	200	7.5K	1		
	ImageNet-O (IN-O) [25]	200	2K	1		ID Acc/ AUPR

- *ID Classification Accuracy (ID Acc)*: This metric measures the model performance on In-Distribution (ID) data, typically the data it was trained on (e.g., ImageNet). A higher ID Acc indicates the model's ability to learn training data's distribution.
- *OoD Classification Accuracy (OoD Acc)*: This metric measures the model performance on Out-of-Distribution (OoD) data, which could differ significantly from the training data. A higher OoD Acc indicates a better generalization of the model to handle the OoD shifts.
- For the specific case of ImageNet-O, [24], we use the Area Under the Precision-Recall Curve (AUPR) metric. A higher AUPR indicates a better generalization of the model to handle the OoD detection.

Protocol. Neural Architecture Search (NAS) is notorious for its computationally expensive nature, requiring the training and evaluation of numerous candidate architectures. To address this challenge and efficiently obtain the large number of architectures needed for our benchmark (i.e., $3,000$), we make use of the One-Shot NAS approach [58, 20, 59, 12, 6, 7].

In One-shot NAS, a single supernet is first constructed. This supernet contains all possible architectures within the defined search space and is trained only once. Then, during evaluation, various architectures (i.e., subnets) can be efficiently extracted from the supernet. Importantly, these subnets inherit the weights from the pre-trained supernet, and their performance has been shown to be comparable or even superior to that of architectures trained alone [20, 6].

To support a wide range of model sizes, we leverage three supernets: Autoformer-Tiny/Small/Base, which were previously proposed for ID accuracy [6]. We randomly sample $1,000$ architectures from each supernet, resulting in a total of $3,000$ architectures in our OoD-ViT-NAS benchmark. Once obtained, these architectures are evaluated on 8 aforementioned OoD datasets.

4 Investigation on Out-of-Distribution Generalization of ViT

In this section, we provide the first comprehensive investigation of how ViT architectures affect OoD generalization using our OoD-ViT-NAS benchmark. In Sec. 4.1, we first demonstrate that ViT architectures considerably impact OoD accuracy. In Sec. 4.2, while existing works [2, 3, 6, 7, 8, 9, 12, 18] have made significant strides in improving ViT's ID accuracy, their findings could not be applicable for ViT's OoD generalization due to the not very high correlation between ViT's ID and OoD accuracy. In Sec. 4.3, we conduct the first study to explore NAS for ViT's OoD generalization. Specifically, we study 9 Training-free NAS based on their OoD generalization performance on our benchmark. Finally, in Sec. 4.4, we analyze the influence of individual ViT architectural attributes (i.e., embedding dimension, number of heads, MLP ratio, number of layers) on OoD generalization. Additional results of these analysis can be found in Appx. G, H, I, J, L

4.1 ViT architecture designs have a considerable impact on OoD generalization

In this section, we highlight that ViT architectures considerably impact OoD accuracy. This observation encourages future research to put more focus on ViT architecture research for OoD generalization.

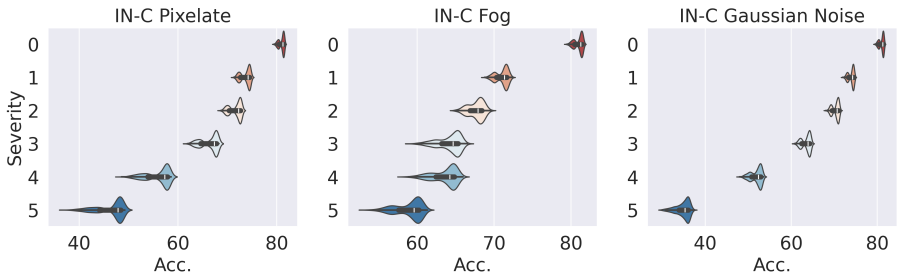

Figure 3: Visualization of OoD accuracy range across OoD shift severity. We conduct the analysis on $1,000$ architectures in Autoformer-Small search space within our OoD-NAS-ViT benchmarks. Level 0 denotes the clean examples. All corruptions can be found in Fig. G.6, in Appx. G. **We generally observe that the range of OoD accuracy widens as the severity of the OoD shift increases.**

Experimental Setups. To show how ViT architecture designs impact OoD generalization, we compute the range of OoD accuracy for each search space on an OoD dataset. This range reflects the variation in OoD performance for different architectures within a search space. For example, in Fig. 2, each sub-plot represents the range of OoD accuracy for three different search spaces in our OoD-ViT-NAS benchmark on one OoD dataset. We compute the average OoD accuracy range across all datasets as general statistics. For reference, the range of ID accuracy is also included.

Results. The results are shown in Fig. 2. Additional results can be found in the Appx.G. The average range of OoD accuracy across the three search spaces in our benchmark is 3.81%/4.86%/2.74%, which is comparable to or even outperforming state-of-the-art (SOTA) training OoD generalization method, such as those based on domain invariant representation learning [30, 31]. For example, under a similar OoD setting, the current SOTA [30] shows an improvement of 1.9% OoD accuracy. This observation highlights the significant influence of ViT architectural designs on OoD accuracy. By carefully designing ViT architecture, the OoD accuracy could improve significantly.

We further explore how the severity of the OoD shift affects the range of ViT's OoD accuracy. We conduct similar experimental setups as before, analyzing $1,000$ architectures from the Autoformer-Small search space within our OoD-ViT-NAS benchmark for $1,000$ architectures in Autoformer-Small search space within our benchmark on IN-C. The results are visualized in Fig. 3. We observe that the range of OoD accuracy widens as the severity of the OoD shift increases. This suggests that under stronger OoD shifts, the architecture design becomes even more critical for OoD generalization.

When visualizing OoD accuracy, we observe a bimodal distribution. We figure out that the embedding dimension, as the primary ViT structural attribute, influences this bimodality. For example, among architectures from the Autoformer-Small search space of our benchmark, most architectures with a lower embedding dimension (320) fall within the lower OoD accuracy mode, while those with higher dimensions (384 and 448) tend to reside in the higher accuracy mode. This observation will be further discussed in detail in Sec. 4.4.

4.2 Can ID accuracy serve as a good indication for OoD accuracy?

While existing works [2, 3, 6, 7, 8, 9, 12, 18] study the impact of ViT architectures to ID accuracy, studies on OoD accuracy are limited. To what extent can we directly apply existing findings of ViT architecture insights for ID to OoD accuracy? To answer this question, we investigate the relationship between ViT ID and ViT OoD accuracy.

Several studies [69, 70, 71, 72] investigates the relationship between ID and OoD accuracy for the CNNs model. However, there is no work on such study particularly for ViT. Utilizing our OoD-ViT-NAS benchmark, we provide the first comprehensive study on the relationship between ViT ID and OoD accuracy. Through our investigation, we find that the correlation between ViT ID and ViT OoD accuracy is not very high. This suggests that architectural insights optimized for ViT ID accuracy, as presented in previous work [2, 3, 6, 7, 8, 9, 12, 18] may not be applicable for ViT OoD generalization.

Experimental Setup. Following previous work [71], we use Kendall's τ rank correlation coefficient to compute the correlation between OoD and ID accuracy of all $1,000$ architectures from a search space on one OoD dataset. Our examination comprehensively computes the correlations across all 8 OoD datasets, 3 search spaces, and $3,000$ architectures within our OoD-ViT-NAS benchmark. We compute the average correlations across search spaces and datasets as general statistics.

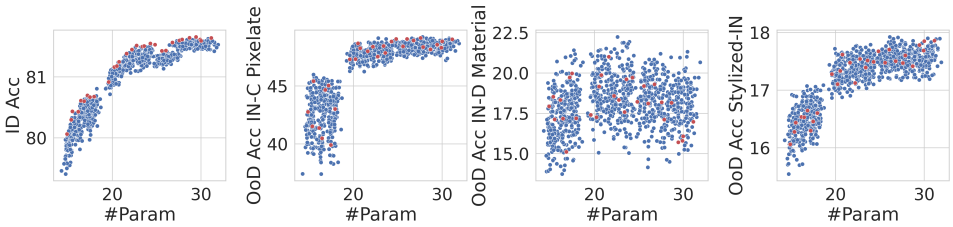

Figure 4: Analysis of OoD Generalization Performance of Pareto Architectures for ID accuracy. Blue dots ● represent architectures in the search space, while red dots ● represent the ID Pareto architectures. See Appx. I for additional results. **We find that Pareto architectures for ID accuracy generally perform sub-optimally under OoD shift.**

Besides the investigation of the correlation of various architectures, we further study the relationship between ViT ID and ViT OoD accuracy for Pareto architectures, representing the top-performing architectures for a certain model size. As shown in Fig. 4-a, the red dots ● represent Pareto architectures for ID accuracy In this study, we analyze $1,000$ architectures from the Autofomer-Small search space within our OoD-ViT-NAS benchmark. To identify Pareto architectures for ID accuracy, we divide the total parameter budget into 30 equal intervals and select the architecture with the best ID performance within each interval.

Results. The correlation results are illustrated in Fig. 1-a. The individual correlations can be found in the Appx. H. We show that the correlation between ID and OoD accuracy is generally not very high. This suggests that current architectural insights based solely on ID accuracy might not effectively translate to OoD generalization.

Among all OoD datasets, the IN-P dataset exhibits the strongest correlation with ID accuracy. This can be attributed to its weaker OoD shift compared to other datasets (see the visualization in Appx. E). As a result, the OoD examples in IN-P are not very different from ID examples, leading to a relatively high correlation between OoD and ID performance. For the remaining seven datasets with stronger OoD shifts, the correlations remain relatively low.

The results of the Pareto architectures analysis are illustrated in Fig. 4. Additional results can be found in the Appx. I. We observe that Pareto architectures for ID accuracy generally perform sub-optimally under the OoD shift. This observation further supports our previous finding that ID accuracy might not be a very good indicator of OoD accuracy.

4.3 Explore Training-free NAS for OoD Generalization

Recently, there has been a new research focus on Training-free NAS, aimed at identifying high-performing architectures without the computational expense of training each candidate. To do so, [10, 11, 17, 62, 60, 73, 74] propose zero-cost proxies to predict the performance of candidate architectures in the initialization or the first training iteration, significantly accelerating NAS. While these works focus on ID accuracy, a few attempts have been made in searching for architectures robust against adversarial attacks [62, 75]. However, there is no work to explore Training-free NAS for ViT for OoD generalization.

Experimental Setup. To address this gap, we comprehensively explore the existing 9 Training-free NAS for OoD generalization on $3,000$ ViT architectures within our OoD-ViT-NAS benchmark. Our study includes common and SOTA Training-free NAS originally proposed for CNNs for ID Acc (Grasp [74], SNIP [73], MeCo [60]), ViTs for ID Acc (DSS [10], AutoProx [17]), and CNNs for adversarial robustness (Jacobian [75], CroZe [62]). We complement this study on Training-free NAS to our OoD-ViT-NAS benchmark to equip the NAS research community with valuable tools to develop more effective Training-free NAS for OoD generalization.

Results. Our exploration provides several practical insights for designing a Training-free NAS for ViT for OoD generalization. The results of Kendall τ [76] ranking correlation between the OoD accuracy and Training-free NAS proxies on 8 common large OoD datasets are illustrated in Tab. 2. The average OoD accuracy is computed across OoD datasets and search spaces. Detailed results can be found in Fig. 1 and Appx. J.

Table 2: Comparison of Kendall τ ranking correlation between the OoD accuracies and the Training-free NAS proxies values on 8 common large OoD datasets using our OoD-ViT-NAS benchmark. **Bold** and underline stand for the best and second, respectively. We show that existing Training-free NAS's predictability in ViT OoD accuracy is limited, E.g., the very recently proposed Auto-Prox only achieves 0.3303 correlation. Furthermore, we make the first observation that simple proxies like #Param or #Flops outperform other more complex proxies in predicting both ViT OoD/ID accuracy.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1207 ± 0.1575
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.2889 ± 0.2274
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.0975 ± 0.0819
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.2951 ± 0.2223
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.0841 ± 0.1232
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.3421 ± 0.2365
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.3303 ± 0.2384
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\underline{0.3600} \pm \underline{0.2321}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\mathbf{0.3537 \pm 0.2327}$

We observe that existing Training-free NAS are largely ineffective in predicting OoD accuracy. Even recent Training-free NAS designed for ViT (i.e., DSS [10] and AutoProx-A [17]) or Training-free NAS designed adversarial robustness [62] struggle with predicting OoD accuracy.

Surprisingly, simple zero-cost proxies such as #Param or #Flops outperform all existing, more complex proxies in predicting both OoD accuracy for ViTs. This finding poses a challenge to the Training-free NAS research community: to devise a Training-free NAS that surpasses #Params or #Flops in OoD Acc prediction for ViT.

From Fig. 1-b, we observe that all Training-free NAS methods consistently fail to predict IN-D performance. This is due to the IN-D dataset's unique generation process using Stable Diffusion, which creates images labelled with object names and varying nuisances like background, texture, and material variations. Only the most challenging images are retained, resulting in highly difficult examples, such as distorted images and unrealistic object-background placements (see Fig. E.2). These examples degrade ViT model performance significantly [26] and cause unpredictable behaviour.

Our investigation into ID accuracy for ViTs also reveals a surprising observation. While proposals for Training-free NAS designed for ViTs (i.e., DSS [10] and AutoProx-A [17]), improve the prediction of ID accuracy compared to counterparts designed for CNNs. Our study marks the first attempt to explore simple Training-free NAS like #Param or #Flops. The ID prediction of such simple proxies surprisingly outperforms SOTA Training-free NAS designed for predicting ID accuracy for ViT.

4.4 ViT Structural Attributes on OoD Generalization: Increasing Embedding Dimension is Generally Helpful

Our OoD-ViT-NAS benchmark with $3,000$ ViT architectures covers diverse design choices in ViT structural attributes, including embedding dimension (Embed_Dim), network depth, number of heads (#Heads), and MLP ratio (MLP_Ratio), which allows for finding a wide range of ViT with different structures and complexities. Utilizing our comprehensive benchmark, we are the first to provide an analysis of the impact of these ViT structural attributes. We investigate which structural attributes in ViTs could lead to better OoD generalization. *Through our analysis, we find that increasing the embedding dimension of a ViT architecture can generally improve OoD generalization.* The additional analysis on another search space [1] further confirms our finding. The details for this additional analysis can be found in Appx. C

Experimental Setup. To verify the effectiveness of ViT architectural attributes on our OoD generalization benchmark, we present the results from two perspectives: (1) an analysis of rank correlation for our OoD-ViT-NAS benchmark and (2) a comparison of OoD accuracy across different embedding dimensions. To gain insights into the relationship between embedding dimension (Embed_Dim) and OoD performance, we created the visualizations for all architectures in our OoD-ViT-NAS benchmark. These visualizations compare the average OoD accuracies across different ViT architectures with varying embedding dimensions and depths. Examples of these visualizations are shown in Fig. 5. Additional results can be found in the App.K.

