Dissecting Representation Structure in Vision Transformers: A Rigorous Architectural Study

Kim-Cuc Nguyen[1] Ngai-Man Cheung[1]
[1]Singapore University of Technology and Design
kimcucnguyen.cs@gmail.com ngaiman_cheung@sutd.edu.sg

Abstract

Representation structure is crucial for understanding Vision Transformer (ViT) architectures and their generalization behavior. However, prior studies neither isolate nor analyze module-level features nor investigate how their interactions contribute to performance estimation. In this work, we conduct a rigorous analysis of feature information across diverse architectural scales, empirically uncover the relationship between ViT representation and generalization behavior, and leverage these insights to guide efficient ViT design. Our contributions are fivefold: Across diverse architectural scales, 1) We identify feature collapse at initialization, which leads to redundancy, and propose a reduction scheme to mitigate this issue. 2) We quantify feature information using entropy and the minimum eigenvalue, demonstrating that these metrics serve as reliable indicators for generalization prediction. 3) We show that feature in the token space provides a more faithful representation than those in embedding space. 4) We discover an unexpected finding: features produced by linear submodules within ViT layers are critical for the prediction of generalization performance. 5) Our proposed proxy improves the correlation ranking by 18-48% over prior baselines and can effectively identify ViT architectures that achieve higher accuracy at lower or comparable computational cost.

1. Introduction

The Vision Transformer (ViT) [14] has been successfully developed and applied across various domains [17, 33, 66], achieving promising results, particularly in classification tasks [2, 13, 49]. The critical components of ViT have been considered to be multi-head self-attention and the multi-layer perceptron (MLP), which notably distinguish it from convolutional neural networks (CNNs) [8, 22, 25, 44]. These specific modules enable the ViT architecture to achieve superior performance, but also result in less interpretable representations. By understanding the specific feature information in ViT, we can design more flexible architectures with stronger generalization ability.

Generalization potential refers to a model's inherent capacity to generalize prior to training, providing a way to estimate its expected performance without optimization. Although it remains a conundrum, it is a valuable concept for effectively designing ViT architectures. Moreover, to facilitate the adaptation and application of ViT, Neural Architecture Search (NAS) [5, 6, 12, 19, 34, 43, 45, 53, 60] has been introduced to automatically discover optimal ViT architectures within a predefined search space, aiming to achieve the best possible accuracy under computational constraints. However, existing ViT NAS methods demand extensive computational resources, as they require training a large Supernet, evaluating numerous sub-architectures, and selecting the optimal one. To mitigate this cost, training-free performance proxies [57, 68, 69] can be used to estimate the performance of ViT architectures without exhaustively training all candidates in the search space.

Research gap. Existing studies primarily focus on overall architectural design rather than analyzing the specific roles of individual modules. While several proxy-based methods have been proposed to predict network performance, most are adapted from CNN proxies [30, 32, 50, 53], which do not generalize well to ViT architectures due to their distinct structural properties. Current ViT proxies [36, 57, 68, 69] typically operate in the weight space or treat the entire architecture as a single entity, overlooking the contribution of individual modules or features. Moreover, the high dimensionality and depth of ViT features make it challenging to quantify their information effectively.

In this paper, we address existing research gaps by *proposing a framework and conducting the first analytical study of different types of feature information across various aspects of ViT architectural scales*. Furthermore, we quantitatively characterize feature information using entropy and the minimum eigenvalue, and show that the outputs of linear submodules are both critical and ubiquitous for generalization prediction.

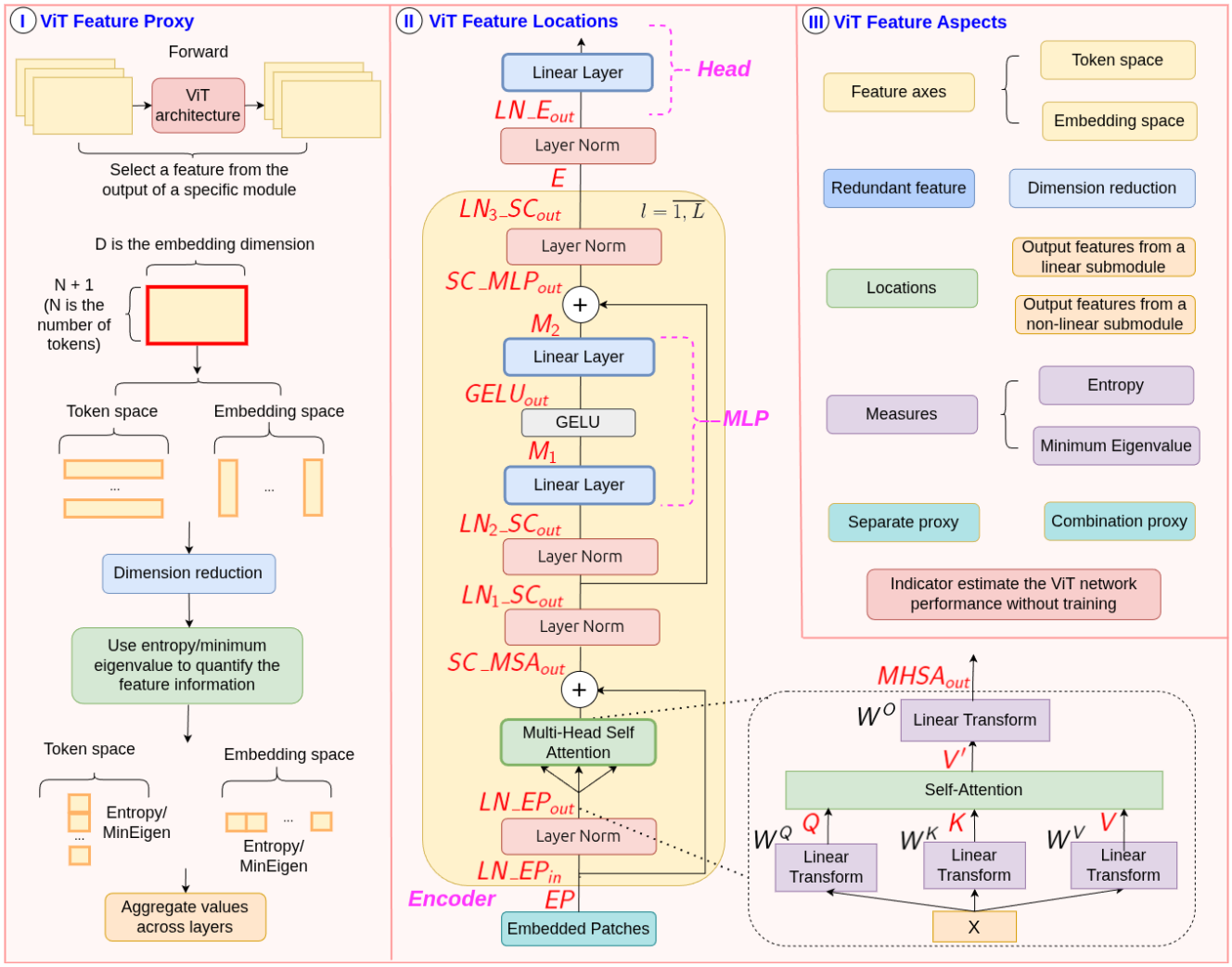

Figure 1. Overview of our proposed framework for rigorous ViT representation analysis. **(I)** Procedure for computing feature proxies from specific modules (Sec. 3). **(II)** Visualization of feature locations across ViT modules, with red text indicating feature types and extraction points. **(III)** Analysis of various feature aspects (Sec. 4). We observe feature collapse at initialization (Sec. 4.1) and propose a reduction scheme to mitigate redundancy. In Tab. 1, **we demonstrate that entropy and the minimum eigenvalue effectively capture feature informativeness, with features output of linear submodule showing strong correlation with test accuracy**, particularly in the token space. The proposed optimal feature proxies predict ViT performance without training, improving correlation ranking and achieving higher accuracy with lower computational cost (Sec. 4.2, 5, F).

2. Preliminaries

Definition 1 (Shannon Entropy)**.** The entropy $H(X)$ of a discrete random variable X with possible outcomes $x_1, x_2, \ldots, x_n$ and probability function $P(X)$ is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{1}$$

Definition 2 (Pearson Correlation Matrix)**.** Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be a feature matrix, where each column X_i represents one feature vector. The Pearson correlation matrix $\mathbf{P} \in \mathbb{R}^{D \times D}$ is defined as:

$$P_{ij} = \frac{\mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]}{\sigma_{X_i}\, \sigma_{X_j}} \tag{2}$$

where σ_{X_i} is the standard deviation of X_i.

3. Vision Transformer Feature Proxies

In our analysis, we investigate which types of feature and their combinations are the most suitable as indicators for predicting ViT performance. To address this, we first define the core concepts and introduce our proposed methods as follows:

Feature Compatibility. It refers to how features are jointly connected and how their combinations work without conflict. While previous ViT studies assume the whole network, skip connection, or later layers contain more information, table 4 show that this assumption does not hold for ViT generalization.

Feature axes. Given an input image $X \in \mathbb{R}^{C \times H \times W}$,

it is divided into non-overlapping patches, each flattened and linearly projected into a D dimensional embedding. A positional embedding and a learnable class token are then added, forming $X \in \mathbb{R}^{t_{\text{dim}} \times e_{\text{dim}}}$, where $t_{\text{dim}} = N + 1$ is the token dimension and $e_{\text{dim}} = D$ is the embedding dimension. For feature analysis, we treat rows as variables in the token space and columns as variables in the embedding space.

ViT Feature. ViT architecture consists of an Encoder, Patches, and MLP head, illustrated in figure 1. For each module type, we extract one representative feature from its output during a forward pass on a batch of data. Starting from the input, EP denotes the feature after patch embedding. In the Encoder, repeated over L layers, we obtain features from all layers within each module. LN_EP_{in} and LN_EP_{out} represent features before and after the first LayerNorm, respectively. The query, key, and value features, denoted as Q, K, and V, are concatenated into $QKV = XW^{QKV}$ for efficiency. V' denotes the feature after self-attention, while $MHSA_{out}$ is the multi-head attention output. SC_MSA_{out} represents the feature after the skip connection following multi-head self-attention. Next, $LN_1\_SC_{out}$ and $LN_2\_SC_{out}$ represent the features after the first and second LayerNorm operations following the skip connection, respectively. Within the MLP, M_1 and M_2 denote features from the first and second linear layers, and $GELU_{out}$ is the activation output. SC_MLP_{out} is the feature after the skip connection following the MLP module, and $LN_3\_SC_{out}$ denotes the feature after the subsequent LayerNorm operation. Finally, E represents the encoder output, and LN_E_{out} is the feature before the ViT head. Feature combinations are denoted using the plus operator (e.g., $M_1 + M_2$).

Feature dimension reduction. Representation information measures are computed on X', and the results are scaled by $\frac{t_{\text{dim}}}{u}$ in the token space or $\frac{e^l_{\text{dim}}}{u}$ in the embedding space to maintain consistency with the original dimensionality. This weighting compensates for dimensionality reduction and ensures that the final information metric reflects the contribution of the entire feature space.

Entropy Proxy. Let $H(x_i)$ denote the entropy of a feature vector x_i. For each row or column of the feature matrix X, we compute $H(x_i)$ and select the u vectors with the lowest entropy to form a reduced representation X'.

$$X' = \text{Select } u \text{ vectors from X with lowest } H(x_i) \tag{3}$$

The entropy proxy function $F_{\mathcal{I}}$ is then computed from the reduced matrix X'.

$$F_{\mathcal{I}}(X') = \sum_{i=1}^{u} H(x_i) \tag{4}$$

Minimum Eigenvalue Proxy. We compute the Pearson correlation matrix $P(X)$ (Eq. 2) to measure the linear correlation between feature vector variables x_i and x_j. We define the row-wise correlation score s_i, which sum the absolute correlations in each row of $P(X)$.

$$s_i = \sum |P(X)_{ij}| \tag{5}$$

$$X' = \text{Select } u \text{ vectors with lowest } s_i \tag{6}$$

$$F_{\mathcal{I}}(X') = \lambda_{\min}\big(P(X')\big) \tag{7}$$

$\lambda_{\min}(P)$ is proportional to the degree of feature diversity and independence, where higher diversity correlates with better classification accuracy, stronger generalization, and more stable training in ViTs.

Algorithm 1 Compute ViT Feature Proxy

Require: Feature type f; feature map $X \in \mathbb{R}^{t_{dim} \times e^l_{dim}}$ corresponding to a single sample in the batch, extracted from module f at layer l; reduction dimension u; proxy indicator $\mathcal{I} \in \{\text{Entropy}, \text{MinEigen}\}$

```
proxy ← 0
for l ← 0 to L do
    if dim is token then
        X = {x_1, ..., x_{t_dim}}, x_i ∈ R^{e^l_dim}
        α_l ← t_dim / u
    else
        X = {x_1, ..., x_{e^l_dim}}, x_i ∈ R^{t_dim}
        α_l ← e^l_dim / u
    end if
    X' ← Reduce(X, u, I) {select u elements using I-specific rule} (see Eqs. (3), (6))
    p_l ← α_l · F_I(X') (see Eqs. (4), (7))
    proxy ← proxy + p_l
end for
return proxy
```

4. Discovering the Impact of Feature Information on ViT Generalization Predictors

Procedure. Our framework and analysis components are illustrated in Figure 1. We meticulously designed the experimental testbed based on a one-shot NAS method [51, 52, 63]. Specifically, we sampled 1,000 ViT architectures within the 5–7M parameter range from AutoFormer-Tiny [5], extracted their architectural information, and stored it as an API. Additionally, we collected the test accuracy of these 1,000 architectures using pretrained weights on the ImageNet-1K dataset [42]. The test accuracy of subnets inheriting weights from the supernet achieves comparable results to those trained independently [5, 63]. We propose a unified framework for computing ViT feature proxies, as illustrated in Algorithm 1, and introduce two measures: the *Entropy Proxy* and the *Minimum Eigenvalue Proxy*.

Metrics. We computed the Spearman [26] and Kendall [1] correlations between proxy values and test accuracy to evaluate the effectiveness of representation information in capturing ViT generalization performance.

4.1. Feature Collapse Occurs at Initialization

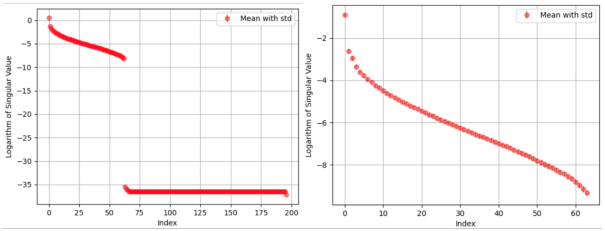

Figure 2. Logarithm of the singular value spectrum of the feature Q in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

Table 1. Spearman's ρ and Kendall's τ correlations between test accuracy and each of the **entropy** and **minimum eigenvalue** proxies are computed for 1,000 ViT architectures (5–7M parameters) across 26 feature types. These proxies are calculated in both token and embedding spaces for individual and combined features, with dimension reductions $u = 1$. **Bold** values indicate correlations higher than those of previous proxies in Table 2. **The entropy and minimum eigenvalue proxies effectively quantify feature information as indicators of ViT generalization. Features output by linear submodules (Q, K, V, V', $MHSA_{out}$, M_1, M_2) and their combinations show strong correlations with test accuracy, whereas features output by non-linear submodules exhibit significantly weaker correlations. Proxies computed in the token space achieve higher correlations than those in the embedding space.**

Proxy		Entropy				Minimum Eigenvalue			
		Token		Embedding		Token		Embedding	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ
Encoder	Q	**0.745**	**0.509**	**0.732**	**0.480**	**0.796**	**0.636**	**0.794**	**0.633**
	K	**0.733**	**0.485**	**0.738**	**0.495**	**0.792**	**0.630**	**0.797**	**0.637**
	V	**0.744**	**0.503**	**0.718**	**0.468**	**0.797**	**0.637**	**0.796**	**0.637**
	V'	**0.787**	**0.578**	0.666	0.453	**0.792**	**0.628**	**0.699**	**0.540**
	$MHSA_{out}$	**0.790**	**0.585**	0.657	0.439	**0.794**	**0.633**	**0.701**	**0.541**
	M_1	**0.844**	**0.660**	**0.762**	**0.571**	**0.795**	**0.633**	**0.761**	**0.585**
	M_2	**0.809**	**0.599**	**0.676**	**0.463**	**0.761**	**0.589**	**0.698**	**0.537**
	$GELU_{out}$	0.137	0.104	0.131	0.100	0.016	0.013	0.129	0.104
	SC_MSA_{out}	-0.039	-0.023	-0.056	-0.038	0.008	0.007	-0.138	-0.109
	SC_MLP_{out}	-0.023	-0.014	-0.043	-0.029	0.042	0.034	-0.138	-0.109
	LN_EP_{in}	-0.024	-0.014	-0.050	-0.033	0.021	0.017	-0.138	-0.109
	LN_EP_{out}	-0.031	-0.019	-0.071	-0.047	0.052	0.042	-0.142	-0.113
	$LN_1\_SC_{out}$	-0.039	-0.023	-0.056	-0.038	0.008	0.007	-0.138	-0.109
	$LN_2\_SC_{out}$	-0.029	-0.017	-0.055	-0.038	0.037	0.029	-0.139	-0.110
	$LN_3\_SC_{out}$	-0.023	-0.014	-0.043	-0.028	0.042	0.034	-0.138	-0.109
Patches	EP	-0.192	-0.158	-0.192	-0.157	NA	NA	-0.192	-0.157
MLP Head	E	-0.055	-0.034	-0.088	-0.059	0.076	0.061	-0.126	-0.096
	LN_E_{out}	-0.011	-0.006	0.216	0.133	0.091	0.074	-0.086	-0.067
QKV		**0.773**	**0.544**	**0.720**	**0.496**	**0.795**	**0.634**	**0.734**	**0.529**
$M_1 + M_2$		**0.849**	**0.665**	**0.752**	**0.562**	**0.797**	**0.639**	**0.752**	**0.576**
$QKV + M_1 + M_2$		**0.838**	**0.652**	**0.837**	**0.649**	**0.798**	**0.640**	**0.836**	**0.654**
$QKV + MHSA_{out}$		**0.805**	**0.604**	**0.773**	**0.553**	**0.798**	**0.639**	**0.771**	**0.567**
$MHSA_{out} + M_1 + M_2$		**0.851**	**0.669**	**0.744**	**0.556**	**0.798**	**0.640**	**0.745**	**0.570**
$QKV + MHSA_{out} + M_1 + M_2$		**0.856**	**0.676**	**0.759**	**0.574**	**0.798**	**0.640**	**0.831**	**0.650**
$Q + K + V$		**0.740**	**0.499**	**0.728**	**0.477**	**0.798**	**0.640**	**0.798**	**0.639**
$SC\_MSA_{out} + SC\_MLP_{out}$		-0.040	-0.024	-0.056	-0.037	0.027	0.022	-0.143	-0.114

4.2. Comparison with State-of-the-Art Methods

We compare the top three Token-Entropy proxies with reduced dimension $u = 1$: M_1+M_2, $MHSA_{out}+M_1+M_2$, and $QKV + MHSA_{out} + M_1 + M_2$, against the baselines across 3,000 ViT architectures: Tiny (5-7M), Small (15-19M), and Base (45-47M), with 1,000 architectures in each parameter range. As shown in Table 2, our proxies improve correlation rankings by 18%, 25%, and 48% for the 5–7M, 15–19M, and 45–47M ranges, respectively.

Table 2. Spearman's ρ and Kendall's τ correlation rankings for three sets of 1,000 ViT architectures with parameter ranges of 5–7M, 15–19M, and 45–47M, comparing the top-3 Token-Entropy proxies ($u = 1$) with baselines. **Our proposed proxies significantly improve the correlation rankings, demonstrating that they are strong indicators of ViT performance.**

Proxy	5-7M		15-19M		45-47M	
	ρ	τ	ρ	τ	ρ	τ
SNIP [30]	0.313	0.207	0.280	0.190	0.056	0.037
GraSP [50]	-0.101	-0.066	-0.064	-0.043	0.023	0.015
TE-score [7]	-0.319	-0.219	-0.084	-0.057	-0.106	-0.072
NASWOT [36]	0.382	0.278	0.232	0.162	0.243	0.171
DSS [68]	0.622	0.439	0.468	0.315	-0.119	-0.079
AutoProxA [57]	0.675	0.477	0.446	0.299	-0.126	-0.084
DSS++ [69]	0.638	0.448	0.450	0.302	-0.140	-0.094
$M_1 + M_2$	**0.849**	**0.665**	**0.718**	**0.529**	**0.368**	**0.251**
$MHSA_{out} + M_1 + M_2$	**0.851**	**0.669**	**0.717**	**0.529**	**0.360**	**0.246**
$QKV + MHSA_{out} + M_1 + M_2$	**0.856**	**0.676**	**0.720**	**0.533**	**0.350**	**0.239**

5. Efficient ViT Architecture Design

We use the previous proxies and our Token-Entropy proxies with reduced dimension $u = 1$ to search for the optimal subnet on the same set of 1,000 ViT architectures within 5-7 parameter ranges. As shown in the table 3, our proposed proxies successfully identify the optimal subnet, improving accuracy by 0.11% compared to previous proxies across the 1,000 ViT architectures in the 5–7M ranges, respectively. Moreover, our proxies achieve an 87.5%–98.6% reduction in computation time when searching over 1,000 ViT architectures, requiring only 0.06 hours compared to 0.48 – 4.3 hours for previous proxies in the 5–7M range.

Table 3. Results on ImageNet using different proxies to search for the optimal architecture among 1,000 ViT architectures, with the 5–7M parameter ranges shown in the top and bottom sections, respectively

Proxy	Param(M) ↓	FLOPs(B) ↓	Top-1 ↑	Time(h) ↓
SNIP [30]	6.9	1.7	74.89	1.00
GraSP [50]	5.7	1.4	74.47	1.38
TE-score [7]	5.9	1.5	74.60	4.30
NASWOT [36]	6.9	1.7	74.94	0.82
DSS [68]	6.9	1.6	75.29	0.60
AutoProxA [57]	6.9	1.5	75.32	0.48
DSS++ [69]	6.9	1.6	75.30	0.80
$M_1 + M_2$	**6.9**	**1.5**	**75.41**	**0.05**
$MHSA_{out} + M_1 + M_2$	**6.9**	**1.5**	**75.43**	**0.06**
$QKV + MHSA_{out} + M_1 + M_2$	**6.9**	**1.5**	**75.38**	**0.08**

6. Conclusion

In this work, we provide a rigorous analysis of how feature representations relate to ViT generalization, considering feature information, axes, dimensions, and locations. Our proposed feature proxies improve correlation ranking by 18–48% over baselines across parameter ranges. For efficient ViT architecture design, our method achieves higher accuracy across architectures while significantly reducing training, search, and proxy computation costs. Extensive experiments across different ViT architectural scale and datasets confirm the effectiveness and broad applicability of our approach, establishing a principled framework for understanding ViT features, predicting generalization, and enabling efficient ViT design.