Results. In Fig. 1-c, we find that the embedding dimension generally has the highest correlation with OoD accuracy among all ViT architectural attributes. This positive correlation indicates that *Em-*

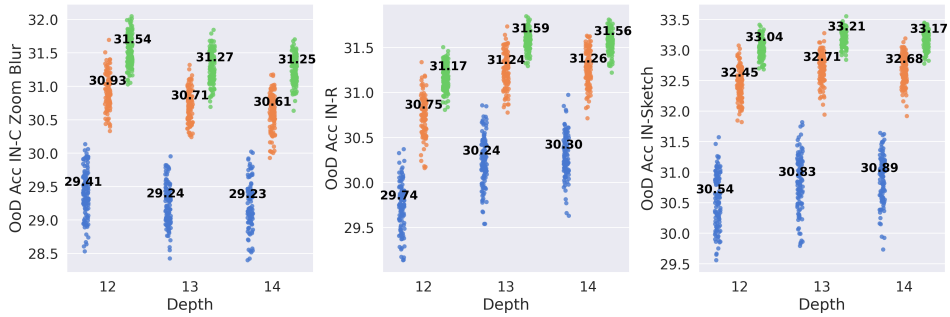

Figure 5: The effect of #Embed_Dim on robustness generalization of ViTs. The numbers denote the mean OoD accuracy across ViT architectures with specific colour-coded embedding dimensions and depths. The data points with blue ●, orange ●, and green ● colours represent ViT architectures with an embedding dimension of 320, 384, and 448, respectively. **Generally, a higher OoD accuracy is obtained when the embedding dimension of ViT architectures increases for most OoD shifts.** See Fig. M.25 and M.26 in Appx. K for additional plots and results on other OoD shifts.

bed_Dim could play a crucial role in achieving OoD generalization performance. Our comprehensive OoD-ViT-NAS benchmark sheds light on a previously unknown relationship: the potential impact of embedding dimension (Embed_Dim) on OoD generalization in ViTs. This trend holds across most OoD shifts in our benchmark (Fig. 5), suggesting that among other architectural attributes, the design choice of Embed_Dim might significantly influence a model's OoD generalization.

Our experiments yield several intriguing phenomena. Based on Fig. 1-c, we observe that *network depth has a slight impact on overall OoD generalization performance (correlation:* 0.19*).* Also, as shown in Fig. 5, for a given embedding dimension (represented by a distinct colour), we report the mean OoD accuracy, showing how the mean OoD accuracy changes among ViT architectures of varying depths, which aligns with our empirical insight. Fig. 5 shows that while increasing depth can be beneficial for improving ViT's OoD generalization in some cases, there exist shallower models that tend to perform better in terms of OoD accuracy compared to those with deeper models.

It is evident from Fig. 1-c, where both the MLP ratio (0.09) and the number of heads (0.07) exhibit very low correlation values with overall OoD performance. These findings highlight that increasing the MLP ratio and the number of heads may not substantially enhance a model's robustness to OoD data. Due to space constraints, we defer additional experiments to the Appx. L, showing that the network depth, MLP ratio, and #Heads might have non-obvious impacts on OoD generalization.

Increasing Embedding Dimension help ViT learn more high-frequency patterns, leading to improve OOD generalization. In this section, we design a frequency study to understand our finding: why increasing ViT Embedding Dimension can generally improve ViT's OOD generalization. In the literature, the models obtain higher performance on preserving High-Frequency-Component (HFC) samples tend to learn more HFC [77, 78]. By learning more HFC, the models improve OOD generalization [77, 79]. Our hypothesis is that Increasing embedding dimension helps ViTs learn more HFC resulting in improving OOD generalization. We adapt the experiment from [77] to verify our hypothesis. The details on this experimental setup can be found in the Appx. D. In a nutshell, we filter HFC by hyper-parameter radius r, where the higher the r, the lesser HFC. As shown in Fig. 6, we observe that when increasing Embedding Dimension, the performances obtained on filtering-HFC samples are improved. This observation holds true across setups varying radius r, supporting that increasing Embedding Dimension helps ViT learn more HFC. In contrast, increasing other ViT structural attributes does not help improve ViT learn more HFC.

Robust ViT architectures designed by our finding. Our study provides significant insights for guiding the design of ViT architectures. Specifically, among ViT structural attributes, increasing embedding dimension can generally improve OoD generalisation of ViT architectures. Our insight leads to a simple method which can achieve ViT architectures that can outperform well-established human-designed. We demonstrate the superiority of ViT based on our insights in Tab. 3. Scaling up ViT architecture (e.g., from ViT-B-32 to ViT-L-32) by humans typically involves compound scaling of various ViT structural attributes. However, our findings suggest that not all ViT structural attributes need to be increased to benefit OoD generalisation. Among these attributes, increasing the embedding dimension is the most crucial factor for improving OoD generalisation. By only increasing the

Table 3: Comparison of ViT architectures designed based on our insights (**Embedding Dimension**) and well-established ViT designed by humans in [3, 1].

Architecture	Embed Dim	Depth	#Head	MLP Ratio	Latency (ms)	#Param (M)	IN-R OoD Acc	Δ wrt Latency ↑	Δ wrt #Param ↑
ViT-B-32	768	12	12	4.0	105.00	87.53	41.58	-	-
Ours	840	12	12	4.0	113.32	98.64	**48.28**	**0.8054**	**0.6032**
ViT-L-32 [1]	1024	16	24	4.0	258.83	305.61	44.33	0.0179	0.0126
Swin-T	96	12	32	4.0	100.31	28.29	46.22	-	-
Ours	128	12	32	4.0	165.80	49.91	**48.28**	**0.0315**	**0.0954**
Swin-S [3]	96	24	32	4.0	184.48	49.60	47.77	0.0184	0.0725

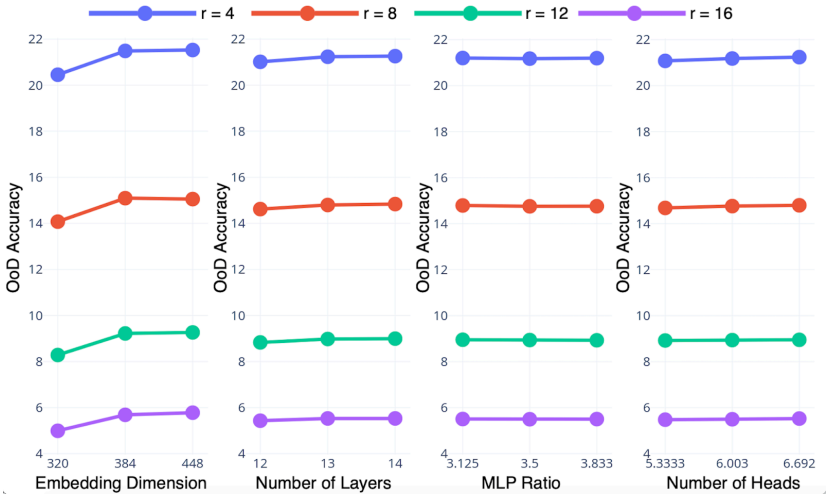

Figure 6: Following setting in [77, 80, 79], the ViTs which were trained on original ID data, are now tested on high frequency components (HFC) of OoD samples, with r as the radius for frequency filtering. The higher the OoD accuracy, the more HFC learned in the model.

embedding dimension, ours ViT architectures (e.g., Increasing embedding dimension of ViT-B-32) are significantly more efficient and outperform compound scaling architectures (e.g., ViT-L-32).

5 Conclusion

In this work, we introduce OoD-ViT-NAS, the first comprehensive benchmark for NAS on OoD generalization of ViT architectures. Using this benchmark, we conduct a comprehensive investigation on OoD generalization for ViT. Firstly, we show that ViT architecture design significantly impacts OoD accuracy. Secondly, we show that the architectural findings from existing works for ID performance could not apply to OoD generalization due to the low correlation between ID and OoD accuracy. Thirdly, we conduct the first study of NAS for ViT's OoD generalization and show that existing Training-free NAS methods struggle with OoD prediction. Surprisingly, simple proxies like #Param or #Flops outperform other complex Training-free NAS. Finally, we conduct the first study on the impact of ViT architectural attributes on OoD generalization. Our study reveals that increasing a ViT architecture's embedding dimensions can generally improve OoD generalization. We believe our benchmark OoD-ViT-NAS and comprehensive analysis will catalyze and streamline future research on understanding how ViT architecture design influences OoD generalization.

Acknowledgement. This research is supported by the National Research Foundation, Singapore under its AI Singapore Programmes (AISG Award No.: AISG2-TC-2022-007); The Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021). This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[2] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pages 459–479. Springer, 2022.

[3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[4] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022.

[5] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022.

[6] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12270–12280, 2021.

[7] Haibin Wang, Ce Ge, Hesen Chen, and Xiuyu Sun. Prenas: Preferred one-shot learning towards efficient neural architecture search. In *International Conference on Machine Learning*, pages 35642–35654. PMLR, 2023.

[8] Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vitas: Vision transformer architecture search. In *European Conference on Computer Vision*, pages 139–157. Springer, 2022.

[9] Chen Tang, Li Lyna Zhang, Huiqiang Jiang, Jiahang Xu, Ting Cao, Quanlu Zhang, Yuqing Yang, Zhi Wang, and Mao Yang. Elasticvit: Conflict-aware supernet training for deploying fast vision transformer on diverse mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5829–5840, 2023.

[10] Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Xing Sun, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10894–10903, 2022.

[11] Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search with zero-cost proxy guided evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[12] Chengyue Gong and Dilin Wang. Nasvit: Neural architecture search for efficient vision transformers with gradient conflict-aware supernet training. *ICLR Proceedings 2022*, 2022.

[13] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding Robustness of Transformers for Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10231–10241, oct 2021.

[14] Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, et al. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv preprint arXiv:2109.05211*, 2021.

[15] Salman Rahman and Wonkwon Lee. Out of distribution performance of state of art vision model. *arXiv preprint arXiv:2301.10750*, 2023.

[16] Minghao Chen, Kan Wu, Bolin Ni, Houwen Peng, Bei Liu, Jianlong Fu, Hongyang Chao, and Haibin Ling. Searching the search space of vision transformer. *Advances in Neural Information Processing Systems*, 34:8714–8726, 2021.

[17] Zimian Wei, Peijie Dong, Zheng Hui, Anggeng Li, Lujun Li, Menglong Lu, Hengyue Pan, and Dongsheng Li. Auto-prox: Training-free vision transformer architecture search via automatic proxy discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15814–15822, 2024.

[18] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, Zhiwu Lu, and Ping Luo. Hr-nas: Searching efficient high-resolution neural architectures with lightweight transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2982–2992, 2021.

[19] Jihao Liu, Xin Huang, Guanglu Song, Hongsheng Li, and Yu Liu. Uninet: Unified architecture search with convolution, transformer, and mlp. In *European Conference on Computer Vision*, pages 33–49. Springer, 2022.

[20] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 702–717. Springer, 2020.

[21] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in neural information processing systems*, 34:18590–18602, 2021.

[22] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[23] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[24] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[26] Chenshuang Zhang, Fei Pan, Junmo Kim, In So Kweon, and Chengzhi Mao. Imagenet-d: Benchmarking neural network robustness on diffusion synthetic object. *arXiv preprint arXiv:2403.18775*, 2024.

[27] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[28] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[30] Haoyue Bai, Yifei Ming, Julian Katz-Samuels, and Yixuan Li. HYPO: Hyperspherical out-of-distribution generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

[31] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[32] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[33] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International conference on machine learning*, pages 7313–7324. PMLR, 2021.

[34] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013.

[35] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.

[36] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

[37] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[38] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[39] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 561–578. Springer, 2020.

[40] Yimeng Chen, Tianyang Hu, Fengwei Zhou, Zhenguo Li, and Zhi-Ming Ma. Explore and exploit the diverse knowledge in model zoo for domain generalization. In *International Conference on Machine Learning*, pages 4623–4640. PMLR, 2023.

[41] Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 28656–28679. PMLR, 2023.

[42] Jungwuk Park, Dong-Jun Han, Soyeong Kim, and Jaekyun Moon. Test-time style shifting: Handling arbitrary styles in domain generalization. In *International Conference on Machine Learning*, pages 27114–27131. PMLR, 2023.

[43] Liang Chen, Yong Zhang, Yibing Song, Ying Shan, and Lingqiao Liu. Improved test-time adaptation for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24172–24182, 2023.

[44] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021.

[45] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18710–18719, 2022.

[46] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021.

[47] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8886–8895, 2021.

[48] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

[49] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14383–14392, 2021.

[50] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

[51] Yingjun Du, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*, 2020.

[52] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9624–9633, 2021.

[53] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[54] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[55] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021.

[56] Geon Yeong Park, Sangmin Lee, Sang Wan Lee, and Jong Chul Ye. Training debiased subnetworks with contrastive weight pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7929–7938, 2023.

[57] Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021.

[58] Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernets with alpha-divergence. In *International Conference on Machine Learning*, pages 10760–10771. PMLR, 2021.

[59] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentivenas: Improving neural architecture search via attentive sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6418–6427, 2021.

[60] Tangyu Jiang, Haodi Wang, and Rongfang Bie. Meco: Zero-shot nas with one data and single forward pass via minimum eigenvalue of correlation. *Advances in Neural Information Processing Systems*, 36, 2024.

[61] Steffen Jung, Jovita Lukasik, and Margret Keuper. Neural architecture design and robustness: A dataset. *arXiv preprint arXiv:2306.06712*, 2023.

[62] Hyeonjeong Ha, Minseon Kim, and Sung Ju Hwang. Generalizable lightweight proxy for robust nas against diverse perturbations. *Advances in Neural Information Processing Systems*, 36, 2024.

[63] Yongtao Wu, Fanghui Liu, Carl-Johann Simon-Gabriel, Grigorios Chrysos, and Volkan Cevher. Robust nas under adversarial training: benchmark, theory, and beyond. In *The Twelfth International Conference on Learning Representations*.

[64] Boyu Chen, Peixia Li, Chuming Li, Baopu Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Glit: Neural architecture search for global and local image transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2021.

[65] Mingyang Zhang, Xinyi Yu, Haodong Zhao, and Linlin Ou. Shiftnas: Improving one-shot nas via probability shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5919–5928, 2023.

[66] Shiguang Wang, Tao Xie, Jian Cheng, Xingcheng Zhang, and Haijun Liu. Mdl-nas: A joint multi-domain learning framework for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20094–20104, 2023.

[67] Yaowei Li, Ruijie Quan, Linchao Zhu, and Yi Yang. Efficient multimodal fusion via interactive prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2604–2613, 2023.

[68] Li Fei-Fei, Jia Deng, and Kai Li. Imagenet: Constructing a large-scale image database. *Journal of vision*, 9(8):1037–1037, 2009.

[69] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7721–7735. PMLR, 18–24 Jul 2021.

[70] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019.

[71] Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. Id and ood performance are sometimes inversely correlated on real-world datasets. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 71703–71722. Curran Associates, Inc., 2023.

[72] Florian Wenzel, Andrea Dittadi, Peter Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, Bernhard Schölkopf, and Francesco Locatello. Assaying Out-Of-Distribution Generalization in Transfer Learning. In S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho, and A Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 7181–7198. Curran Associates, Inc., 2022.

[73] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[74] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.

[75] Ramtin Hosseini, Xingyi Yang, and Pengtao Xie. Dsrna: Differentiable search of robust neural architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6196–6205, 2021.

[76] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[77] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[78] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[79] Paul Gavrikov and Janis Keuper. Can biases in imagenet models explain generalization? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22184–22194, 2024.

[80] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020.

[81] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[82] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

A Appendix Overview

This appendix provides supplementary information that are not included in the main paper due to space limitations.

Contents

B Limitations and Broader Impact

Given the extensive set of experiments presented in this work, the evaluation of training-free NAS proxies is significantly dependent on the initial robust benchmarks, which can be costly and resource intensive to create in the first place.

As this work studies the robustness of ViT architectures to OoD shift, we could demonstrate that carefully designed ViT architectures can significantly enhance OoD generalization. Our approach focuses on evaluating training-free NAS for ViT architectures, offering valuable insights that can complement research exploring the effects of robust architectural design. Finally, we publicly release our OoD-NAS-ViT and code base for future research.

C Analysis on Human-design ViT Search Space

Our findings on embedding dimensions are derived from analysing ViT architectures sampled through AutoFormer. To further validate these results, we also investigate the impact of ViT structural attributes on OoD generalisation within the human-designed ViT search space [1].

Table C.1: We conduct additional experiments on human design ViT to further confirm our main findings that, among ViT structural attributes, embedding dimension is the most important ViT structural attribute to OoD generalisation. We train each architecture from on IN-100 and evaluate on IN-R. **The trade-off between OoD Acc and computational metrics (i.e., Latency and #Param) is quantifies by Δ, which is the ratio of increase in OoD Acc and increase in computational metrics.** Higher Δ is better. Note that increasing #Head remains the same #Param but increases Latency in ViT setup [1].

Architecture	Configuration				Latency	#Param	OoD Acc	Δ wrt Latency $\uparrow$	Δ wrt #Param $\uparrow$
	Embed Dim	#Head	Depth	MLP Ratio	(ms)	(M)			
Vanila	768	12	12	3072	105.00	87.53	41.58	-	-
Increased Embed-Dim	840	12	12	3072	113.32	98.64	**48.28**	**0.8053**	**0.6032**
Increased #Head	768	128	12	3072	120.44	87.53	43.47	0.1224	**-**
Increased Depth	768	12	16	3072	137.36	115.88	37.80	-0.1168	-0.1333
Increased MLP-Ratio	768	12	12	3840	120.05	101.70	42.61	0.0685	0.0728

Experimental setup. We begin with the vanilla ViT-B-32 architecture [1], varying each structural attribute independently. For the altered ViT architectures, we increase these attributes to ensure that the capacities of the altered models remain comparable. Each architecture is trained on IN-100 and evaluated on IN-R.

Results. As shown in Tab. C.1, the results further confirms our findings that embedding dimension is the most important ViT structural attribute to OoD generalisation.

D Detailed description of the Frequency Analysis Setup

We adapt the experiment from [77] to verify our hypothesis. We quantify how the amount of HFC learnt in the ViTs changes if the embedding dimension of ViTs changes. Particularly, we first filter HFC from testing images of IN-R following [77, 79, 80] then evaluate the performance of 1000 ViTs in our search space on IN-R with filtering data points. Following [80] to generate HFC-preserving images, we first convert original images to FFT images. Then, we filter HFC by hyper-parameter radius r (r is set to 4, 8, 12, 16 in our experiments). In a nutshell, the higher the r, the lesser HFC.

E The details for our benchmark OoD-ViT-NAS

E.1 Detailed description of Autoformer Search Space

We strictly follow the search space in Autoformer [6] to construct our OoD-ViT-NAS. The variable factors of basic transformer blocks include the embedding dimension, Q-K-V dimension, number of heads, MLP ratio, and network depth. The detailed search spaces are illustrated in Tab. E.2

Table E.2: Detailed search space used to construct our OoD-ViT-NAS. We strictly follow the search space in Autoformer [6].

		Embedding Dim	Q-K-V Dim	MLP Ratio	Head Num	Depth Num
Supernet-Tiny	**Max**	192	192	3.5	3	12
	Min	240	256	4	4	14
	Step	24	64	0.5	1	1
Supernet-Small	**Max**	320	320	3	5	12
	Min	448	448	4	7	14
	Step	64	64	0.5	1	1
Supernet-Base	**Max**	528	512	3	8	14
	Min	624	640	4	10	16
	Step	48	64	0.5	1	1

E.2 Detailed description of OoD Datasets

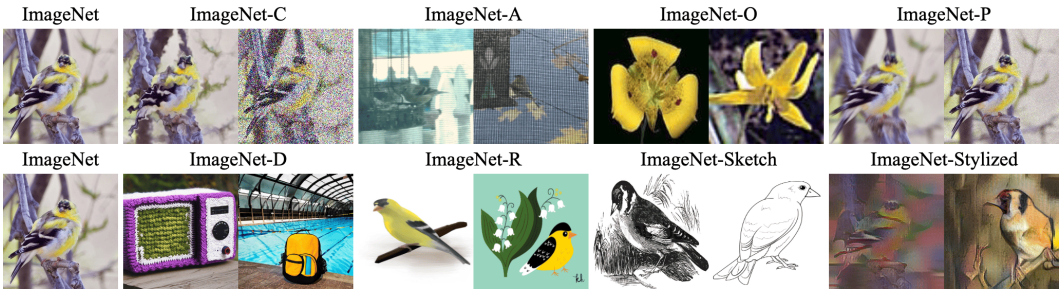

Figure E.1: Visualization of different OoD shifts across 8 datasets used to construct our OoD-ViT-NAS benchmark. The description of each dataset can be found in Sec. E.

To construct OoD-ViT-NAS benchmark, we evaluate 3,000 architectures within our benchmark on common and most SOTA OoD datasets, including:

- *ImageNet-1k* [68]: This is a large and common image dataset widely used in computer vision research. It contains over 1.3 million labeled high-resolution images belonging to 1,000 different object categories (classes). Each image is labeled with a class (e.g., "cat", "airplane", "chair").
- *ImageNet-C* [24]: This dataset builds upon the original ImageNet test set by applied algorithmically corruptions. These corruptions simulate real-world factors that can deviate data from the training set, such as blur, noise, digital, and weather effects. ImageNet-C offers a comprehensive OoD scenarios with 15 different corruption types, each with 5 severity levels, resulting in a total of 75 unique OoD setups.
- *ImageNet-P* [24]: This benchmark is constructed similarly to Imagnet-C. The difference is that ImageNet-P utilizes perturbation sequences generated from each ImageNet validation image.

- *ImageNet-A* [25]: This dataset is a real-world OoD scenarios by leveraging adversarially filtered images that are highly likely to fool current image classifiers. ImageNet-A selects a 200 classes out of 1,000 classes from ImageNet-1K so that errors among these 200 classes would be considered egregious [68].
- *ImageNet-O* [25]: Similar to ImageNet-A, this dataset includes adversarially filtered examples specifically designed to challenge OoD detectors trained on ImageNet. ImageNet-O selects a 200 classes out of 1,000 classes from ImageNet-1K.
- *ImageNet-R* [27]: This dataset is a rendition of ImageNet, containing images with manipulated textures and local image statistics.
- *ImageNet-Sketch* [28]: This dataset introduces a unique OoD challenge by providing black-and-white sketch images corresponding to the ImageNet-1K test set. This significant divergence in visual representation tests a model's ability to generalize beyond photographic data.
- *Stylized ImageNet* [29]: This dataset consists of a stylized version of ImageNet generated through techniques like AdaIN [81] style transfer, resulting in variations like greyscale, silhouettes, and edges. This dataset assesses a model's ability to handle data with different artistic interpretations.
- *ImageNet-D* [26]: This dataset is a variation of ImageNet generated through diffusion models. These models aim at creating images with rich diversity in backgrounds, textures, and materials.

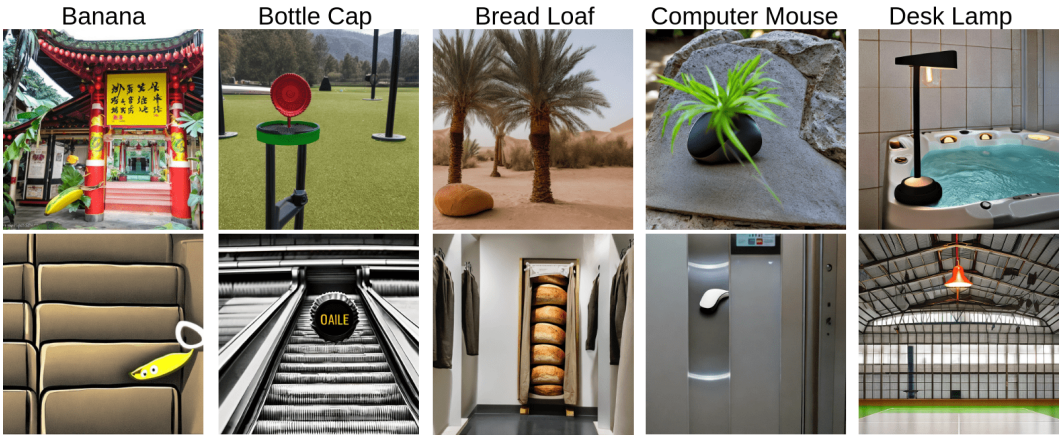

Figure E.2: Examples from ImageNet-D [26]. These examples are generated by Stable Diffusion [82] and only hard examples are kept. These examples could be distorted or unrealistic in object-background placements.

We visualize a few examples of OoD datasets in Fig. E.1. For ImageNet-C, we provide the visualization of different OoD shift severity in Fig. E.3.

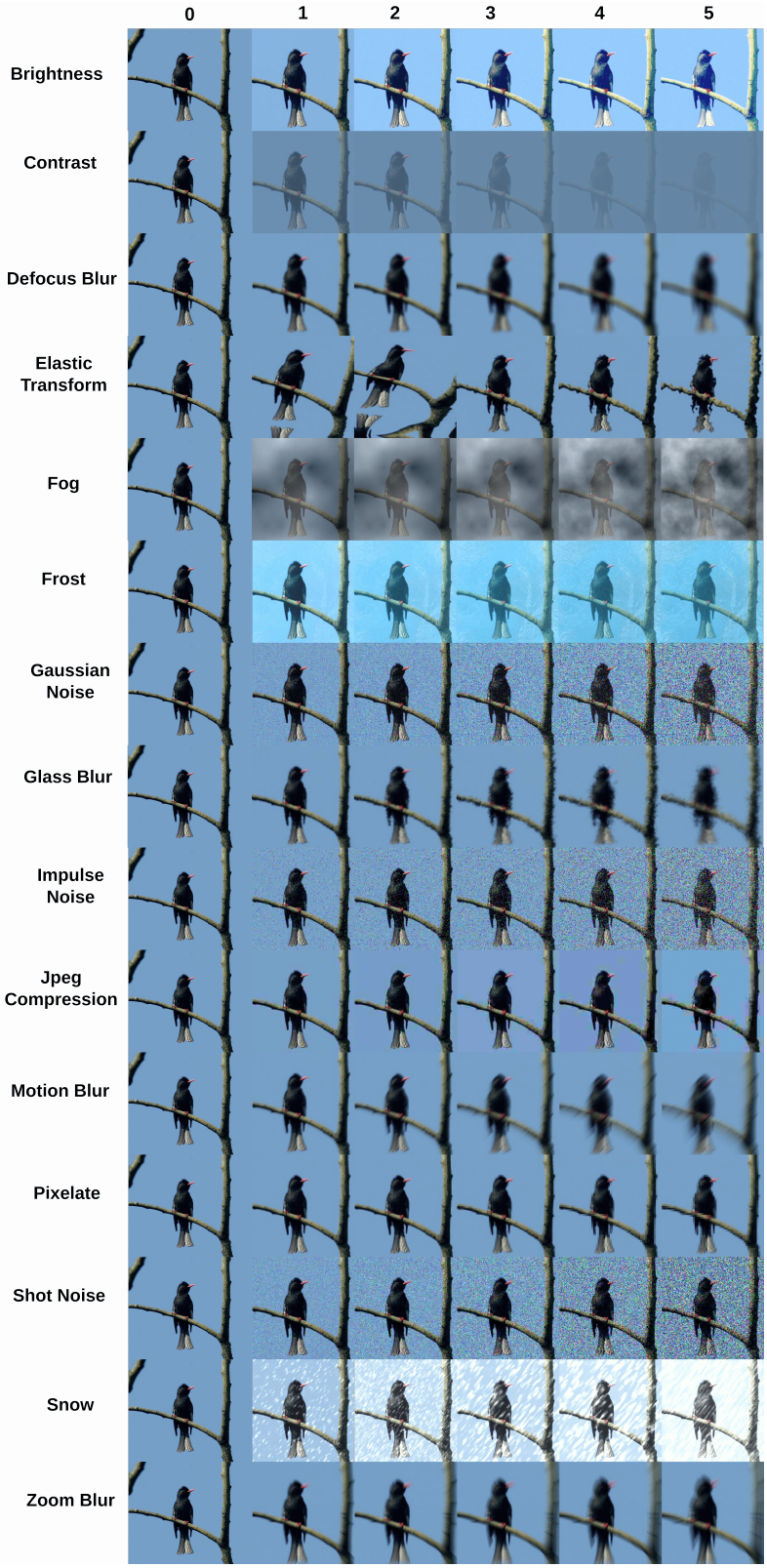

Figure E.3: Visualization of different corruptions and 5 different OoD shift severity for ImageNet-C [24]. We note that level 0 means clean examples from ImageNet [68]

F Overview of using the OoD-NAS-ViT benchmark

Our OoD-NAS-ViT benchmark consists of multiple json individual files. Each file includes the evaluation of one search space (Autoformer-Tiny/Small/Base [6]) on 8 most common and state-of-the-art (SOTA) OoD datasets: ImageNet-C [24], ImageNet-A [25], ImageNet-O [25], ImageNet-P [24], ImageNet-D [26], ImageNet-R [27], ImageNet-Sketch [28], and Stylized ImageNet [29] (Sec. 3).

We further provide a merged json file for each search space. The structure of each merged json is illustrated as in Fig. F.4. Specifically, each merged json file includes 1,000 ViT architectures denoted by the index. Each architecture consists of:

- net_setting: ViT structural attributes information:
  - layer_num: network depth
  - mlp_ration: MLP ration each layer, can be varied among layers
  - num_heads: number of attention heads each layer, can be varied among layers
  - embed_dim: embedding dimension heads each layer, fixed among layers
- params: number of parameters
- flops: number of flops
- performance: performance on different dataset:

  Imagenet: performance on IN-based datasets:
  - clean: performance on IN
  - sketch: performance on IN
  - stylized-imagenet: performance on IN
  - imagenet-R: performance on IN
  - imagenet-O: performance on IN
  - imagenet-A: performance on IN-A
  - corruption: performance on IN-C
    - Fog: performance on one out of 15 corruption in IN-C: Fog, Gaussian Noise, Fog, Snow, Elastic Transform, Jpeg Compression, Frost, Motion Blur, Brightness, Defocus Blur, Glass Blur, Impulse Noise, Shot Noise, Zoom Blur, Constrast, Pixelate
      - 1: performance at OoD shift severity level 1. There are total 5 level of OoD shift severity for each corruption in IN-C
  - corruption_P: performance on IN-P
    - Brightness: performance on one out of 10 corruption in IN-P: Brightness, Motion Blur, Rotate, Scale, Shot Noise, Snow, Tilt, Translate, Zoom Blur, Gaussian Noise
  - imagenet-D: performance on IN-D
    - background: performance on one out of 3 nuisances in IN-D: background, material, texture

With our the provided a merged json file for each search space, we can easily retrieve the OoD shift performance of various ViT architectures and their OoD performance on 8 prevelent and SOTA OoD datasets.

```
{
  "0":{
    "net_setting": {
        "layer_num": 15,
        "mlp_ratio": [3.5, 3.0, 3.5, 4.0, 3.5, 3.5, 3.5, 3.5, 3.5, 4.0, 3.0, 4.0, 3.0],
        "num_heads": [6, 5, 5, 7, 6, 7, 7, 7, 5, 6, 5, 7, 5],
        "embed_dim": [448, 448, 448, 448, 448, 448, 448, 448, 448, 448, 448, 448, 448]
    },
    "params": 64.959352,
    "flops":  5952.001594,
    "performance": {
      "Imagenet": {
        "clean": 81.51800002197265,
        "sketch": 33.04971507080507,
        "stylized-imagenet": 17.406000002441406,
        "imagenet-R": 31.52999999923706,
        "imagenet-O": 2.1998250527064194,
        "imagenet-A": 13.613333323542276,
        "imagenet-C": {
          "Fog":{
            "1": 71.72,
            "2": 68.42200000610352,
            "3": 65.80000002563476,
            "4": 65.5780000024414,
            "5": 60.92200000854492

          },
           ...
          "Pixelate": {
            "1": 74.20800000854493,
            "2": 72.34400001464844,
            "3": 67.53000000366211,
            "4": 57.36600000366211,
            "5": 47.82799999938965
          }
        },
        "imagenet-P": {
          "Brightness":  76.014
           ...
          "Gaussian Noise": 79.008
        },
        "imagenet-D": {
          "background":  11.07739130824836,
          "material":  16.189767331561075,
          "texture": 14.961653801332037
        }
      }
    }
  }
  ...
  "999":{...}
```


Figure F.4: The structure of the merged json file for our OoD-ViT-NAS benchmark for Autoformer-Small search space. The structures of the merged json files for Autoformer-Tiny/Base are similars.