References

[1] Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510, 2007. 3

[2] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021. 1

[3] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022. 2

[4] 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. 4, 5

[5] 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. 1, 3, 2, 4, 5, 19

[6] 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. 1

[7] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. *arXiv preprint arXiv:2102.11535*, 2021. 4, 8, 9, 17

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 1

[9] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 5

[10] Hien Dang, Tho Tran, Stanley Osher, Hung Tran-The, Nhat Ho, and Tan Nguyen. Neural collapse in deep linear networks: from balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*, 2023. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 4, 5, 13, 18, 19

[12] 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. 1

[13] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12124–12134, 2022. 1

[14] 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. 1, 5, 19

[15] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019. 4

[16] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018. 4

[17] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 1

[18] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pages 10929–10974. PMLR, 2023. 1

[19] Chengyue Gong and Dilin Wang. Nasvit: Neural architecture search for efficient vision transformers with gradient conflict-aware supernet training. *ICLR Proceedings 2022*, 2022. 1

[20] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in neural information processing systems*, 34:15908–15919, 2021. 5

[21] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019. 4

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 2, 13

[24] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11936–11945, 2021. 3, 5

[25] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1

[26] Zar JH. Spearman rank correlation. encyclopedia of biostatistics. *John Wiley & Sons, Ltd.(eds) vol (7). Online ISBN: 9780470011812— DOI*, 10:0470011815, 2005. 3

[27] 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. 4

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5, 19

[29] Thomas Laurent, James H von Brecht, and Xavier Bresson. Feature collapse. *arXiv preprint arXiv:2305.16162*, 2023. 1

[30] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. 1, 4, 8, 9, 17

[31] Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. *arXiv preprint arXiv:1906.06307*, 2019. 4

[32] Guihong Li, Yuedong Yang, Kartikeya Bhardwaj, and Radu Marculescu. Zico: Zero-shot nas via inverse coefficient of variation on gradients. *arXiv preprint arXiv:2301.11300*, 2023. 1

[33] 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. 1

[34] 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. 1

[35] 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. 5

[36] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. Neural architecture search without training. In *International conference on machine learning*, pages 7588–7598. PMLR, 2021. 1, 4, 8, 9, 17

[37] Quynh N Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems*, 33:11961–11972, 2020. 4

[38] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2

[39] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021. 2

[40] Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *International Conference on Machine Learning*, pages 28729–28745. PMLR, 2023. 2

[41] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[43] 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. 1, 4, 5

[44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[45] 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. 1

[46] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12165–12174, 2022. 1

[47] Tom Tirer, Haoxiang Huang, and Jonathan Niles-Weed. Perturbation analysis of neural collapse. In *International Conference on Machine Learning*, pages 34301–34329. PMLR, 2023. 2

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 5, 19

[49] 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. 1

[50] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020. 1, 4, 8, 9, 17

[51] 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. 3

[52] 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. 3

[53] 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. 1, 3

[54] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022. 2

[55] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 5

[56] Zimian Wei, Hengyue Pan, Lujun Li, Peijie Dong, Zhiliang Tian, Xin Niu, and Dongsheng Li. Tvt: Training-free vision transformer search on tiny datasets. *arXiv preprint arXiv:2311.14337*, 2023. 2

[57] 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*, pages 15814–15822, 2024. 1, 4, 2, 3, 5, 8, 9, 13, 15, 17, 18

[58] Jing Xu and Haoxiong Liu. Quantifying the variability collapse of neural networks. In *International Conference on Machine Learning*, pages 38535–38550. PMLR, 2023. 2

[59] Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? on the role of simplicity bias in class collapse and feature suppression. In *International Conference on Machine Learning*, pages 38938–38970. PMLR, 2023. 2

[60] Caixia Yan, Xiaojun Chang, Zhihui Li, Lina Yao, Minnan Luo, and Qinghua Zheng. Masked distillation advances self-supervised transformer architecture search. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[61] Yongyi Yang, Jacob Steinhardt, and Wei Hu. Are neurons actually collapsed? on the fine-grained structure in neural representations. In *International Conference on Machine Learning*, pages 39453–39487. PMLR, 2023. 2

[62] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818, 2022. 1

[63] 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 *European Conference on Computer Vision*, pages 702–717. Springer, 2020. 3

[64] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021. 5

[65] Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR, 2023. 1

[66] 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. 1

[67] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 2

[68] 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. 1, 4, 2, 3, 5, 8, 9, 17, 18, 19

[69] 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. 1, 4, 2, 3, 5, 8, 9, 17, 18, 19

[70] Chuang Zhu, Wenkai Chen, Ting Peng, Ying Wang, and Mulan Jin. Hard sample aware noise robust learning for histopathology image classification. *IEEE transactions on medical imaging*, 41(4):881–894, 2021. 2

Dissecting Representation Structure in Vision Transformers: A Rigorous Architectural Study

Supplementary Material

In the supplementary material, we provide additional explanations of the method, detailed experimental configurations, extended results, further analyses, and ablation studies to support our findings. These contents are not included in the main paper due to space constraints.

Contents

A. Related Work

We summarize related work as follows:

Feature collapse. [29] is among the first works to explore its features in detail, explaining them through the concept of word representations. Importantly, the authors demonstrate that words belonging to the same concept receive identical representations, a phenomenon they define as feature collapse. Using ranking of embedding to represent the information without the join of label [18] has been used in self-supervised learning. [65] introduce the concept of entropy collapse. In our work, we observe the feature collapse at the initialization of the ViT.

Feature selection in ViT. Feature selection in ViTs involves identifying the most informative components of the features extracted by the model. Effective feature selection can reduce computational cost, improve generalization, and enhance model interpretability. [41] proposed a method to prune redundant tokens based on the input by the design prediction module to identify the important token from current features. [62] introduced to use an adaptive token mechanism to reduce the number of token processed in ViT. [46] proposed removing uninformative patches by identify-

ing the most effective patches in the final layer and using this information for patch selection in earlier layers.

Proxy for ViT. [68] proposed measuring the estimated performance of ViT architecture via synaptic diversity and synaptic saliency. [57] proposed a method to search for a ViT proxy that is generalizable for various domains and datasets. [56] proposed ViT proxy search for tiny datasets with knowledge distillation. We summarize related work in Supp.

Feature representation in ViT. [67] shows that the attention maps become the same when the transformer goes deeper. They demonstrate that the feature map is only identical in the top layers of the deep ViT model. Therefore, their findings indicate that in the deeper layers of ViT, the self-attention mechanism struggles to learn meaningful representations, which limits the model's ability to achieve the expected performance improvements. [39] analyzes the internal representations of ViTs and shows that ViTs exhibit more uniform representations across all layers. Moreover, they demonstrate that ViTs capture more global information than CNNs in the lower layers. [54] demonstrates that attention features behave as a low-pass filter from a Fourier-domain perspective, and proposes addressing this issue by introducing a high-pass filtering component that mixes feature information across multiple frequency bands. [3] studies deep features of pretrained ViT models and uses them as dense descriptors for segmentation tasks.

Neural collapse. Features are the output of each module in the neural network. Therefore, feature collapse has a close relationship with neural collapse. The phenomenon of neural collapse means that the variability of the outputs of the penultimate layer in one class decreases. [47] proposes to observe and prove the reduction in one class. [58] proposes a method to quantify the collapse phenomenon in the neural network. [10] analyzes the collapse phenomenon in deep linear neural networks. [40] investigates the neural collapse that happens in the intermediate layer. [61] shows that the collapse hides the structure of feature representation. [59] investigates the phenomena of class collapse.

B. Computing Resources

All experiments in this paper are conducted on the following workstations:

- A workstation with an AMD Ryzen CPU and two NVIDIA A6000 GPUs, each with 46 GB of memory.
- A workstation with an AMD Ryzen CPU and four NVIDIA A5000 GPUs, each with 24 GB of memory.

C. Dataset

In this study, we employed six datasets in our experiments:

- ImageNet-1K [11]: A large-scale, diverse, and widely used dataset for computer vision, particularly for image classification tasks. The dataset contains 1,331,167 images, with 1,281,167 images in the training set and 50,000 images in the validation set. It covers 1,000 classes, and each image is labeled with its corresponding class. ImageNet-1K serves as a standard benchmark for CNNs, Vision Transformers (ViTs), and other deep learning models. Model performance on this dataset is typically evaluated using Top-1 and Top-5 classification accuracy.
- ImageNet-C [23]: A robustness benchmark in computer vision for the image classification task. The dataset consists of various types of corruptions, such as Gaussian noise, impulse noise, and changes in brightness. It is designed to simulate real-world conditions where inputs may be degraded, allowing evaluation of model robustness.
- CIFAR-10 [28]: A dataset commonly used for image classification tasks. The dataset consists of 60,000 color images of size 32×32 pixels. It is divided into 50,000 training images and 10,000 test images. The dataset contains 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, with each class containing 6,000 images.
- CIFAR-100 [28]: A commonly used image classification dataset, CIFAR-100 consists of 60,000 color images of size 32×32 pixels and contains 100 classes. The dataset includes 50,000 training images and 10,000 test images. Each class has 500 training images and 100 test images.
- Oxford Flowers [38]: An image classification dataset containing 8,189 images across 102 flower categories commonly found in the United Kingdom. Each class has at least 40 images. The dataset is split into 2,040 images for training and 6,149 images for testing.
- Chaoyang [70]: A medical histopathology image classification dataset derived from colon tissue slides. It contains 6,160 image patches at a resolution of 512×512, spanning four classes: normal, serrated, adenocarcinoma, and adenoma. The dataset is divided into two subsets: 4,021 images for training and 2,139 images for testing.

We choose ImageNet-1K for our primary analysis because it is a large-scale, diverse dataset and ensures consistency with prior studies on ViT architectures, particularly in ViT design, including AutoFormer [5], DSS [68], AutoProxA [57], and DSS++ [69]. Furthermore, in constructing our API of 3,000 ViT architectures across diverse parameter ranges, we sample weights from the AutoFormer supernet trained on ImageNet-1K and evaluate each architecture's test accuracy on ImageNet-1K. Therefore, to compute the correlation rankings between proxy values and in-distribution test accuracy across these 3,000 architectures, it is essential to use the same dataset for a fair comparison.

However, our proposed ViT feature proxies are not strictly dependent on ImageNet-1K for two key reasons.

First, when passing features through the ViT architecture to quantify information, we use only one feature map from a single batch of data. Second, in the proxy experiments, we perform only a forward pass without computing gradients or updating model weights. This design makes our method highly efficient and well-suited as a ViT generalization predictor, allowing the evaluation of several thousand architectures in a short time.

When using ImageNet-C to evaluate the out-of-distribution generalization predictor, the ViT architectures are trained on ImageNet but not on the corrupted images in ImageNet-C. The networks are then tested on ImageNet-C to assess the robustness of the architectures based on out-of-distribution accuracy.

D. ViT Architectures

D.1. AutoFormer Search Space

Our 3,000 ViT architectures, covering three different parameter ranges, are constructed based on the AutoFormer [5] search space. This is a large search space commonly used for ViT NAS [53, 57, 68, 69]. The ViT architectures in AutoFormer have five changeable attributes: embedding dimension, Q-K-V dimension, MLP ratio, number of heads, and depth.

- **Embedding dimension:** Determines the width of the model. This value is the same for all transformer layers in a ViT.
- **Q-K-V dimension:** Refers to the sizes of the query, key, and value vectors used in multi-head self-attention.
- **MLP ratio:** The size of the hidden layer in the feedforward (MLP) block relative to the embedding dimension. This value can vary across transformer layers in a ViT.
- **Number of heads:** The number of separate attention mechanisms computed in parallel within a multi-head self-attention (MHSA) module. This value can vary across transformer layers in a ViT.
- **Number of depths:** The total number of transformer layers in a ViT.