G Additional results on the analysis of OoD accuracy range

In the main paper, we visualize 11 OoD accuracy ranges and ID accuracy range for reference. In this Appx. section, we provide the visualization of the remaining OoD accuracy ranges. The results are illustrated in Fig. G.5. Our observation on other OoD accuracy ranges are generally consistent with our findings in Sec. 4.1.

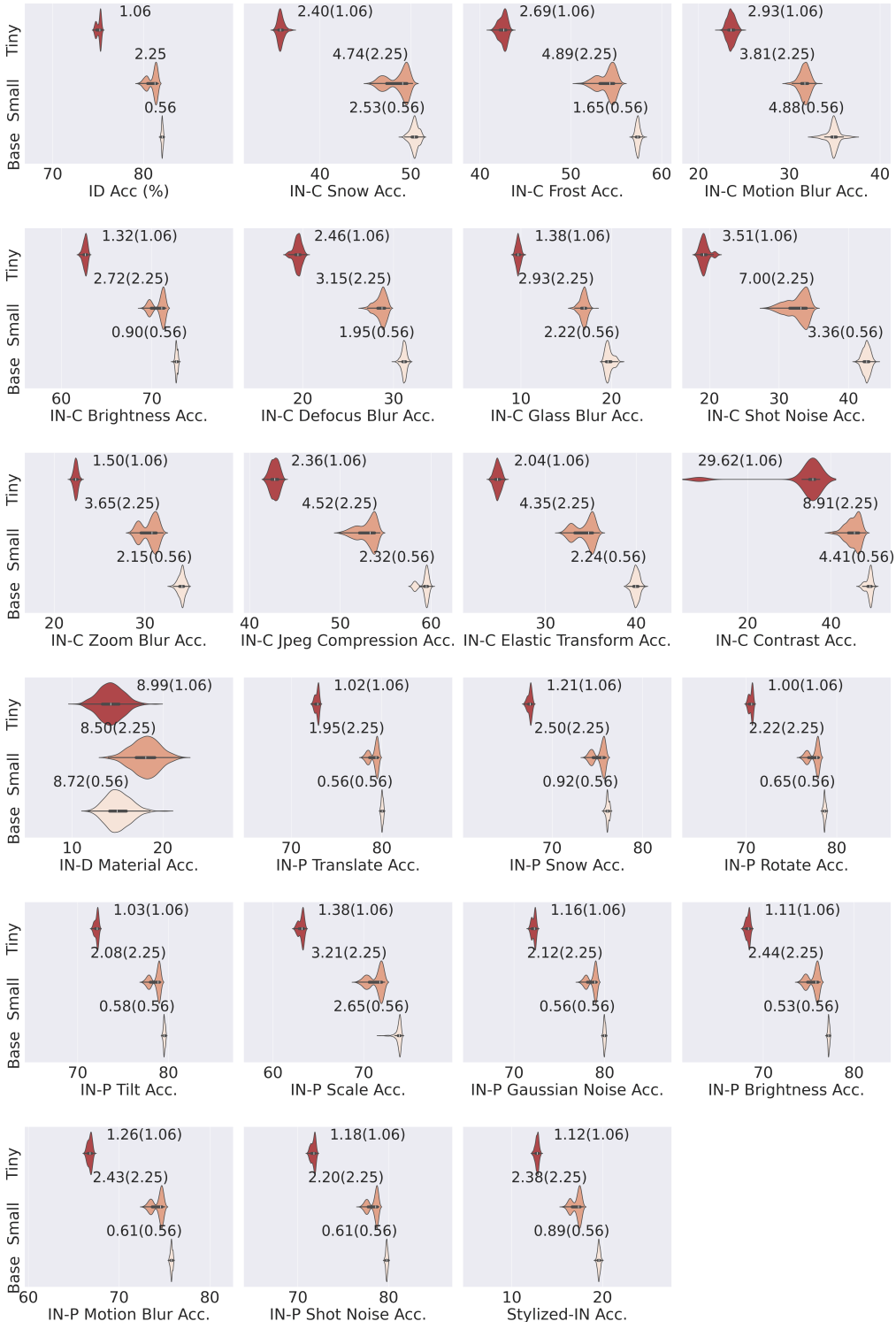

Figure G.5: As in Figure 2, our analysis on the OoD accuracy range highlights the significant influence of ViT architectural designs on OoD accuracy. The numbers within each violin plot for each sub-figures (e.g., IN-D Material 8.99 (1.06), 8.50 (2.25), and 8.72 (0.56)) denote the corresponding OoD(ID) accuracy range of architectures sampled from AutoFormer-Tiny/Small/Base search space, respectively.

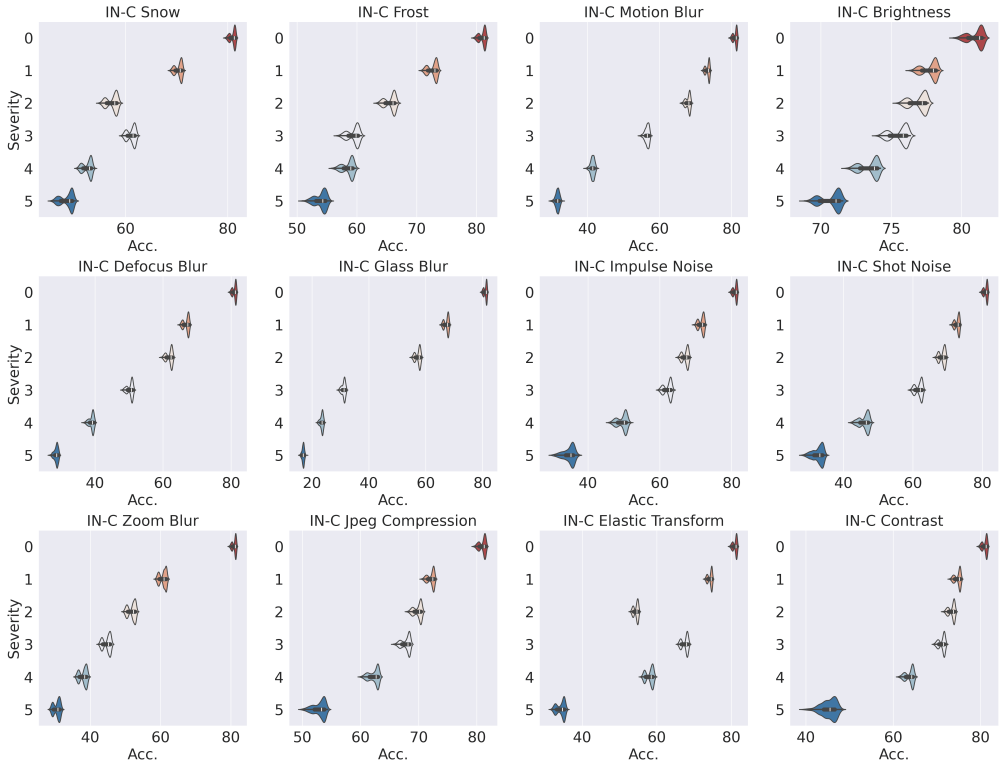

Figure G.6: Visualization of OoD accuracy range across IN-C OoD shift severity. The experiments are conducted on $1,000$ architectures in Autoformer-Small search space within our OoD-NAS-ViT benchmarks. Level 0 denotes the clean examples. We generally observe that the range of OoD accuracy widens as the severity of the OoD shift increases.

H Additional results on the analysis of the correlation between ID and OoD accuracy

In the main paper, we provide the Kendall τ correlation between ID accuracy and 8 OoD datasets. For some OoD datasets with different OoD shift types, we average the correlation with ID accuracy of different OoD shift types to obtain average correlation with ID accuracy for that OoD dataset. In this Appx. section, we provide the detailed correlations with ID accuracy of each OoD shifts in such OoD data. Specifically, we provide the detailed correlation for IN-C, IN-D, and IN-P in Fig. H.7, Fig. H.8, and Fig. H.9, respectively.

Furthermore, we provide a comprehensive correlation between OoD accuracy and ID accuracy in Fig. H.10.

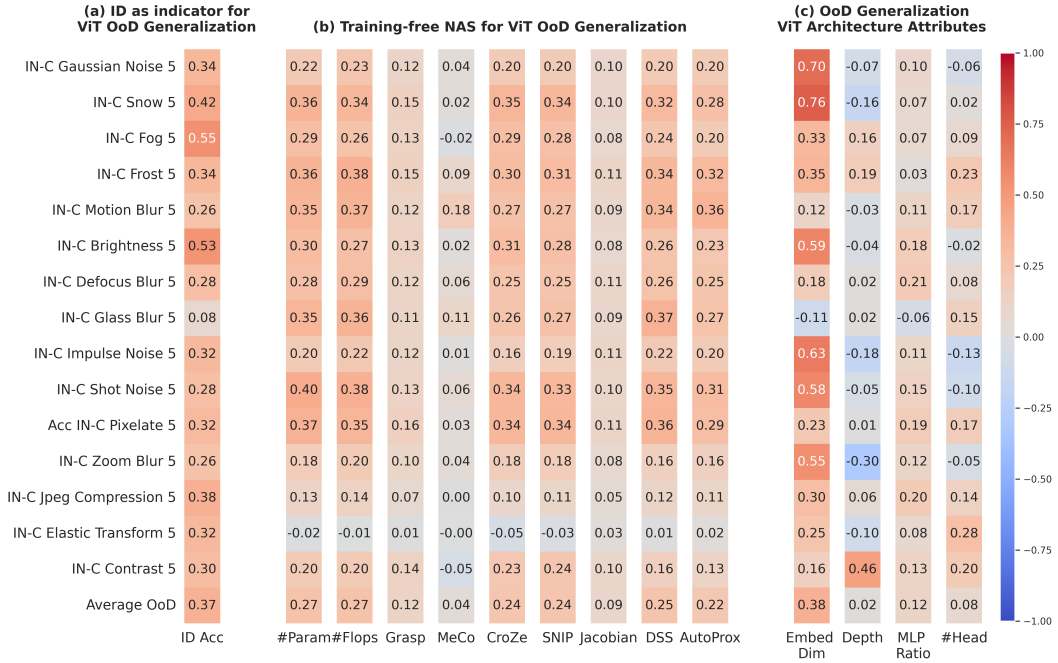

Figure H.7: Kendall τ rank correlation coefficient between ID and OoD accuracies computed on all 3000 architectures in our OoD-ViT-NAS benchmark. Measurements are computed on different corruptions of IN-C.

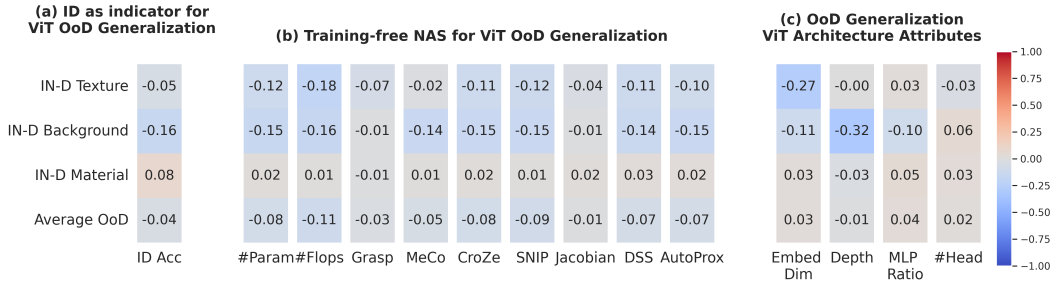

Figure H.8: Kendall τ rank correlation coefficient between ID and OoD accuracies computed on all 3000 architectures in our OoD-ViT-NAS benchmark. Measurements are computed on various IN-D OoD shifts.

I Additional results on the analysis of OoD Performance of Pareto architectures for ID

In the main paper, due to space constraints, we only provide the Pareto analysis results on a few representative OoD datasets. In this Appx. section, we provide additional results on this Pareto architecture analysis in Fig. M.24, M.19, M.20, M.21, M.22, M.23. In the following scatter plots, blue dots ● represent architectures in the search space, while red dots ● represent the ID Pareto architectures.

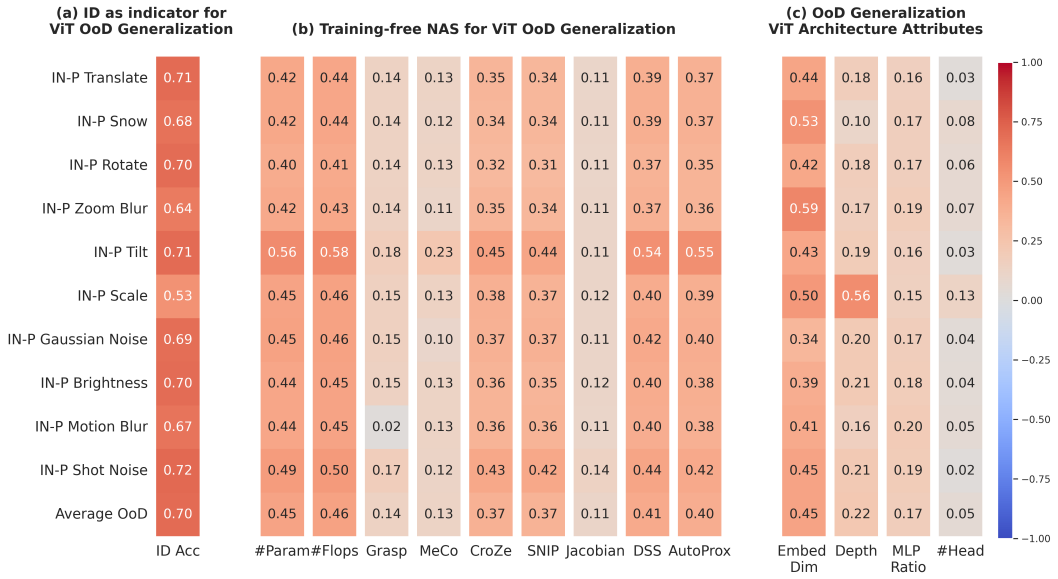

Figure H.9: Kendall τ rank correlation coefficient between ID and OoD accuracies computed on all 3000 architectures in our OoD-ViT-NAS benchmark. Measurements are computed on various IN-P OoD shifts.