In Table 4, we present the details of our API based on the AutoFormer search spaces. The table lists (α, β, δ), where α is the lower bound, β is the upper bound, and δ is the step size of a given parameter. For example, for the depth parameter $\{12, 13, 14\}$, the lowest value is 12, the highest value is 14, and the step size is 1. This configuration follows the approach in [5].

When selecting ViT architectures to compute correlation rankings, we reduced the range of parameters in each set of 1,000 architectures to 5–7M, 15–19M, and 45–47M. This adjustment makes the evaluation process less dependent on model size. Although models with more parameters generally achieve higher accuracy, some architectures with similar parameter counts can outperform others. By restricting the parameter range, we ensure that the correlation rankings more accurately reflect the effectiveness of the proxies rather than differences in model scale.

Model	Tiny	Small	Base
Embed	(192, 240, 24)	(320, 448, 64)	(528, 624, 48)
Q-K-V	(192, 256, 64)	(320, 448, 64)	(512, 640, 64)
MLP	(3.5, 4, 0.5)	(3, 4, 0.5)	(3, 4, 0.5)
Head	(3, 4, 1)	(5, 7, 1)	(8, 10, 1)
Depth	(12, 14, 1)	(12, 14, 1)	(14, 16, 1)
Params	**5-7M**	**15-19M**	**45-47M**

Table 4. ViT architectures API Tiny, Small, Base configuration

D.2. PiT Search Space

We follow [24, 57, 68, 69] to obtain the PiT search space. The ViT architectures in the PiT search space have three stages and four changeable attributes: base dimension, MLP ratio, number of heads, and depths.

- **Base dimension**: It is the embedding dimension, can be chosen from $\{16, 24, 32, 40\}$, with a minimum value of 16, a maximum value of 40, and a step size of 8. The base dimension is the same across all three stages of a ViT architecture.
- **MLP ratio**: The size of the hidden layer in the feedforward (MLP) block relative to the embedding dimension. It can be selected from $\{2, 4, 6, 8\}$, with a minimum value of 2, a maximum value of 8, and a step size of 2. The MLP ratio is the same for all transformer blocks in a given architecture.
- **Number of heads**: The number of separate attention mechanisms computed in parallel within a multi-head self-attention (MHSA) module. This attribute is represented as $[h_1, h_2, h_3]$, where h_1, h_2, h_3 corresponds to the number of heads in the three stages of the ViT architecture. Possible values include {[2,2,2], [2,2,4], [2,2,8], [2,4,4], [2,4,8], [2,8,8], [4,4,4], [4,4,8], [4,8,8], [8,8,8]}.
- **Depths**: The number of transformer blocks in each stage. This attribute is represented as $[b_1, b_2, b_3]$, where b_1, b_2, b_3 corresponds to the number of blocks in the three stages. Possible values include {[1,6,6], [1,8,4], [2,4,6], [2,6,4], [2,6,6], [2,8,2], [2,8,4], [3,4,6], [3,6,4], [3,8,2]}.

E. Design Rationale for ViT Feature Proxies

We propose a unified framework to compute ViT feature proxies, as illustrated in Algorithm 1. To assess the representational quality of ViT features, we introduce two proxy measures: the *Entropy Proxy* and the *Minimum Eigenvalue Proxy*. Both proxies share a computational pipeline comprising layer-wise feature extraction for each module, dimensionality reduction, scaling, and layer-wise proxy aggregation. Despite this shared structure, the two proxies capture different aspects of ViT representations: the Entropy Proxy quantifies information dispersion, whereas the

Minimum Eigenvalue Proxy characterizes geometric independence in the representation space.

Entropy Proxy. Entropy measures the uncertainty or randomness in a feature distribution, reflecting its information diversity. Architectures that naturally produce diverse and informative features tend to generalize better after training. By computing the entropy for each row or column, we can quantify how diverse or uncertain the information in that feature is. A high-entropy feature tends to vary irregularly across samples, often indicating noise or redundant information. In contrast, a low-entropy feature exhibits more consistent patterns, suggesting that it may capture the underlying structure or semantics of the data.

Selecting features with low entropy retains the most stable and representative features, while discarding those that contribute mainly to randomness. The compressed matrix X' maintains the essential and reliable information content, enabling efficient analysis and comparison with reduced dimensionality and noise.

Minimum Eigenvalue Proxy. In a ViT, each layer encodes diverse and complementary visual cues such as edges, shapes, textures, and semantics. When these features collapse, for example, when all tokens become highly similar, the representation loses discriminative power. A larger minimum eigenvalue ($\lambda_{\min}$) of the Pearson correlation matrix indicates that feature vectors span more independent directions, leading to richer and more informative representations. In contrast, a smaller $\lambda_{\min}$ implies that some directions vanish (near-zero variance), reflecting feature collapse or redundancy and resulting in degraded performance. In other words, $\lambda_{\min}(P)$ is proportional to the degree of feature diversity and independence, where higher diversity correlates with better classification accuracy, stronger generalization, and more stable training in ViTs.

Furthermore, during the experiments, we observed that the Pearson correlation matrix of the feature map $P(X)$ can be approximated by the Gram matrix (see the explanation in the Supp for details). Similar phenomena have been observed and proven for convolutional neural networks in [27]. The relationship between the minimum eigenvalue of the Gram matrix and the training convergence rate of the over-parameterized network has been studied and proven by [15, 16, 21, 31, 37].

Therefore, the minimum eigenvalue of the Pearson correlation matrix can serve as a quantitative proxy for representation diversity and as an indicator of ViT performance, linking representation quality to generalization behavior.

F. Additional Results on Efficient ViT Architecture Design

F.1. AutoFormer Search Space

We use the previous proxies and our Token-Entropy proxies with reduced dimension $u = 1$ to search for the optimal subnet on the same set of 1,000 ViT architectures within 5-7 parameter ranges. As shown in the table 3, our proposed proxies successfully identify the optimal subnet, improving accuracy by 0.11% compared to previous proxies across the 1,000 ViT architectures in the 5–7M ranges, respectively. Moreover, our proxies achieve an 87.5%–98.6% reduction in computation time when searching over 1,000 ViT architectures, requiring only 0.06 hours compared to 0.48 – 4.3 hours for previous proxies in the 5–7M range.

Table 5. Results on ImageNet when using different proxies to search for the optimal net in 1000 ViT architectures range 5-7M (above) and 15-19M (below)

Proxy	Param(M) ↓	FLOPs(B) ↓	Top-1 ↑	Time(h) ↓
SNIP [30]	6.9	1.7	74.89	1.00
GraSP [50]	5.7	1.4	74.47	1.38
TE-score [7]	5.9	1.5	74.60	4.30
NASWOT [36]	6.9	1.7	74.94	0.82
DSS [68]	6.9	1.6	75.29	0.60
AutoProxA [57]	6.9	1.5	75.32	0.48
DSS++ [69]	6.9	1.6	75.30	0.80
$M_1 + M_2$	**6.9**	**1.5**	**75.41**	**0.05**
$MHSA_{out} + M_1 + M_2$	**6.9**	**1.5**	**75.43**	**0.06**
$QKV + MHSA_{out} + M_1 + M_2$	**6.9**	**1.5**	**75.38**	**0.08**
SNIP [30]	17.8	3.8	80.30	1.08
GraSP [50]	18.1	3.9	80.35	1.40
TE-score [7]	17.9	3.8	80.38	4.44
NASWOT [36]	17.8	3.8	80.37	0.88
DSS [68]	18.0	3.9	80.39	0.74
AutoProxA [57]	17.7	3.9	80.20	0.56
DSS++ [69]	18.2	3.9	80.41	0.84
$M_1 + M_2$	**18.3**	**3.9**	**80.58**	**0.06**
$MHSA_{out} + M_1 + M_2$	**18.1**	**3.8**	**80.74**	**0.07**
$QKV + MHSA_{out} + M_1 + M_2$	**18.3**	**4.0**	**80.50**	**0.07**

We use previous proxies and our Token-Entropy proxies with reduced dimension $u = 1$ to search for the optimal architecture on the same set of 1,000 ViT architectures within two parameter ranges: 5–7M and 15–19M, with results presented in Table 3. As shown, our proposed proxies successfully identify the optimal architecture, improving accuracy by 0.11%-0.33% compared to previous proxies across the 1,000 ViT architectures in the 5–7M and 15–19M ranges, respectively. Moreover, our proxies achieve an 87.5–98.6% reduction in computation time when searching over 1,000 architectures, requiring only 0.06 hours compared to 0.48–4.3 hours for previous proxies in the 5–7M range, and 0.07 hours compared to 0.56–4.44 hours for previous proxies in the 15–19M range.

Using the ImageNet-1K [11], we employ the Token-Entropy proxy $MHSA_{out} + M_1 + M_2$ with $u = 8$ to search for the optimal architecture within the AutoFormer search space across tiny, small, and base parameter ranges, and then retrain the three resulting architectures. We compare our method with both ViT architecture search approaches [4, 5, 43, 57, 68, 69] and manually designed ViT

Table 6. Results of retraining the optimal architecture searched within the AutoFormer search space.

Model	Params ↓	FLOPs ↓	Top-1 ↑	Top-5 ↑	Type	Days
Deit-Ti [48]	5.7M	1.2B	72.2	91.1	Manual	-
TNT-Ti [20]	6.1M	1.4B	73.9	91.9	Manual	-
ViT-Ti [14]	5.7M	-	74.5	-	Manual	-
CPVT-Ti [9]	6.0M	-	74.9	92.6	Manual	-
PVT-Tiny [55]	13.2M	1.9B	75.1	-	Manual	-
ViTAS-C [43]	5.6M	1.3B	74.7	91.6	Auto	32
GLiT-Ti [4]	7.2M	1.4B	76.3	-	Auto	-
AutoFormer-Ti [5]	5.7M	1.3B	74.7	92.6	Auto	24
TF-TAS-Ti [68]	5.9M	1.4B	75.3	92.8	Auto	0.5
AutoProxA [57]	6.4M	-	75.6	-	Auto	0.4
T-Razor-Ti [69]	5.9M	1.4B	75.5	92.9	Auto	0.4
$MHSA_{out}$ + M_1 + M_2 - Ti	**6.1M**	**1.4B**	**76.5**	**93.3**	Auto	**0.33**
DeiT-S [48]	22.1M	4.7B	79.9	95.0	Manual	-
ViT-S/16 [14]	22.1M	4.7B	78.8	-	Manual	-
PVT-Small [55]	24.5M	3.8B	79.8	-	Manual	-
Swin-T [35]	29.0M	4.5B	81.3	-	Manual	-
TNT-S [20]	23.8M	5.2B	81.5	95.7	Manual	-
CPVT-S [9]	23.0M	-	81.5	95.7	Manual	-
T2T-ViT_t-14 [64]	21.5M	-	81.7	-	Manual	-
ViTAS-F [43]	27.6M	6.0B	80.5	95.1	Auto	32
GLiT-S [4]	24.6M	4.4B	80.5	-	Auto	-
AutoFormer-S [5]	22.9M	5.1B	81.7	95.7	Auto	24
TF-TAS-S [68]	22.8M	5.0B	81.9	95.8	Auto	0.5
T-Razor-S [69]	22.3M	5.1B	82.2	95.9	Auto	0.4
$MHSA_{out}$ + M_1 + M_2 - S	**22.9M**	**4.9B**	**83.0**	**95.9**	Auto	**0.17**
ViT-B/16 [14]	86.0M	18.0B	79.7	-	Manual	-
PVT-Large [55]	61.0M	9.8B	81.7	-	Manual	-
DeiT-B [48]	86.0M	18.0B	81.8	95.6	Manual	-
CPVT-B [9]	88.0M	-	82.3	-	Manual	-
TNT-B [20]	65.5M	14.1B	82.9	96.3	Manual	-
Swin-B [35]	88.0M	15.4B	83.5	-	Manual	-
T2T-ViT-24 [64]	64.1M	-	82.6	-	Manual	-
GLiT-B [4]	96.0M	17.0B	82.3	-	Auto	-
AutoFormer-B [5]	54.0M	11.0B	82.4	95.7	Auto	24
TF-TAS-B [68]	54.0M	12.0B	82.2	95.6	Auto	0.5
T-Razor-B [69]	53.8M	11.6B	82.3	95.6	Auto	0.4
$MHSA_{out}$ + M_1 + M_2 - B	**53.9M**	**11.4B**	**83.1**	**95.8**	Auto	**0.08**

models [9, 14, 20, 35, 48, 55, 64], and report the results in Table 6. As can be seen, for each parameter range, the architecture searched using the Token-Entropy proxy $MHSA_{out} + M_1 + M_2$ with $u = 8$ achieves a 1-3% improvement in Top-1 accuracy over manually designed ViT architectures. Furthermore, our architecture improves Top-1 accuracy by 0.7–1% while maintaining a comparable number of parameters and FLOPs to those obtained in [5, 57, 68, 69]. Additionally, our method significantly reduces both training and search time compared to traditional ViT NAS approaches [4, 5, 43] and prior ViT proxy methods [57, 68, 69]. While previous ViT NAS methods require 24–32 GPU days for the search, our approach reduces this to only 0.08–0.33 GPU days. These results demonstrate the effectiveness and efficiency of our proposed proxy in improving the performance of ViT architectures. Details of the evolutionary search and experimental setup are provided in the K.

Transfer learning. We fine-tune the searched architecture $MHSA_{out} + M_1 + M_2$ - S at a resolution of 384×384 on CIFAR-10 and CIFAR-100 [28]. It achieves transfer learning performance comparable to AutoFormer-S [5] and TF-TAS-S [68] (see results and details in the Supp).

F.2. PiT Search Space

We build the search space based on PiT [24] and evaluate our proposed proxy across various ViT architectures. We use token entropy $MHSA_{out} + M_1 + M_2$ with $u = 8$ to search for the optimal architecture in the PiT search space, and then train this architecture on ImageNet-1K [11] to obtain its final performance. As can be seen in table 7, under comparable parameters and FLOPs, our proxy successfully identifies an optimal ViT architecture that improves Top-1 accuracy by 1.1%. These results clearly demonstrate that our proxy is robust and generalizes well across different ViT architectures.

Details of the PiT search space and experimental setup are provided in the K

Table 7. Results of retraining the optimal architecture searched within the PiT search space.

Model	Param(M) ↓	FLOPs(B) ↓	Top-1 ↑	Top-5 ↑
PiT-Ti [24]	4.9	0.7	73.8	91.7
$PiT - Ti_{rand}$ [24]	4.9	0.7	69.7	89.1
TF-TAS-Ti [68]	4.6	0.6	73.7	91.7
T-Razor-Ti [69]	4.9	0.7	74.2	92.0
$MHSA_{out} + M_1 + M_2$	**4.6**	**0.6**	**75.3**	**92.2**
PiT-XS [24]	10.6	1.4	78.2	94.0
$PiT - Ti_{rand}$ [24]	10.5	1.8	74.8	92.2
TF-TAS-XS [68]	10.0	1.8	77.7	93.8
T-Razor-XS [69]	10.1	1.8	78.0	94.0
$MHSA_{out} + M_1 + M_2$	**10.1**	**0.6**	**78.3**	**93.7**

G. Metrics

Suppose we have n ViT architectures $\{arch_i\}_{i=1,\dots,n}$. Each architecture has a ground-truth performance, which is the test accuracy, denoted by $\{g_i\}_{i=1,\dots,n}$. The proxy values of the n ViT architectures are denoted by $\{v_i\}_{i=1,\dots,n}$. The ranking of the ground-truth performance is $\{r_i\}_{i=1,\dots,n}$ with $r_i \in \{1, \dots, n\}$. The ranking of the proxy values is $\{s_i\}_{i=1,\dots,n}$ with $s_i \in \{1, \dots, n\}$.

The Spearman ρ correlation is the correlation between the two ranking variables and is defined as:

$$\rho = \frac{corr(r, s)}{\sqrt{corr(r, r)corr(s, s)}} \tag{8}$$

Spearman ρ measures the monotonic relationship between two variables based on their ranked values, not their raw values.

The Kendall τ correlation measures the relative difference between concordant and discordant pairs and is defined as:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \text{sgn}(g_i - g_j)\,\text{sgn}(v_i - v_j) \tag{9}$$

The values of both Spearman's ρ and Kendall's τ range between -1 and 1.

- 1: The two rankings are the same
- -1: One ranking is the reverse of the other
- 0: No relationship

A higher correlation ranking indicates that the proxy is more strongly correlated with the test accuracy, suggesting that the proxy can serve as a reliable indicator of ViT performance.

H. Relationship Between the Gram Matrix and the Pearson Correlation Matrix

Suppose we have a feature matrix $X \in \mathbb{R}^{n\times d}$. We will explain the relationship between the Gram matrix and the Pearson correlation for the embedding case below, where each column is considered as a variable. In the token space, we have a similar explanation where each row is considered as a variable.

Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_d]$, $\mathbf{x}_i = [x_{1i}, x_{2i}, \ldots, x_{ni}]^\top$. The Gram matrix $G \in \mathbb{R}^{d\times d}$ is defined as:

$$G = X^\top X \tag{10}$$

with entries

$$G_{ij} = \mathbf{x}_i^\top \mathbf{x}_j = \sum_{k=1}^{n} x_{ki} x_{kj}, \tag{11}$$

The sample mean of $\mathbf{x}_i$ is:

$$\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{ki}, \tag{12}$$

and the centered values are defined as:

$$\tilde{x}_{ki} = x_{ki} - \bar{x}_i. \tag{13}$$

The sample standard deviation of $\mathbf{x}_i$ is

$$s_i = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n} \tilde{x}_{ki}^2}. \tag{14}$$

The standardized feature is obtained by subtracting its mean and dividing by its standard deviation, resulting in a vector with zero mean and unit variance.

$$\hat{x}_{ki} = \frac{x_{ki} - \bar{x}_i}{s_i} = \frac{\tilde{x}_{ki}}{s_i} \tag{15}$$

The Pearson correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$r_{ij} = \frac{\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{(n-1)\, s_i s_j} = \frac{1}{n-1}\sum_{k=1}^{n} \hat{x}_{ki}\, \hat{x}_{kj}. \tag{16}$$

Thus, the Pearson correlation is exactly the (i, j) entry of the Gram matrix of standardized features, scaled by $\frac{1}{n-1}$:

$$r_{ij} = \frac{1}{n-1}\langle \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j \rangle \tag{17}$$

where $\hat{\mathbf{x}}_i = [\hat{x}_{1i}, \ldots, \hat{x}_{ni}]^\top$.

Therefore, the Gram matrix of standardized features equals $(n-1)$ times the Pearson correlation matrix.

Each entry r_{ij} measures the linear correlation between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, taking values in the range $[-1, 1]$, where $r_{ij} = 1$ indicates perfect positive correlation, $r_{ij} = -1$ indicates perfect negative correlation, and $r_{ij} = 0$ indicates no linear correlation.

I. Linear Modules in Vision Transformers

QKV. Let $X \in \mathbb{R}^{t_{\text{dim}}\times e_{dim}}$. At layer l and head i, linear transforms are applied to obtain the matrices Q_i, K_i, and V_i, as defined by the equations below:

$$Q_i = XW_i^Q; \quad K_i = XW_i^K; \quad V_i = XW_i^V \tag{18}$$

where $i \in \overline{1, h_l}$, and W_i^Q, $W_i^K \in \mathbb{R}^{e_{dim}\times d_k}$, $W_i^V \in \mathbb{R}^{e_{dim}\times d_v}$.

Additionally, $d_k = d_v = \frac{e_{dim}}{h_l}$, where h_l represents the number of heads at layer l.

Observation. The Q, K, V matrix is achieved by multiplying the input matrix X with weight matrix W to obtain the linear transformation of the input.

Attention map. Suppose we have matrix H as the matrix obtained after concatenating each matrix head i, where A is the attention map. The ViT model using a linear transformation W^O to H, we have $A = HW^O$.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W^O \tag{19}$$

where

$$\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V) \tag{20}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{21}$$

Observation. The attention map can be achieved by linear transformation on the matrix H.

Multi-layer Perceptron. We denote that A is the output feature map before the Multi-layer Perceptron (MLP) component in the Vision Transformer. M_1 and M_2 represents the output of the first and second linear layer in the MLP component, with W_1 and W_2 as the weight matrices of the fully connected layers in the MLP component, and b_1 and b_2 as the bias vectors of the fully connected layers. The activation function is GELU$(\cdot)$

$$M_1 = A{*}W_1{+}b_1; M_2 = GELU(M_1){*}W_2{+}b_2 = C{*}W_2{+}b_2 \tag{22}$$

Observation. M_1 and M_2 can be achieved by a linear module of the previous feature maps.

J. Experimental Setup and Additional Results for ViT Feature Analysis

J.1. Token Space and Embedding Space

We provide an illustration of token space and embedding space in Figure 3. In the token space, each row in the feature map can be considered a variable. In contrast, in the embedding space, each column is treated as a variable for information extraction.

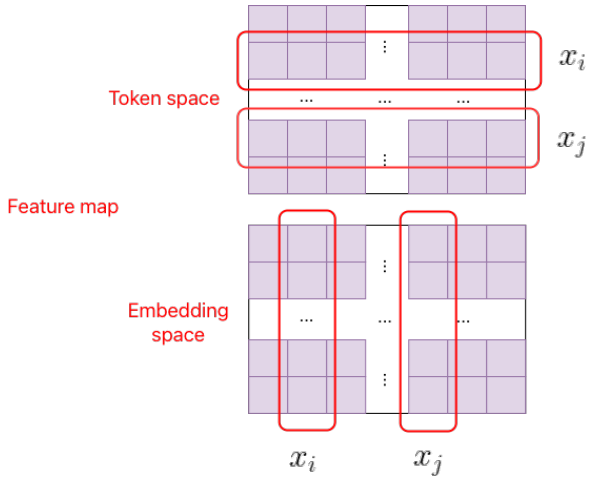

Figure 3. Illustration of features in the token and embedding spaces

J.2. Feature Collapse and Singular Value Analysis

Singular value decomposition analysis. We selected the architecture with the highest accuracy in our API to analyze the collapse of feature information. The ViT architecture was evaluated without pretrained weights to extract feature representations, and one data sample was selected for analysis in both the token and embedding spaces. Specifically, for the feature type Q, we normalized the features, computed the covariance matrix of the feature map, and then took the logarithm of its singular values. We subsequently calculated the mean and standard deviation of these singular values across all layers of the architecture.

As shown in Figure 2, the left panel presents the logarithmic singular value spectrum of the feature Q in the token space, while the right panel shows the corresponding spectrum in the embedding space. It can be observed that, for most components, the singular values are close to zero and their logarithms are negative, with only one component having a logarithmic singular value greater than zero. This indicates that the feature map Q is a low-rank matrix, suggesting that the ViT architecture tends to map many different inputs into nearly the same subspace, revealing a clear feature collapse phenomenon.

We further compute the singular value spectrum for each feature type separately, and observe a consistent pattern across all cases: the feature matrices remain low-rank in both token and embedding spaces. These findings explain why our proposed reduction scheme produces an extremely compact feature representation while retaining essential information.

Suppose we have a feature matrix $X \in \mathbb{R}^{n\times d}$. In embedding space, we have $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$, where $\mathbf{x_i} \in \mathbb{R}^d$ is one sample.