J Additional Results for Benchmarking Zero-cost Proxies

In the main submission, we provides the comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values across all OoD datasets. In this section, we provides the correlation for each dataset for a detailed observation. The results can be found in Tab. J.3, J.4, J.5, J.6, J.7, J.8, J.9, J.10. Our observations on individual OoD datasets are consistent with our findings in Sec. 4.3.

Table J.3: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-C datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1179 ± 0.1713
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.2371 ± 0.2579
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.0392 ± 0.1206
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.2356 ± 0.2593
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.0894 ± 0.1369
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.2454 ± 0.2630
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.2271 ± 0.2758
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\underline{0.2651} \pm \underline{0.2546}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\mathbf{0.2656 \pm 0.2572}$

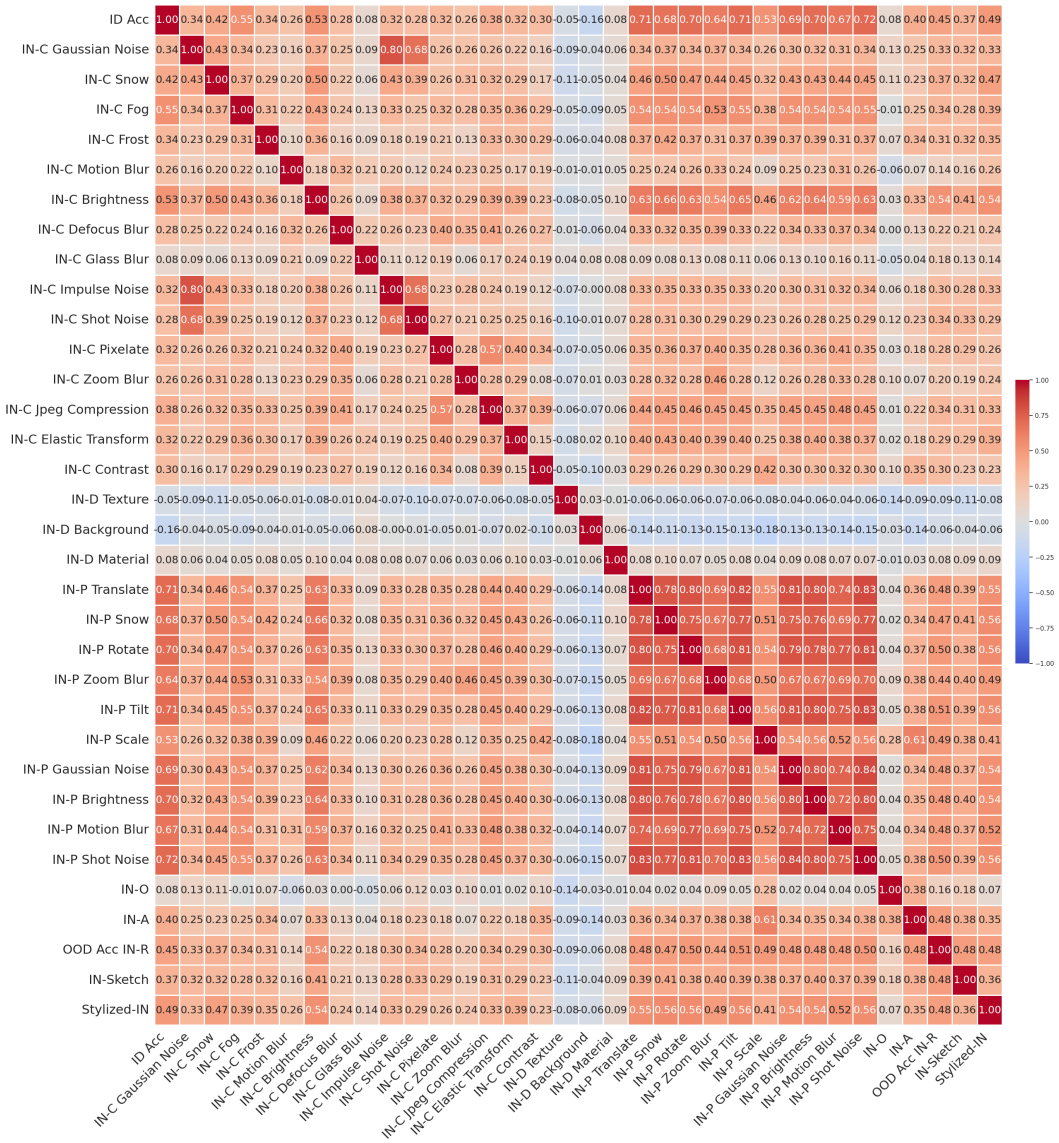

Figure H.10: Kendall rank correlation coefficient between ID and OoD performance computed on all 3000 architectures in our OoD-ViT-NAS benchmark. Measurements are computed on different OoD tasks including IN-C, IN-D, IN-P, IN-A, IN-R, IN-O, Stylized-IN, and IN-Sketch.

K Additional Figures of ViT Structural Attributes: Embedding Dimension

Table J.4: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-P datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1373 ± 0.1907
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.3652 ± 0.3306
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.1324 ± 0.2240
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.3698 ± 0.3307
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.1142 ± 0.1590
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.4128 ± 0.3621
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.3975 ± 0.3888
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\underline{0.4487} \pm \underline{0.3475}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\mathbf{0.4592 \pm 0.3547}$

Table J.5: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-D datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	-0.0306 ± 0.0307
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	$\mathbf{-0.0863 \pm 0.0887}$
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	-0.0518 ± 0.0486
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	-0.0818 ± 0.0906
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	-0.0090 ± 0.0318
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	-0.0743 ± 0.1131
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	-0.0742 ± 0.1066
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\underline{-0.0837} \pm \underline{0.1117}$
#Flops	-	-	0.4705 ± 0.3391	-0.0827 ± 0.1133

L Additional Analysis of ViT Structural Attributes on OoD Generalization

L.1 Ablation Study on the Impact of ViT Architectural Attributes to OoD Generalization

In this section, we demonstrate the effectiveness of each ViT architectural attribute on OoD accuracy from the ablation study perspective. All ablation studies are based on 1,000 ViT architectures sampled from Autoformer-Small search space in our OoD-ViT-NAS benchmark. Through our general analysis in Sec. 4.4 in the main and ablation study, we show that the embedding dimension has the highest impact among ViT architectural attributes, while network depth has a slight impact on OoD generalization.

Experimental Setups. We conduct the ablation study on the impact of ViT architectural attributes on OoD generalization. Particularly, for each ablation study of one ViT architectural attribute, we vary that attribute while keeping all other attributes fixed. Then, we compute Kendall's τ rank correlation coefficient between each attribute and different OoD shifts. While we can directly adjust the depth and embedding dimension, adjusting MLP_Ration and #Head is challenging. This is because these two attributes for each ViT arch are in the form of a list with depth elements. Each element is selected among 3 choices. This results in a huge combination. To deal with this difficulty, we first compute the means of MLP_Ration/#Head for each architecture. Then, during the ablation study, we explore a range of values for MLP ratio and the number of heads (mean #Head = 6 ± 0.05, mean MLP_Ratio =

Table J.6: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on Stylized-ImageNet datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1413 ± 0.1854
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.3175 ± 0.2658
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.0937 ± 0.1739
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.3091 ± 0.2620
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.1017 ± 0.1442
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.3800 ± 0.3294
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.3619 ± 0.3573
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\underline{0.3899} \pm \underline{0.3004}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\mathbf{0.3905 \pm 0.3094}$

Table J.7: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-Sketch datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1395 ± 0.2299
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.3434 ± 0.3696
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.0896 ± 0.1112
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.3610 ± 0.3616
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.0871 ± 0.1872
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.3885 ± 0.4021
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.3599 ± 0.3939
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\mathbf{0.4317 \pm 0.4013}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\underline{0.4060} \pm \underline{0.3917}$

3.5 ± 0.05) to capture the impact of these more nuanced architectural variations. We note that this range of values is small, allowing us to approximately fix these two attributes.

First, we assess the impact of embedding dimension on OoD generalization by fixing the configurations of all other ViT structural attributes (i.e., network depth = 13, mean #Head = 6 ± 0.05, and mean MLP_Ratio = 3.5 ± 0.05). The correlation results are shown in Fig. L.11. We observe an overall positive correlation of 0.65. This further supports our observation in Sec. 4.4 that increasing the embedding dimension generally could lead to better OoD performance across most OoD shifts for these ViT architectures.

Table J.8: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-R datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1435 ± 0.1956
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.3576 ± 0.3416
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.1224 ± 0.1820
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.3751 ± 0.3342
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.0935 ± 0.1518
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.4364 ± 0.4047
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.4164 ± 0.4027
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\mathbf{0.4773 \pm 0.4179}$
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\underline{0.4507} \pm \underline{0.4028}$

Table J.9: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-A datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1663 ± 0.1483
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.4588 ± 0.1477
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.2495 ± 0.0180
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.4642 ± 0.1415
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.1050 ± 0.1172
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	0.5930 ± 0.1021
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	$\mathbf{0.6048 \pm 0.0794}$
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	0.5923 ± 0.1288
#Flops	-	-	$\mathbf{0.4705 \pm 0.3391}$	$\underline{0.5959} \pm \underline{0.1230}$

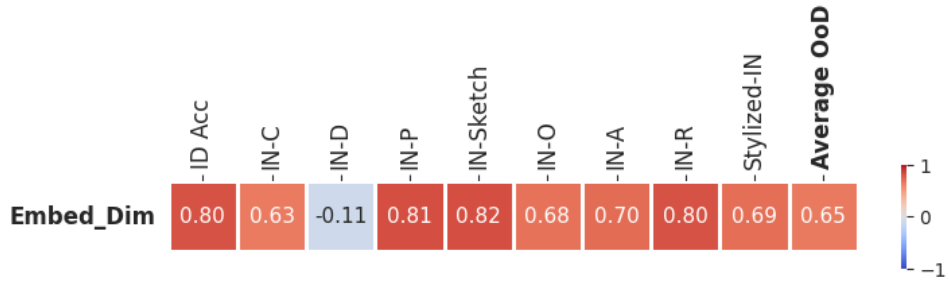

Figure L.11: Kendall's τ rank correlation coefficient between varying Embed_Dim and OoD accuracy.

To demonstrate how the number of layers (i.e., network depth) in a ViT architecture affects the model's OoD generalization, we fix other ViT architectural attributes (i.e., Embed_Dim = 384, mean #Head = 6 ± 0.05, and mean MLP_Ratio = 3.5 ± 0.05). The ranking correlation for all architectures between depth and OoD accuracy in Fig. L.12 suggests a weak correlation between the ViT network depth and OoD accuracy. This observation is consistent with our finding in Sec. 4.4 that the depth has a minimal influence on OoD generalization.

Table J.10: Comparison of Kendall τ ranking correlation between the ID/OoD accuracies and the zero-cost proxy values on ImageNet-O datasets in the Autoformer search space. **Bold** and underline stands for the best and second, respectively.

Training-free NAS	Originally Proposed For		Correlation with ID Acc	Correlation with OoD Acc
	Performance	Architecture		
Grasp [74]	ID Acc	CNNs	0.1490 ± 0.1951	0.1503 ± 0.1638
SNIP [73]	ID Acc	CNNs	0.3750 ± 0.3023	0.3178 ± 0.3219
MeCo [60]	ID Acc	CNNs	0.1440 ± 0.2371	0.1049 ± 0.2860
CroZe [62]	Adv Robustness	CNNs	0.3823 ± 0.3046	0.3277 ± 0.3307
Jacobian [75]	Adv Robustness	CNNs	0.1053 ± 0.1509	0.0911 ± 0.1242
DSS [10]	ID Acc	ViTs	0.4165 ± 0.3461	$\underline{0.3546} \pm \underline{0.3478}$
AutoProx-A [17]	ID Acc	ViTs	0.4023 ± 0.3827	0.3490 ± 0.3915
#Param	-	-	$\underline{0.4607} \pm \underline{0.3318}$	$\mathbf{0.3583} \pm \mathbf{0.3559}$
#Flops	-	-	$\mathbf{0.4705} \pm \mathbf{0.3391}$	0.3445 ± 0.3887

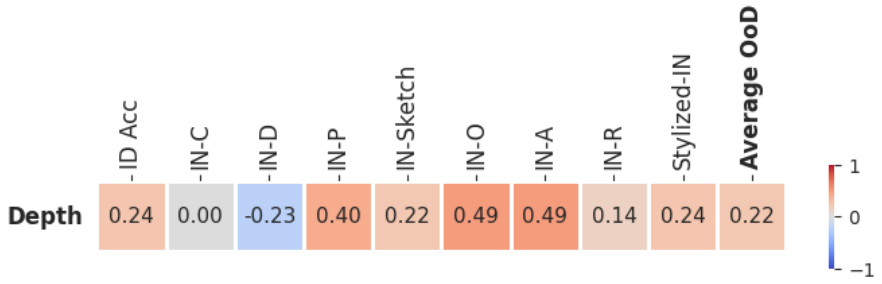

Figure L.12: Kendall's τ rank correlation coefficient between varying network depth and all OoD accuracy.

For MLP_Ratio and #Heads, we observe that the rank correlation coefficients of MLP_Ratio and #Heads in Fig. L.13 and L.14, respectively, reveals a weak correlation between the #Heads/MLP_Ratio and OoD generalization. This suggests that within the explored range, these two architectural attributes have a non-obvious impact on the model's OoD generalization. To delve deeper into this observation, Section L.2 presents a layer-wise analysis of these two architectural attributes in ViT models.

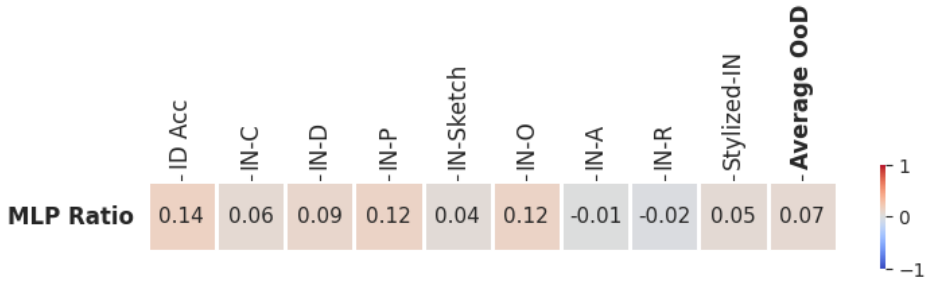

Figure L.13: Kendall's τ rank correlation coefficient between mean MLP ratio and all OoD accuracy. We fix Embed_Dim = 384, Depth = 13, and mean #Head = 6 ± 0.05.