We compute the covariance matrix as follows:

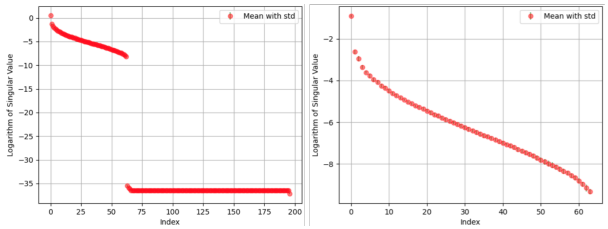

Figure 4. Logarithm of the singular value spectrum of the feature K in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^\top \tag{23}$$

where

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i} \tag{24}$$

Next, we compute the singular values via singular value decomposition:

$$C = USV^\top \tag{25}$$

$S \in \mathbb{R}^{d\times d}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. Finally, we obtain the logarithmic values $\log(\sigma_j)$.

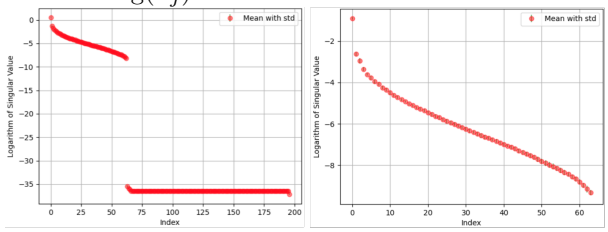

Figure 5. Logarithm of the singular value spectrum of the feature V in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

Figure 4 shows the singular values of the feature map K in ViT, in token space and embedding space, respectively. Figures 5 present the singular values of the feature map V in token space and embedding space. Figures 6 display the singular values of the feature map $MHSA_{out}$ in both spaces. Figures 7 and Figures 8 illustrate the singular values of the feature maps M_1 and M_2 in token space and embedding space, respectively.

The singular value spectrum of the covariance matrix of each feature map illustrates that most feature rows and

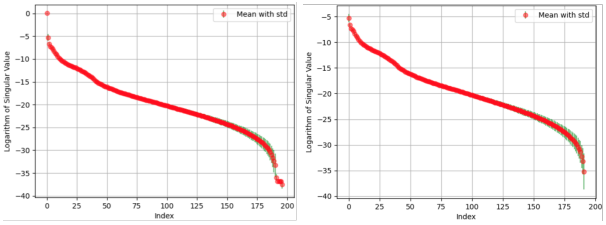

Figure 6. Logarithm of the singular value spectrum of the feature $MHSA_{out}$ in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

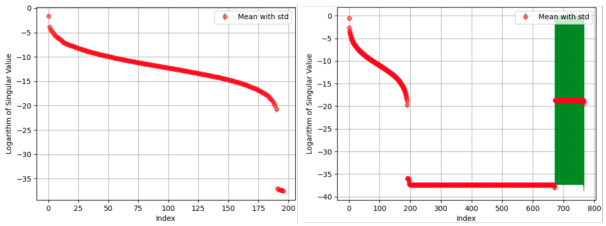

Figure 7. Logarithm of the singular value spectrum of the feature M_1 in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

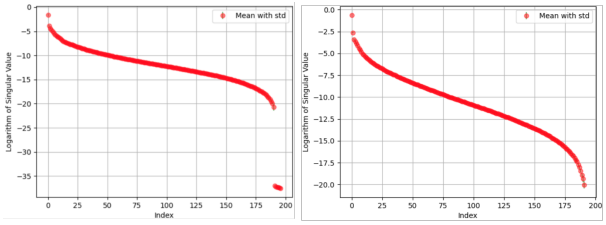

Figure 8. Logarithm of the singular value spectrum of the feature M_2 in the token space (left) and embedding space (right). The x-axis denotes variable indices, and the y-axis shows the mean and standard deviation of the logarithmic singular values across all layers. **Most singular values are near zero, indicating a low-rank structure and feature collapse.**

columns are dependent, as the logarithm of their singular values is significantly less than zero. For example, Figure 4 in the left shows only one logarithm of singular value slightly greater than zero. Similarly, Figure 4 in the right contains one value near zero, while the remaining logarithmic singular values are highly negative. These observations indicate that most feature maps are low-rank matrices in both the token and embedding spaces. Consequently, we can compact the feature map by selecting only the most informative rows or columns for feature quantification.

This finding is particularly important because the original feature maps often contain hundreds to thousands of rows or columns, making computations over the full feature map computationally expensive. Moreover, when using the minimum eigenvalue as a proxy for information, the method assumes all variables are independent. The singular value spectrum enables us to identify and remove dependent rows or columns, resulting in a more compact representation and more effective capture of essential information in the feature map.

J.3. Evaluation Results Across Various ViT Architecture Sets Using Different Random Seeds

We sample different sets of ViT architectures with different random seeds to assess the sensitivity of our proposed methods. We compute the Spearman's ρ and Kendall's τ correlation rankings between the proxies and test accuracy for three sets of 1,000 ViT architectures with seeds 0, 10, and 20. The results are presented in Table 8, Table 9, and Table 10.

Table 8. Spearman's ρ and Kendall's τ correlation rankings for three sets of 1,000 ViT architectures with parameter ranges of 5–7M, 15–19M, and 45–47M, **when seed = 0**, comparing the top-3 Token-Entropy proxies ($u = 1$) with baselines. **Our proposed proxies significantly improve the correlation rankings, demonstrating that they are strong indicators of ViT performance.**

Proxy	5-7M		15-19M		45-47M	
	ρ	τ	ρ	τ	ρ	τ
SNIP [30]	0.313	0.207	0.280	0.190	0.056	0.037
GraSP [50]	-0.101	-0.066	-0.064	-0.043	0.023	0.015
TE-score [7]	-0.319	-0.219	-0.084	-0.057	-0.106	-0.072
NASWOT [36]	0.382	0.278	0.232	0.162	0.243	0.171
DSS [68]	0.622	0.439	0.468	0.315	-0.119	-0.079
AutoProxA [57]	0.675	0.477	0.446	0.299	-0.126	-0.084
DSS++ [69]	0.638	0.448	0.450	0.302	-0.140	-0.094
$M_1 + M_2$	**0.849**	**0.665**	**0.718**	**0.529**	**0.368**	**0.251**
$MHSA_{out} + M_1 + M_2$	**0.851**	**0.669**	**0.717**	**0.529**	**0.360**	**0.246**
$QKV + MHSA_{out} + M_1 + M_2$	**0.856**	**0.676**	**0.720**	**0.533**	**0.350**	**0.239**

Table 9. Spearman's ρ and Kendall's τ correlation rankings for three sets of 1,000 ViT architectures with parameter ranges of 5–7M, 15–19M, and 45–47M, **when seed = 10**, comparing the top-3 Token-Entropy proxies ($u = 1$) with baselines. **Our proposed proxies significantly improve the correlation rankings, demonstrating that they are strong indicators of ViT performance.**

Proxy	5-7M		15-19M		45-47M	
	ρ	τ	ρ	τ	ρ	τ
SNIP [30]	0.381	0.252	0.263	0.173	0.027	0.019
GraSP [50]	-0.091	-0.061	-0.079	-0.053	-0.078	-0.052
TE-score [7]	-0.287	-0.188	0.037	0.026	-0.052	-0.034
NASWOT [36]	0.426	0.314	0.214	0.145	0.246	0.174
DSS [68]	0.671	0.478	0.439	0.297	-0.119	-0.079
AutoProxA [57]	0.720	0.516	0.411	0.277	-0.141	-0.094
DSS++ [69]	0.685	0.487	0.419	0.283	-0.142	-0.094
$M_1 + M_2$	**0.871**	**0.687**	**0.689**	**0.510**	**0.412**	**0.279**
$MHSA_{out} + M_1 + M_2$	**0.874**	**0.692**	**0.687**	**0.509**	**0.407**	**0.276**
$QKV + MHSA_{out} + M_1 + M_2$	**0.864**	**0.676**	**0.691**	**0.514**	**0.402**	**0.272**

As can be seen, with different seeds and different sets of ViT architectures, the correlation values vary slightly. However, the overall trend and the relative performance gap between our proxies and previous proxies remain consistent. Specifically, when the seed is 0, our proxies im-

Table 10. Spearman's ρ and Kendall's τ correlation rankings for three sets of 1,000 ViT architectures with parameter ranges of 5–7M, 15–19M, and 45–47M, **when seed = 20**, comparing the top-3 Token-Entropy proxies ($u = 1$) with baselines. **Our proposed proxies significantly improve the correlation rankings, demonstrating that they are strong indicators of ViT performance.**

Proxy	5-7M		15-19M		45-47M	
	ρ	τ	ρ	τ	ρ	τ
SNIP [30]	0.332	0.220	0.298	0.208	0.065	0.043
GraSP [50]	-0.061	-0.041	-0.131	-0.087	-0.012	-0.007
TE-score [7]	-0.266	-0.173	0.077	0.052	-0.146	-0.098
NASWOT [36]	0.401	0.290	0.254	0.183	0.291	0.223
DSS [68]	0.655	0.465	0.524	0.353	-0.118	-0.080
AutoProxA [57]	0.700	0.499	0.499	0.336	-0.161	-0.108
DSS++ [69]	0.667	0.472	0.506	0.304	-0.171	-0.115
$M_1 + M_2$	**0.854**	**0.670**	**0.744**	**0.556**	**0.467**	**0.323**
$MHSA_{out} + M_1 + M_2$	**0.855**	**0.672**	**0.743**	**0.555**	**0.462**	**0.320**
$QKV + MHSA_{out} + M_1 + M_2$	**0.844**	**0.658**	**0.748**	**0.560**	**0.453**	**0.314**

prove upon previous ViT proxies by 18%, 25%, and 48% for the 5–7M, 15–19M, and 45–47M parameter ranges, respectively. When the seed is 10, the improvements are 15.4%, 25%, and 53% for the 5–7M, 15–19M, and 45–47M ranges, respectively. When the seed is 20, the improvements are 15.5%, 22%, and 58.5% for the 5–7M, 15–19M, and 45–47M ranges, respectively.

These results demonstrate the stability and strong performance of our proxies across various ViT architectures and random seeds.

As the model size increases, overparameterization and complex training dynamics become dominant factors, generally reducing the predictive power of proxy metrics. Specifically, for large architectures with 45–47M parameters, the previous ViT proxies show near-zero or even negative correlations, indicating that they fail to provide meaningful predictions. In contrast, our proposed proxies maintain higher correlation rankings under these challenging conditions, demonstrating their robustness and effectiveness as a performance indicator.

J.4. Influence of Feature Axes

We set the dimension reduction parameter $u = 1$ and compute the Spearman correlation between each type of our proxies and the test accuracy across 1,000 ViT architectures (5-7M) in both token and embedding spaces.

We compare the separated ViT Entropy proxies computed in the token space and embedding space, as shown in Figure 9. In the token space, V' and $MHSA_{out}$ improve the Spearman correlation ranking by 15%, while M_1 and M_2 yield a 10% improvement. For the remaining ViT Feature Entropy proxies, the correlation in the token space improves only slightly compared to the embedding space, for example, by as little as 0.01%. For proxies related to skip connections and layer normalization, the Spearman correlation remains approximately zero in both token and embedding spaces.

The comparison of combined ViT Entropy proxies in the token and embedding spaces is shown in Figure 10. As illustrated, the proxies $M_1 + M_2$, $MHSA_{out} + M_1 + M_2$, and $QKV + MHSA_{out} + M_1 + M_2$ in the token space yield a 10% higher correlation than their counterparts in the embedding space. Additionally, the QKV proxy in the token space improves correlation by 5% compared to the embedding space. The proxies $QKV + M_1 + M_2$ and $Q + K + V$ achieve identical correlation in both spaces. The combination of skip connection proxies results in the same negative correlation in both token and embedding spaces. As can be seen, ViT Entropy proxies in the token space better represent the ViT generalization predictor compared to those in the embedding space.

We compare the separate ViT Minimum Eigenvalue proxies in the token space and embedding space in Figure 11. As shown, V' and $MHSA_{out}$ in the token space increase the correlation by 10% compared to their counterparts in the embedding space. The proxies Q, K, V, and M_1 achieve the same correlation in both spaces. Notably, proxies related to skip connections and layer normalization show a 20% improvement in correlation when computed in the token space. Figure 12 further demonstrates that the combined ViT Minimum Eigenvalue proxies in token space improve correlation by 5% compared to those in the embedding space.

J.5. On the Impact of Dimension Reduction

Table 11 presents the experimental results for the Entropy Proxies evaluated using different values of $u = 1, 4, 8, 16$. These results illustrate how varying u affects the proxy's behavior and its sensitivity to the underlying token and embedding spaces. Similarly, Table 12 reports the corresponding results for the Minimum Eigenvalue Proxies, also evaluated with $u = 1, 4, 8, 16$, allowing for a direct comparison of how each proxy responds to changes in u. Together, these tables provide a comprehensive view of the performance and characteristics of both proxies across the selected range of u values.

We compare the correlation rankings of the Token-Entropy proxy under different dimensionality reductions with $u = 1, 4, 8, 16$ in Figure 13 and Figure 14. As can be seen, across different values of $u = 1, 4, 8, 16$, the proxies achieve similar correlation rankings.

Figures 15 and 16 compare the Entropy proxies in the embedding space under different reduced dimensions. For the separate Embedding-Entropy proxies, the correlation rankings remain the same across $u = 1, 4, 8, 16$. In the combined Embedding-Entropy proxies, $M_1 + M_2$ and $MHSA_{out} + M_1 + M_2$ with $u = 8$ improve the correlation by 10% compared to $u = 1, 4, 16$. The remaining proxies show consistent correlation rankings across all values of u.

We compare the Minimum Eigenvalue proxies in the token space across different reduced dimensions u in Fig-

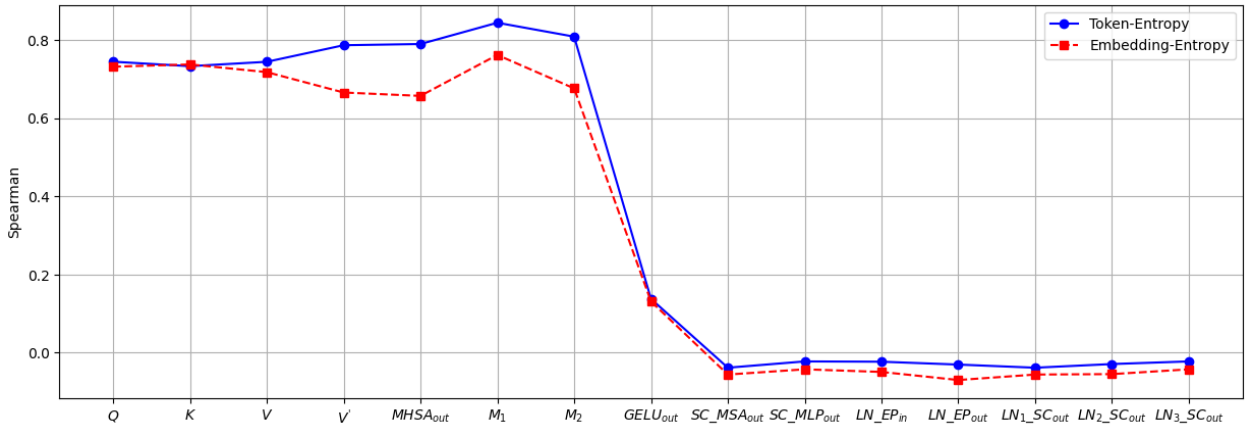

Figure 9. Comparison of Spearman's ρ correlation rankings of separate Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in token and embedding spaces. The blue line represents the results of proxies in token space, while the red line represents the results in embedding space. **Token space improves correlation rankings compared to embedding space, especially for meaningful indicators such as V', $MHSA_{out}$, M_1, and M_2.**

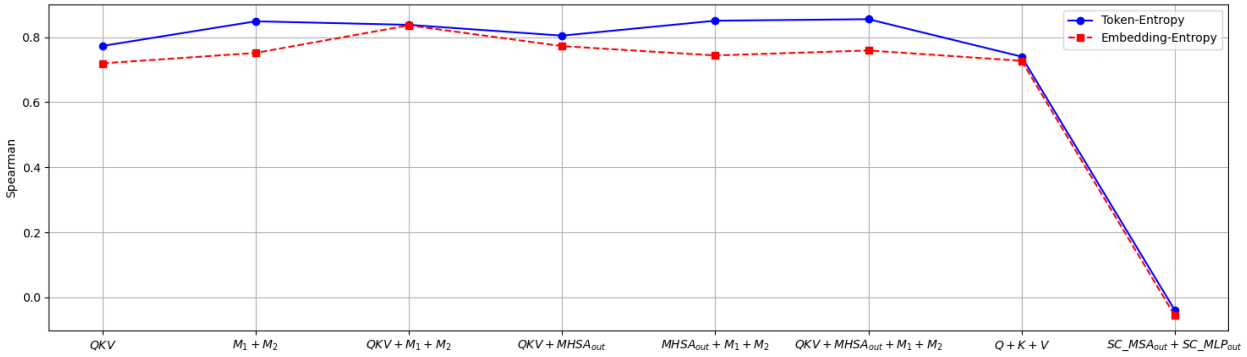

Figure 10. Comparison of Spearman's ρ correlation rankings of combined Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in token and embedding spaces. The blue line represents the results of proxies in token space, while the red line represents the results in embedding space. **Token space improves correlation rankings compared to embedding space by 5% to 10% for the predictors QKV, $M_1 + M_2$, $MHSA_{out} + M_1 + M_2$, and $QKV + MHSA_{out} + M_1 + M_2$.**

ure 17 and Figure 18. As shown, the proxies achieve the highest correlation when $u = 1$, improving the correlation by at least 10% compared to other dimensional reductions. For the K feature, $u = 1$ increases the correlation ranking by 30% compared to $u = 8$ and $u = 16$. Notably, for $MHSA_{out}$, $u = 1$ significantly improves the correlation ranking by 27.5% compared to $u = 16$. In the case of combined proxies, $u = 1$ improves the correlation by 5% to 10% compared to $u = 4$, 8, and 16.

Figures 19 and 20 show the differences in correlation rankings of the Minimum Eigenvalue proxies in the embedding space under varying reduced dimensions.

As can be seen, the features Q, K, V, and M_2 improve correlation by 10-15% when $u = 1$ compared to $u = 4$, 8, and 16. The features V', $MHSA_{out}$, and M_1 improve correlation rankings by 21-23% under the same conditions. For the combined feature proxies, $Q+K+V$ improves correlation ranking by 4% with $u = 1$ compared to $u = 4$, 8, and 16. Additionally, QKV, $QKV + M_1 + M_2$, $QKV + MHSA_{out}$, and $QKV + MHSA_{out} + M_1 + M_2$ improve correlation by 8-11.5% with $u = 1$ compared to $u = 4$, 8, and 16. Furthermore, M_1+M_2 and $MHSA_{out}+M_1+M_2$ improve correlation by 15-19% with $u = 1$ compared to $u = 4$, 8, and 16. However, features related to skip connections and layer normalization achieve similar correlations across all values of $u = 1$, 4, 8, and 16.

J.6. Comprehensive Results in Higher Parameter Ranges

Table 13 presents the correlation rankings of ViT Feature Entropy proxies when the token space is reduced to dimension $u = 1$. The results are based on 1,000 ViT sampled

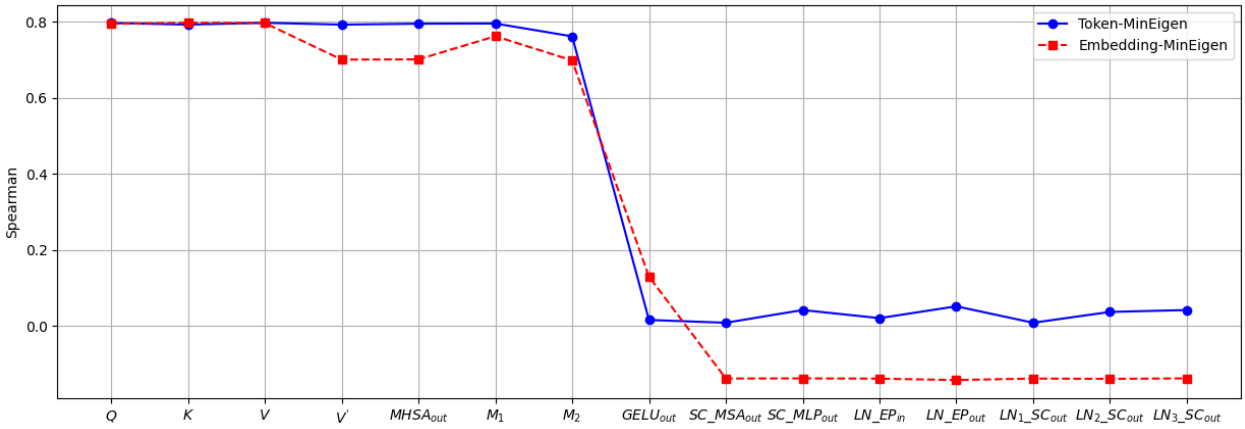

Figure 11. Comparison of Spearman's ρ correlation rankings of separate Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in token and embedding spaces. The blue line represents the results of proxies in token space, while the red line represents the results in embedding space. **Token space improves correlation rankings compared to embedding space, particularly for meaningful indicators such as V', $MHSA_{out}$, M_1, and M_2, with a maximum improvement of 10%.** For the proxies related to skip connections and layer normalization, the improvement reaches 20%.

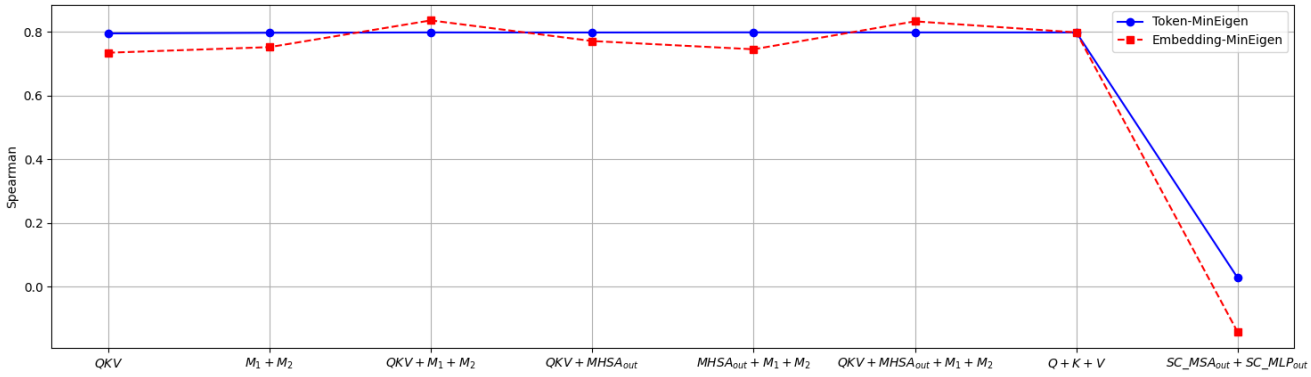

Figure 12. Comparison of Spearman's ρ correlation rankings of combined Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in token and embedding spaces. The blue line represents the results of proxies in token space, while the red line represents the results in embedding space. **Token space improves correlation rankings compared to embedding space by 5% for the predictors QKV, $M_1 + M_2$, and $MHSA_{out} + M_1 + M_2$.**

architectures from AutoFormer-Small with 15–19 million parameters. As shown, the feature proxies M_1, $M_1 + M_2$, $MHSA_{out} + M_1 + M_2$, and $QKV + MHSA_{out} + M_1 + M_2$ improve the correlation ranking by 25% compared to the baselines DSS, AutoProxA, and DSS++. The proxy $QKV + M_1 + M_2$ improves the correlation by 21%, while proxies Q, K, V, V', $MHSA_{out}$, M_2, QKV, $Q + K + V$, and $QKV + MHSA_{out}$ show improvements of around 10%. In contrast, proxies related to skip connections and layer normalization achieve lower correlation rankings than the baseline.