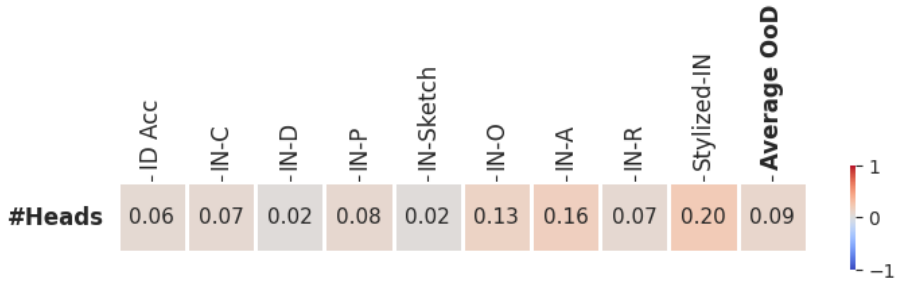

Figure L.14: Kendall's τ rank correlation coefficient between the mean number of heads and all OoD accuracy. We fix Embed_Dim = 384, Depth = 13, and mean MLP_Ratio = 3.5 ± 0.05.

L.2 Layer-Wise Analysis

The number of heads and MLP ratio vary across layers, which allows for searching more diverse architectures. In this section, we provide the layer-wise analysis of the influence of MLP_Ratio/#Heads in each layer on OoD generalization.

Experimental Setups: In Autoformer-Small search space, the MLP ratio at any given layer — referred to as the i-th layer — can be set to 3.0, 3.5, or 4. In addition to architectures in our benchmark, we create a set of 108 sampled architectures from Autoformer-Small. These architectures are fixed with three architectural design attributes: (Depth $= 12$, Embed_Dim $= 320$, #Heads for all layers $= 5$). With these parameters fixed, the total number of potential MLP_Tatio configurations is still extensive (i.e., 3^{12}), making it impractical to absolutely fix the MLP_Ration in our ablation study. To specifically assess the influence of the MLP_Ration at the i-th layer, we further fix a constant MLP_Ration across all other layers. For example, Fig. M.18 depicts the nine configurations of MLP_Ration to analyze the impact of 5-th layer.

We carry out a similar layer-wise analysis to demonstrate the effect of #Heads at a specific layer (i-th layer). In addition to architectures in our benchmark, we create a set of 108 sampled from Autoformer-Small. These architectures are fixed with three architectural design attributes: (Depth = 12, Embed_Dim =320, #MLP_Ratio for all layers =3.0). To specifically assess the influence of the #Head at the i-th layer, we further fix a constant #Head across all other layers. For example, Fig. M.18 depicts the nine configurations of #Head to analyze the impact of 3-th layer.

Results. The layer-wise analysis for MLP_Ratio are shown in Fig. L.15. Each sub-figure demonstrates the change in OoD Accuracy when varying MLP_Ratio at a particular layer. While increasing it improved OoD accuracy in some layers, it decreased it in others. This aligns with our observations in Sec. 4.4, suggesting no clear overall impact of MLP_Ratio on OoD generalization.

The layer-wise analysis for #Head is illustrated in Fig. L.16. Similar to our observation in the layer-wise analysis for MLP_Ratio, the impact of #Head is non-obvious, which is consistent with our finding in Sec. 4.4.

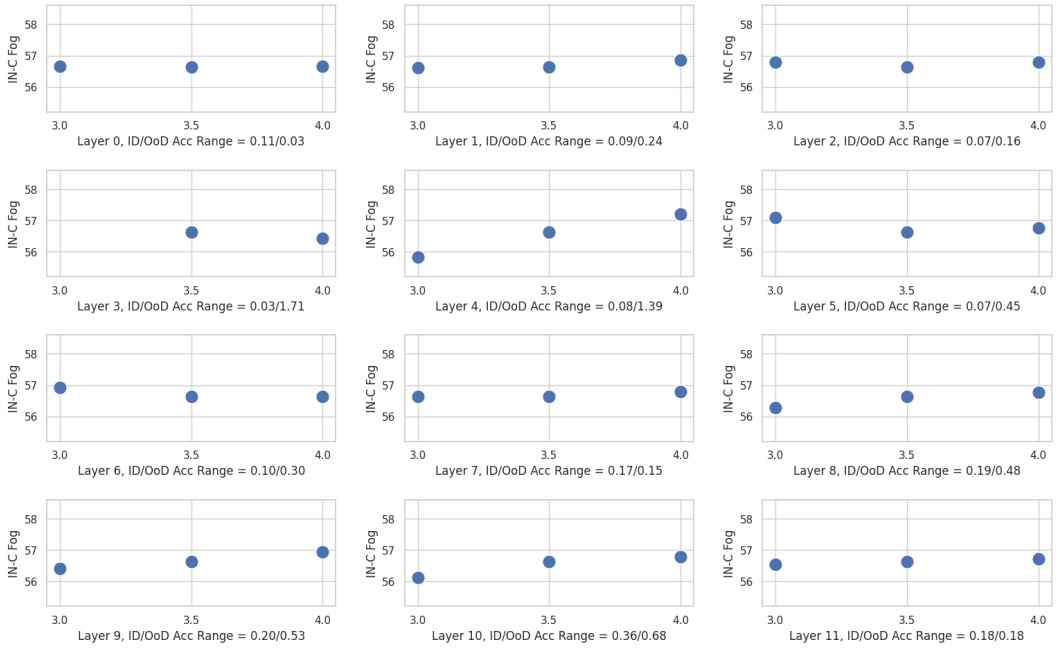

Figure L.15: A visualization on the effect of changing MLP ratio per layer to OoD accuracy in 108 architectures sampled from Autoformer-Small in the layer-wise study. To evaluate the effect of layer i-th, we fix the MLP ratio $= 4.0$ for the remaining layers. We observe that a slightly higher OoD accuracy range can be obtained by changing the MLP ratio at layers 4 and 5.

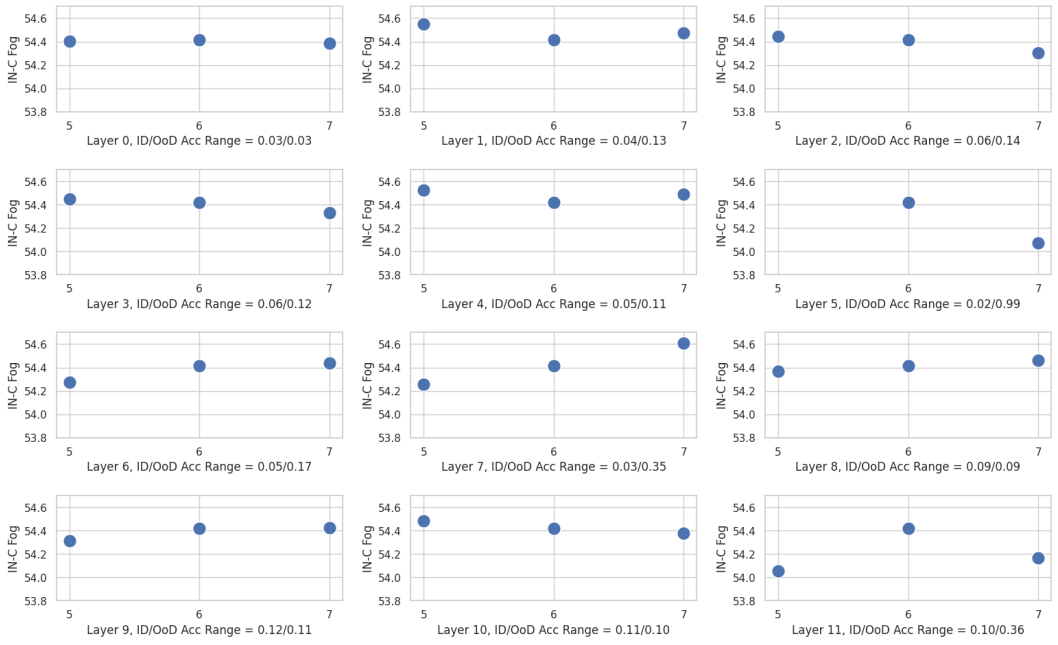

Figure L.16: A visualization on the effect of changing #Heads per layer to OoD accuracy in 108 architectures sampled from Autoformer-Small in the layer-wise study. To evaluate the effect of layer i-th, we fix the #Heads $= 6$ for the remaining layers.

M Reproducibility

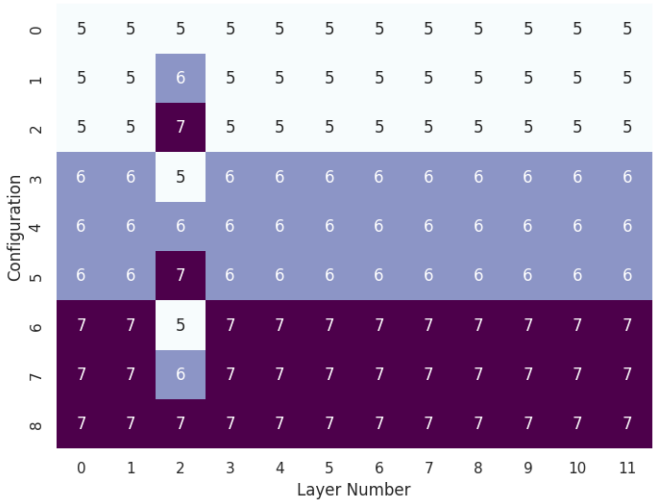

Figure M.17: The possible configurations of #Heads at 3rd layer of 108 architectures sampled from *Autoformer-Small* in layer-wise analysis.

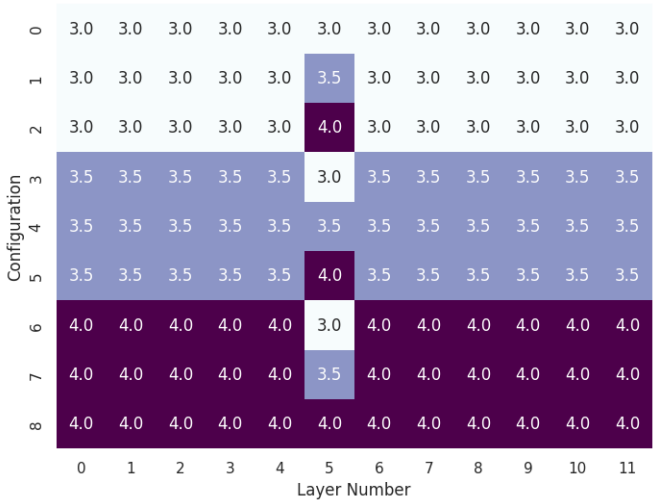

Figure M.18: The possible configurations of MLP ratio at 5-th layer of 108 architectures sampled from Autoformer-Small in the layer-wise study.

M.1 Hyper-parameters

Table M.11: Hyper-parameters for Evaluation on 8 common large-scale OoD datasets. In total, we evaluate 3,000 diverse ViT architectures in our OoD-NAS-ViT benchmark sampled from Autoformer-Tiny/Small/Base search spaces [6]. Input resolution is set to 224×224 pixels, with mean and standard deviation normalization applied using ImageNet statistics ($mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]$). Transformations follow the Standard ImageNet preprocessing, including resize and center crop.

	IN-C	**IN-P**	**IN-A**	**IN-O**	**IN-R**	**IN-Sketch**	**Stylized-IN**	**IN-D**
Batch Size	256	256	256	64	256	256	256	100
Number of workers	10	10	10	4	10	10	10	10

To ensure reproducibility, we provide a detailed description of the hyper-parameters used for evaluating 3,000 ViT architectures in our OoD-NAS-ViT benchmarks on 8 common large-scale OoD datasets: ImageNet-C [24], ImageNet-A [25] [24], ImageNet-O [25], ImageNet-P [24], ImageNet-D [26], ImageNet-R [27], ImageNet-Sketch [28], and Stylized ImageNet [29]. The evaluation for ImageNet-D, ImageNet-O and Stylized ImageNet strictly follows previous works [26, 25, 29] to ensure consistency and comparability. The details of the evaluation are shown in Table M.11. The evaluated architectures are sampled on AutoFormer-Tiny/Small/Base Search Spaces [6].

M.2 Compute Resource

All our experiments are conducted using NVIDIA RTX A6000 GPUs. We utilized 2 GPUs for each experiment. The substantial computational resources required for these evaluations underscore the complexity and scale of our work. Detailed information on the GPU-hour consumed for all experiments to construct our OoD-ViT-NAS Benchmark can be found in Table M.12. In total, the experiments demanded a significant investment of approximately 3900 GPU-hours, reflecting the extensive computational effort involved.

Table M.12: GPU-Hour for Computational Resources. In total, we evaluate 3,000 diverse ViT architectures in our OoD-NAS-ViT benchmark sampled from Autoformer-Tiny/Small/Base search spaces [6].

	IN-C	**IN-P**	**IN-A**	**IN-O**	**IN-R**	**IN-Sketch**	**Stylized-IN**	**IN-D**	**Total**
GPU-Hour	2958	672	12	93	28	53	37	50	**3903**

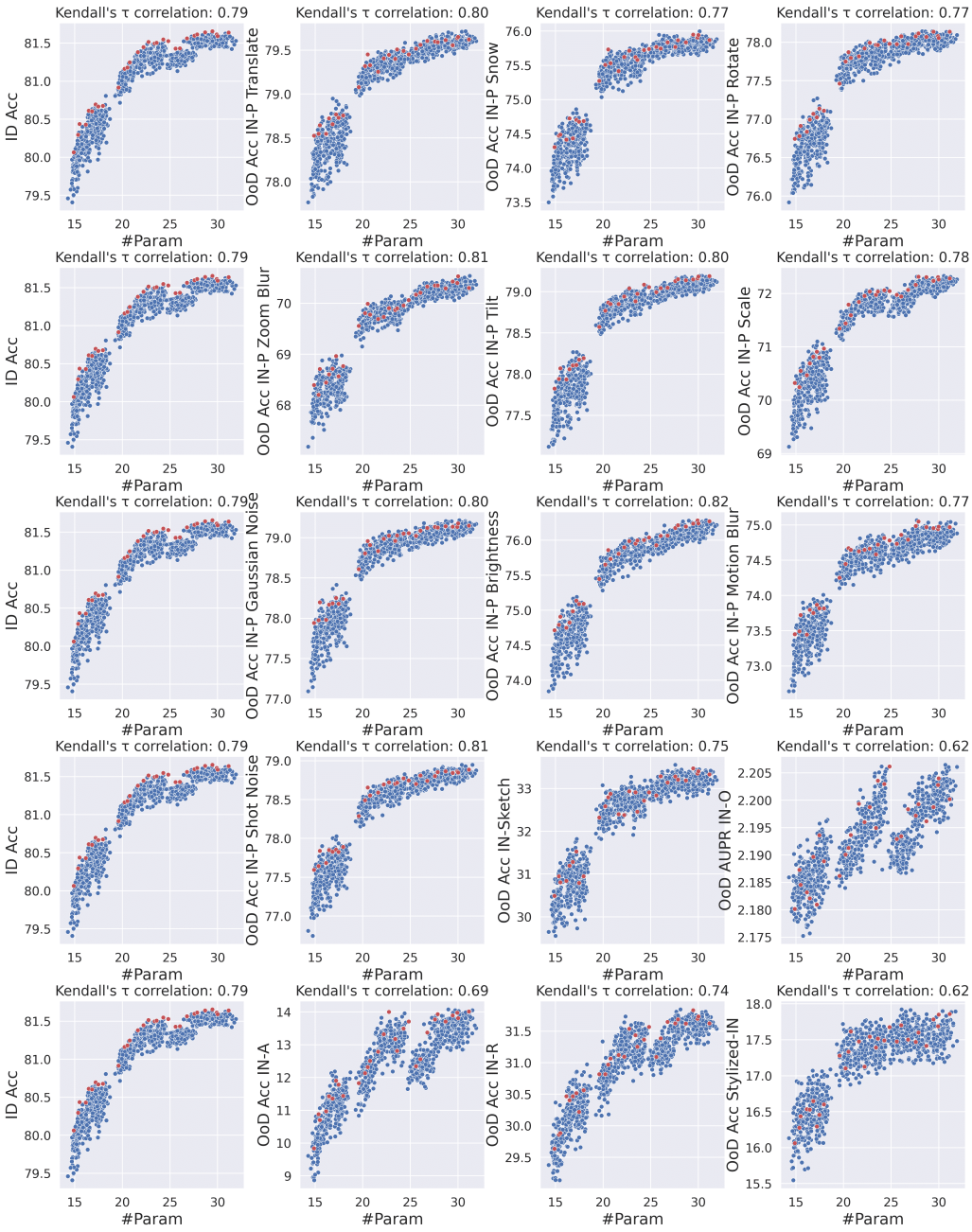

Figure M.19: As in Figure 4, we show that lower OoD accuracy can be obtained for the higher ID accuracy in the Pareto architectures of *Autoformer-Small*. The left panels show the ID accuracy, and each panel on columns 2 to 4 shows results from the OoD accuracy of IN-P, IN-A, IN-R, IN-Sketch, Stylized-IN, and AUPR of IN-O.

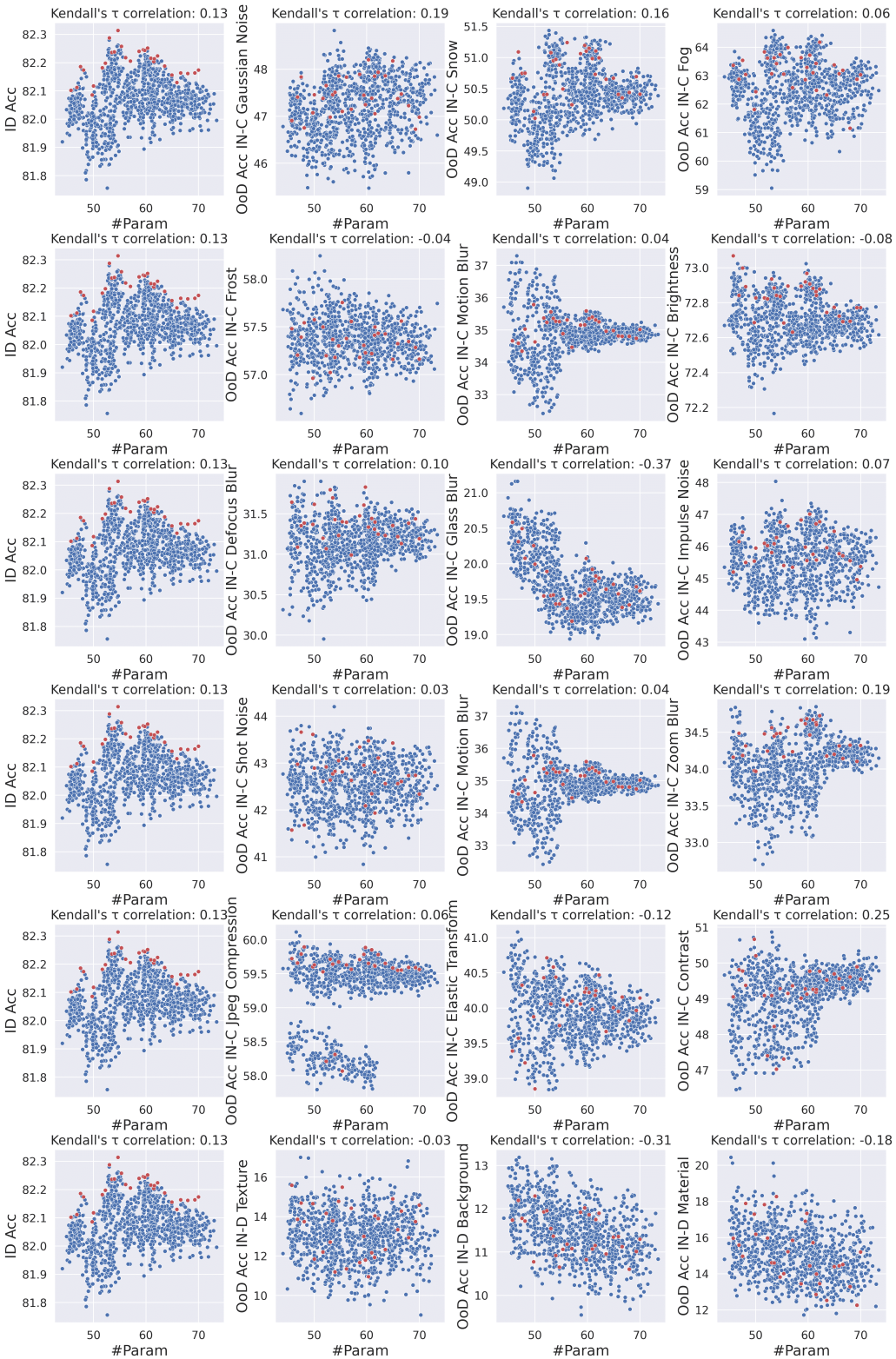

Figure M.20: Visualization of Pareto architectures in *Autoformer-Base*. The left panels show the ID accuracy, and each panel on columns 2 to 4 shows results from the OoD accuracy of IN-C common corruptions and IN-D.

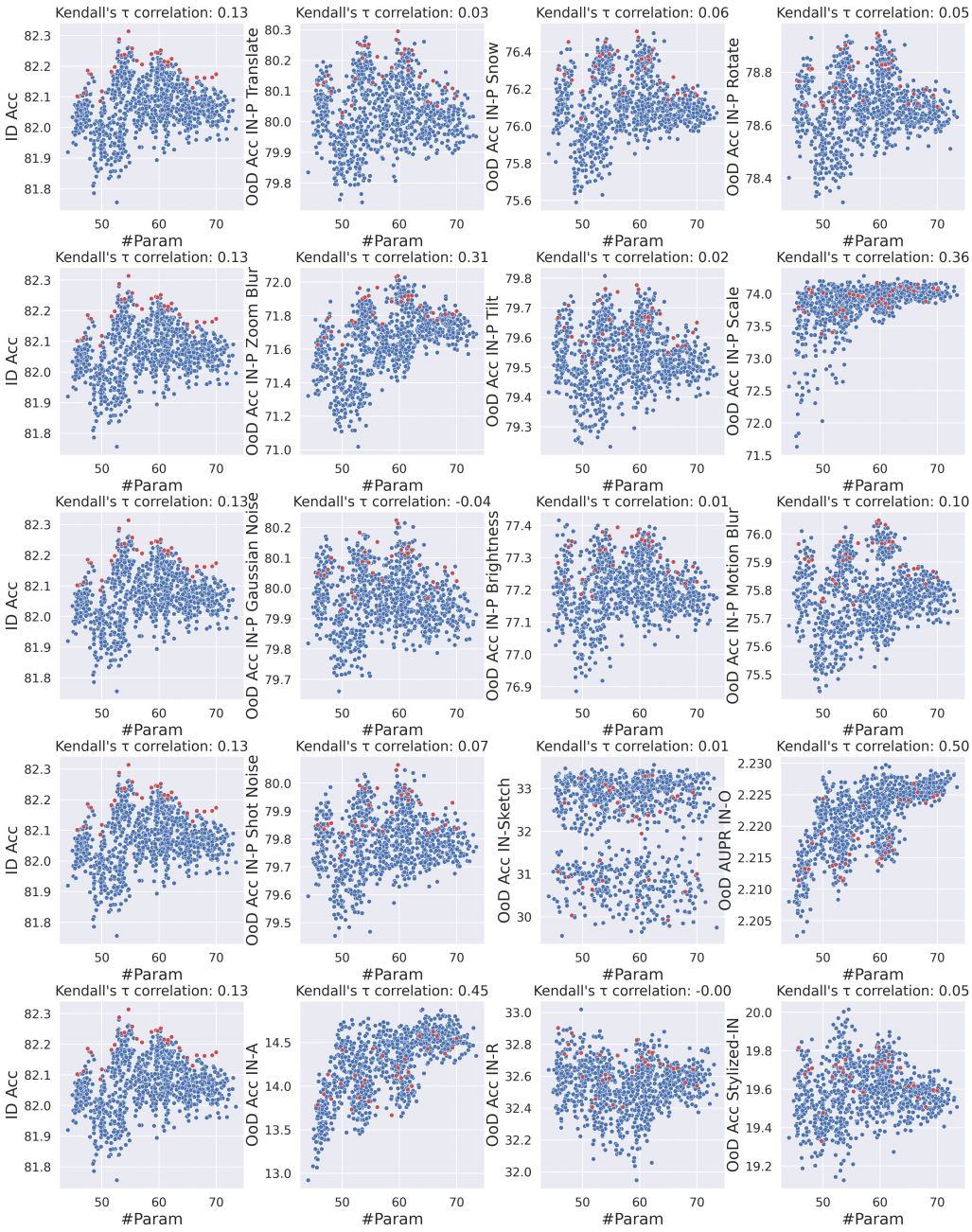

Figure M.21: Visualization of Pareto architectures in *Autoformer-Base*. The left panels show the ID accuracy, and each panel on columns 2 to 4 shows results from the OoD accuracy of IN-P, IN-A, IN-R, IN-Sketch, Stylized-IN, and AUPR of IN-O.

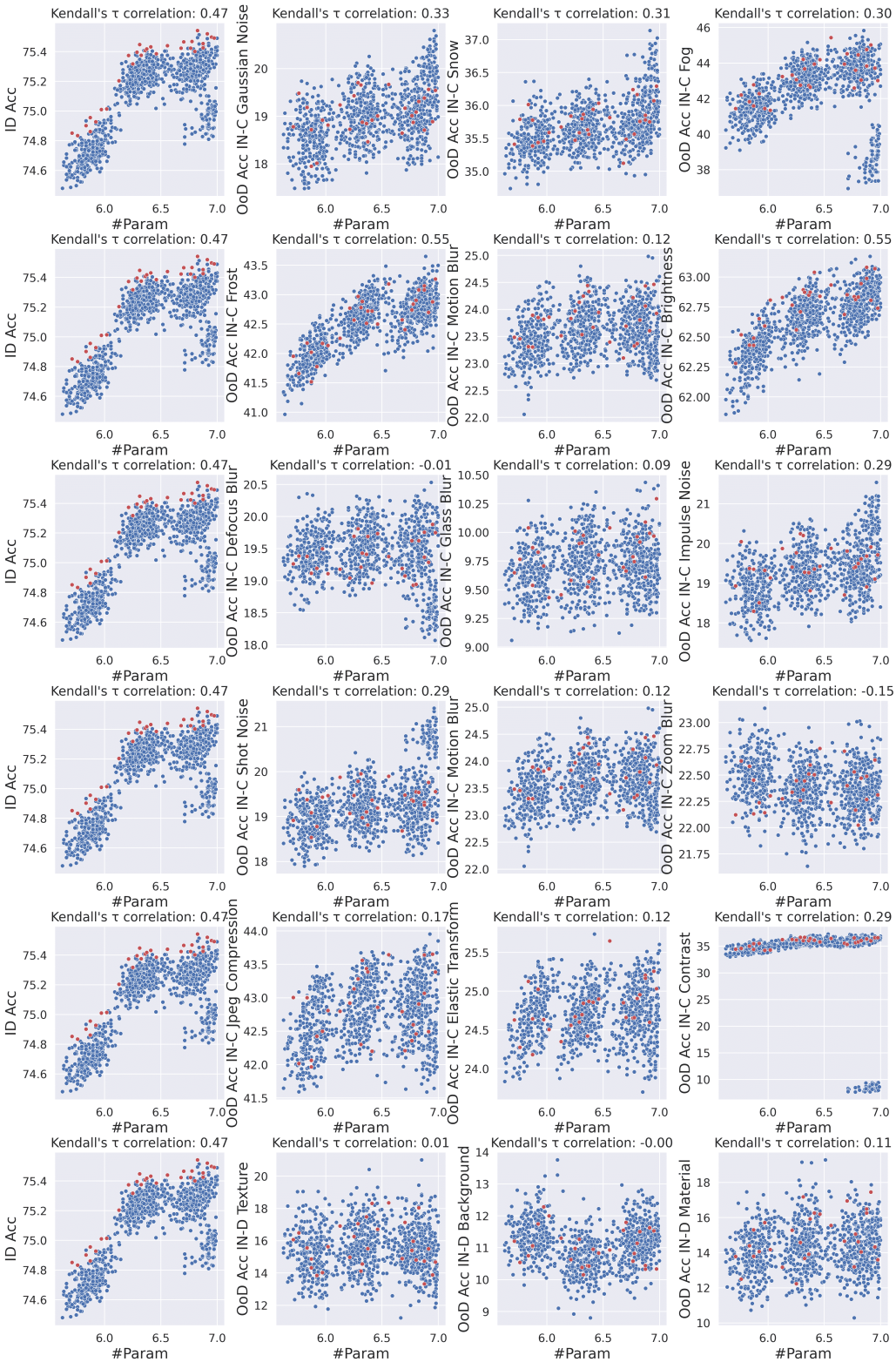

Figure M.22: Visualization of Pareto architectures in *Autoformer-Tiny*. The left panels show the ID accuracy, and each panel on columns 2 to 4 shows results from the OoD accuracy of IN-C common corruptions and IN-D.

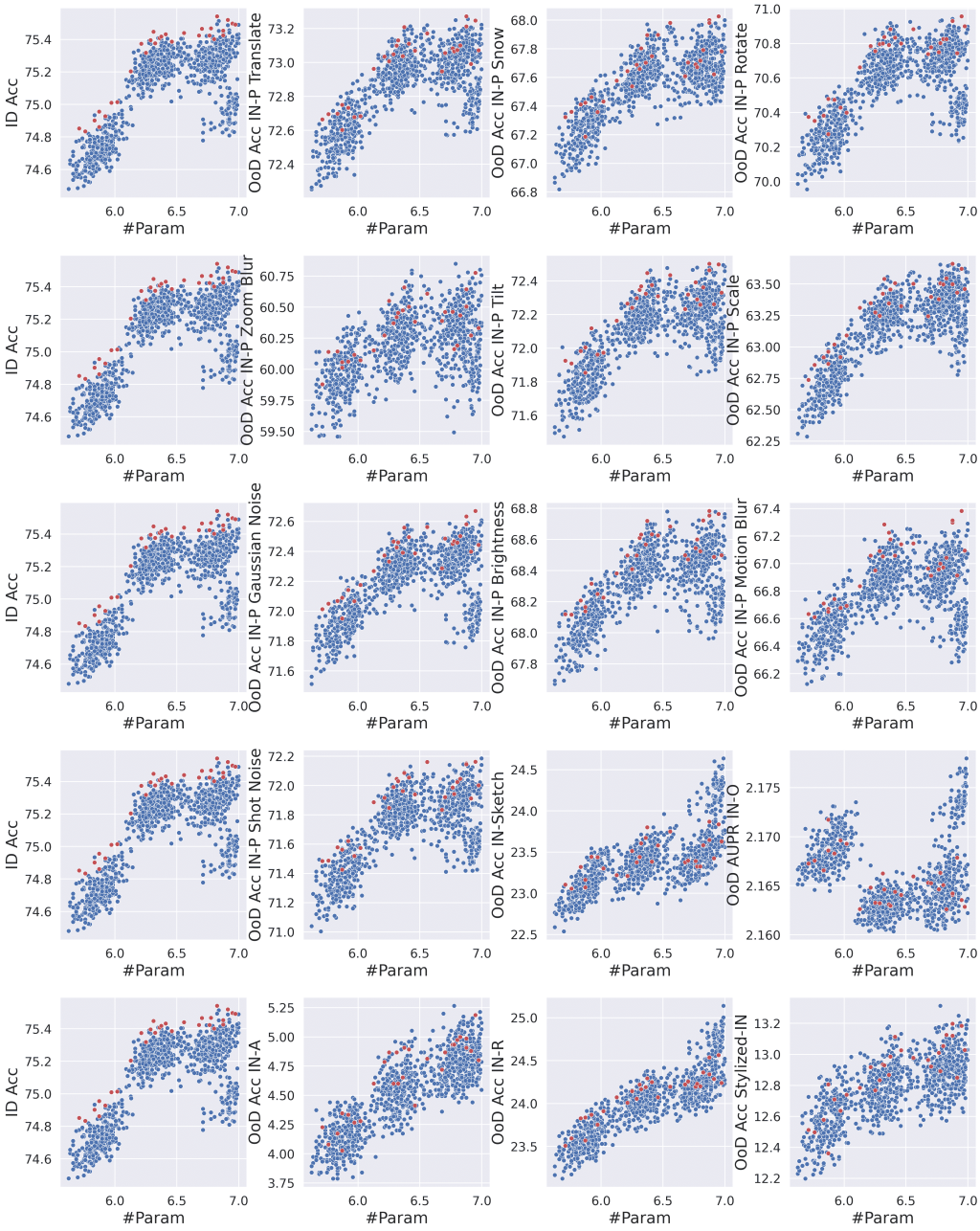

Figure M.23: Visualization of Pareto architectures in *Autoformer-Tiny*. The left panels show the ID accuracy, and each panel on columns 2 to 4 shows results from the OoD accuracy of IN-P, IN-A, IN-R, IN-Sketch, Stylized-IN, and AUPR of IN-O.

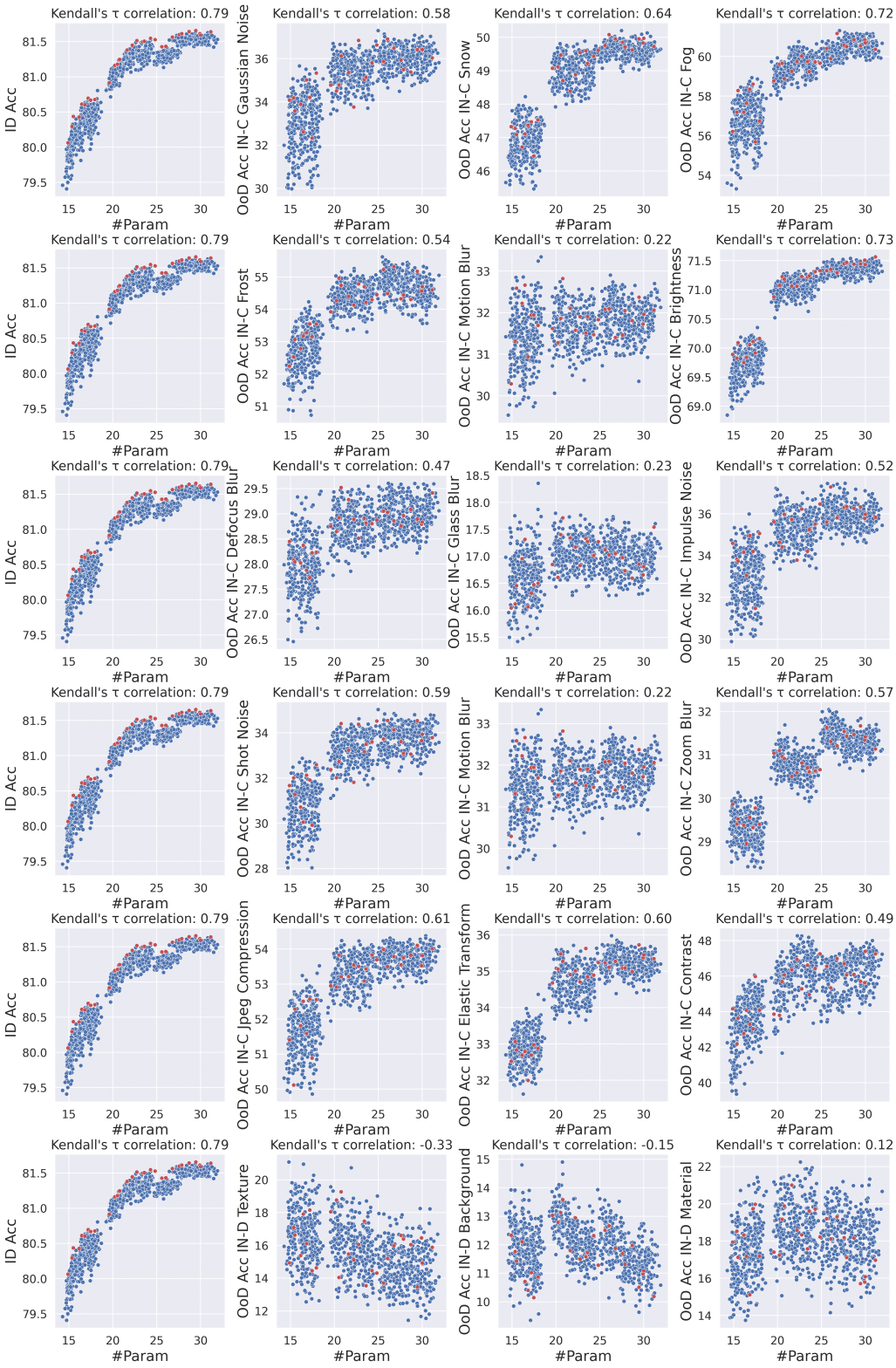

Figure M.24: As in Figure 4, we show that lower OoD accuracy can be obtained for the higher ID accuracy in the Pareto architectures of *Autoformer-Small*. The left panels show the ID accuracy, and each panel in columns 2 to 4 shows results from OoD accuracy of IN-C common corruptions and IN-D. We observe that architectural designs have a greater effect on OoD accuracy than ID accuracy, especially when OoD shifts become more severe.

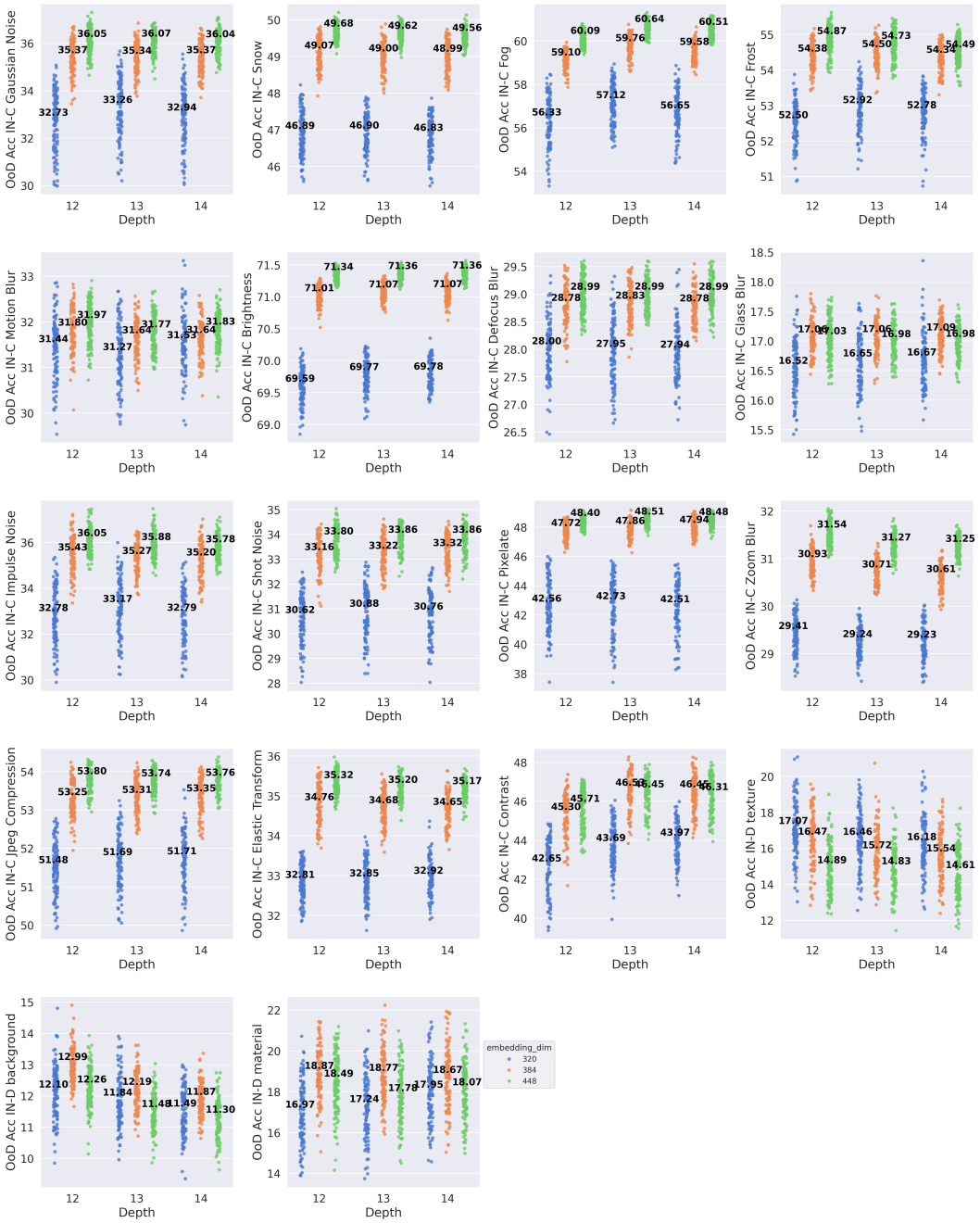

Figure M.25: As in Figure 5, we show the potential impact of embedding dimension (Embed_Dim) on OoD generalization in ViTs architectures sampled from Autoformer-Small. The numbers denote the average OoD performance, and The data points with blue ●, orange ●, and green ● colours represent ViT architectures with the embedding dimension of 320, 384, and 448, respectively. Each panel shows results from the OoD accuracy of IN-C common corruptions and IN-D.

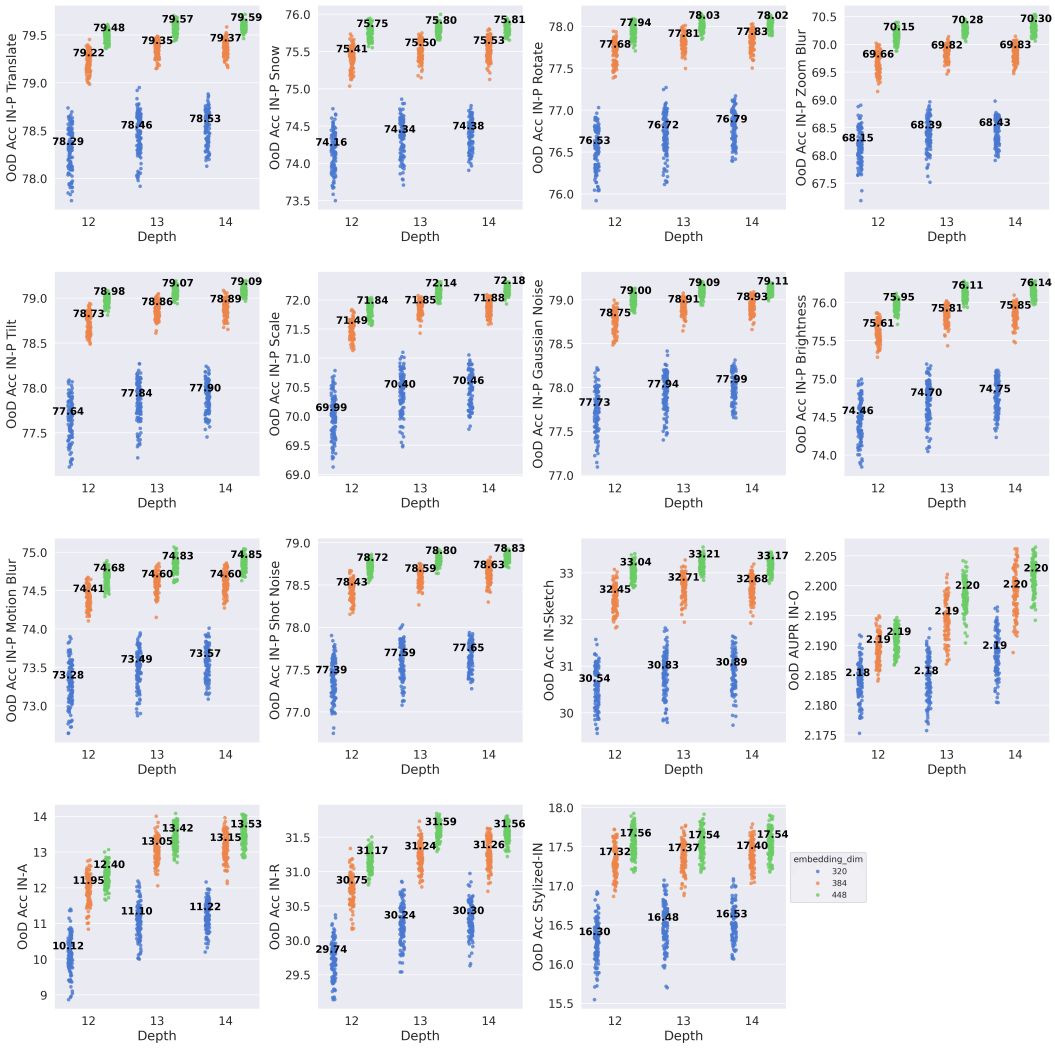

Figure M.26: As in Figure 5, we show the potential impact of embedding dimension (Embed_Dim) on OoD generalization in ViTs architectures sampled from Autoformer-Small. The numbers denote the average OoD performance, and The data points with blue ●, orange ●, and green ● colours represent ViT architectures with the embedding dimension of 320, 384, and 448, respectively. Each panel shows results from the OoD accuracy of IN-P, IN-A, IN-R, IN-Sketch, Stylized-IN, and AUPR of IN-O.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Sec. 3 for our OoD-ViT-NAS benchmark, and Sec. 4 for our analysis. The summary can be found in Fig. 2, Fig. 1, and Tab. 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] ,

   Justification: See Appx. B

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Appx. M

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: See Appx. A

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Appx. M

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Tab. 2. Our benchmark and analysis are conducted across datasets and search space.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appx. M, $\sim$ 3900 GPU hours on a single Nvidia RTX A6000

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appx. B

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Appx. E and Appx. M for description of dataset and experimental setups, respectively. We do not mention the license of the assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: See Appx. A for our OoD-ViT-NAS benchmark. See Sec. 3, and Appx. E for the description of the benchmark.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.