Table 13 shows the correlation rankings of the ViT Feature Entropy proxies under the same dimensionality reduction setting ($u = 1$), using 1,000 ViT sampled architectures from AutoFormer-Base with 45–47 million parameters. When the parameter count is extremely large, the baseline proxies DSS, AutoProxA, and DSS++ exhibit a significant drop in correlation, even resulting in negative values. In contrast, our Entropy proxies in the token space effectively mitigate this issue. As shown, features such as Q, K, V, M_1, M_2, $GELU_{out}$, SC_MSA_{out}, and $LN_1\_SC_{out}$ achieve positive correlation rankings. More importantly, the combined proxies M_1, $M_1 + M_2$, $QKV + M_1 + M_2$, $MHSA_{out} + M_1 + M_2$, and $QKV + MHSA_{out} + M_1 + M_2$ improve the correlation ranking by 30% to 45% compared to the baselines.

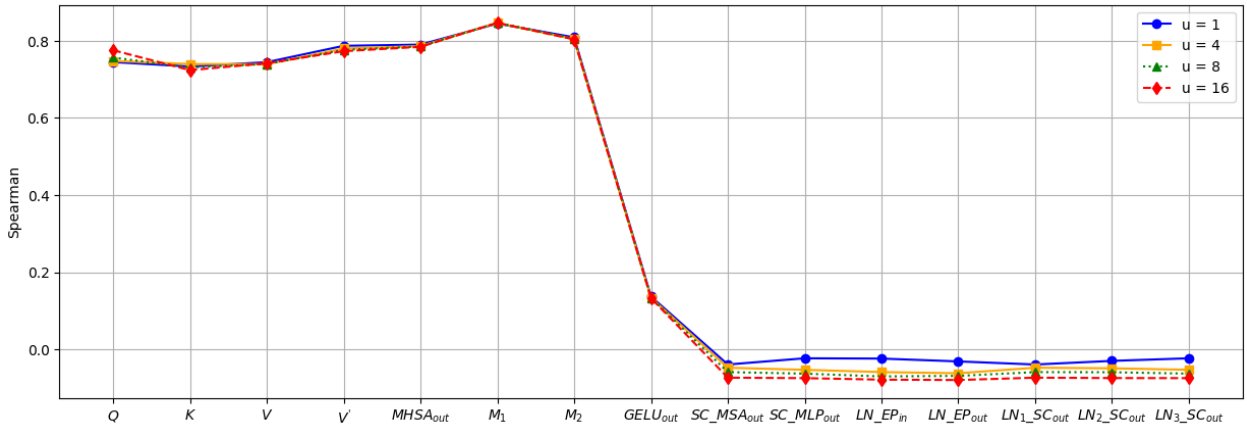

Figure 13. Comparison of Spearman's ρ correlation rankings of separate Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in token space with dimension reduction values $u = 1, 4, 8, 16$.

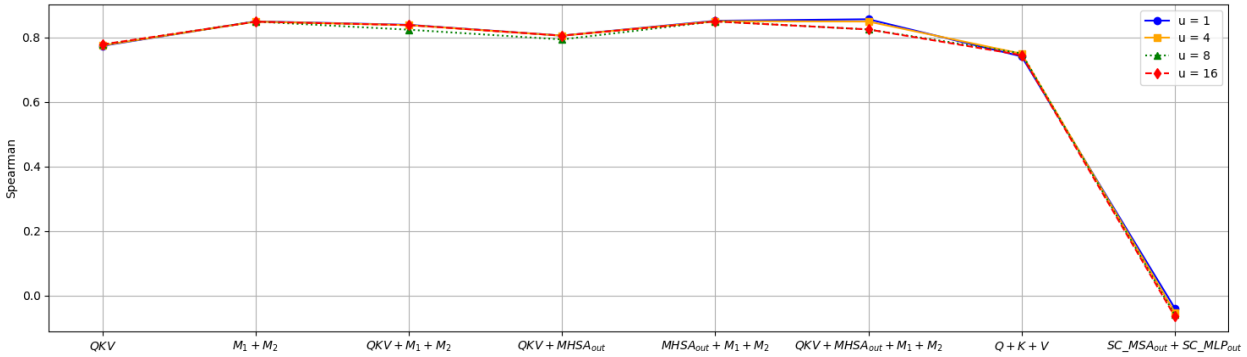

Figure 14. Comparison of Spearman's ρ correlation rankings of combined Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in token space for dimension reduction values $u = 1, 4, 8, 16$.

J.7. Experiments with Various Sampled Features

In our proposed method, we compute ViT feature proxies based on a single feature map corresponding to one sample in a forward batch. To assess the robustness and sensitivity of our method, we conduct three supporting experiments. In the first setup, we select the first feature map in a batch. In the second setup, we select another feature map, for example, feature map 10. In the third setup, we select a random feature map for each module when computing the proxy, meaning the chosen feature map index varies across modules within the same architecture. The results are compared in Table 14. As can be seen, our proxies achieve comparable performance across all three setups, indicating that the method is not dependent on a specific feature map and captures intrinsic properties of the model's representations rather than being sensitive to individual samples.

J.8. Data Independence and Robustness

Our proxies are independent of both labels and datasets, as they extract features from the ViT architecture at initialization without training or gradient updates. To further support this observation, we generate Gaussian features following a standard normal distribution and use them as inputs to the architecture. We then compare the correlation rankings of the optimal proxies obtained using Gaussian inputs with those obtained from the real ImageNet-1K dataset. As shown in Table 15, the Gaussian inputs achieve proxy performance comparable to that of real data.

To assess the robustness of our proposed feature proxies, we add Gaussian noise to the input data and compare the correlation rankings of the optimal proxies under original and noise-added inputs. As shown in Table 16, the correlations with noisy inputs remain comparable to those with the original data, demonstrating the robustness of our method.

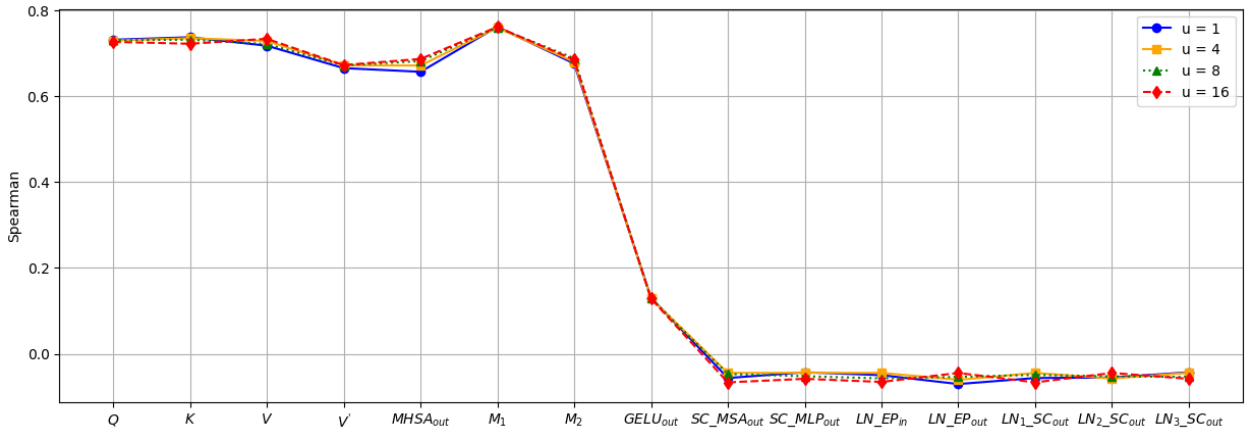

Figure 15. Comparison of Spearman's ρ correlation rankings of separate Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in embedding space with dimension reduction values $u = 1, 4, 8, 16$.

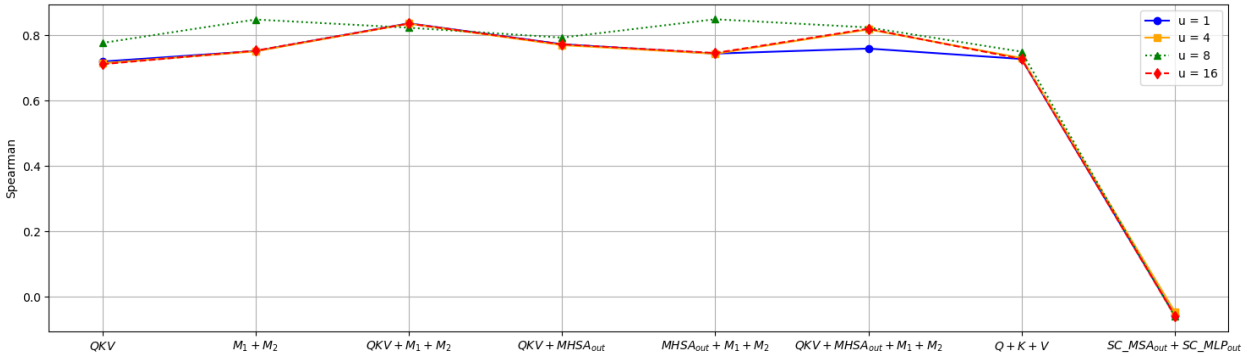

Figure 16. Comparison of Spearman's ρ correlation rankings of combined Entropy ViT feature proxies across 1,000 ViT architectures (5-7M) in embedding space for dimension reduction values $u = 1, 4, 8, 16$.

J.9. Evaluation on Out-of-Distribution Performance

We evaluate our proposed proxy on 1,000 ViT architectures (5–7M parameters) using out-of-distribution (OoD) accuracy. The ViT architectures inherit weights from the Supernet trained on ImageNet-1K [11] and are evaluated on the out-of-distribution ImageNet-C dataset [23] with Gaussian noise corruption at level 1. We then compute the correlation ranking between the proxy values and the OoD accuracy. The results are presented in Table 17.

As can be seen, although our method primarily focuses on in-distribution (ID) prediction, it achieves promising results and a correlation comparable to previous proxies on OoD accuracy. However, since ID and OoD accuracies are weakly correlated and our proxy is highly correlated with ID accuracy, we suggest that designing separate proxies specifically for OoD tasks may be more appropriate and could lead to higher correlation with OoD performance.

J.10. Experiments on Small Datasets Using Distillation Accuracy

We evaluate the robustness of our proposed proxies across different tasks, including knowledge distillation on small datasets.

AutoFormer search space. We sampled 100 ViT architectures from the AutoFormer search space and collected their distillation test accuracies on CIFAR-100 and Oxford Flowers from ViT-Bench-101 [57]. In their training setup [57], the distillation accuracy on each small dataset is obtained by training the student ViT architecture under the guidance of a ResNet-50 teacher with low-resolution input (32×32). The loss used during distillation combines the standard supervised loss computed on labeled data with a distillation loss to obtain the final distillation accuracy for each dataset.

The ground truth accuracy on CIFAR-100 and Flowers corresponds to the distillation accuracy, which is influenced by various factors, including the teacher model, the loss

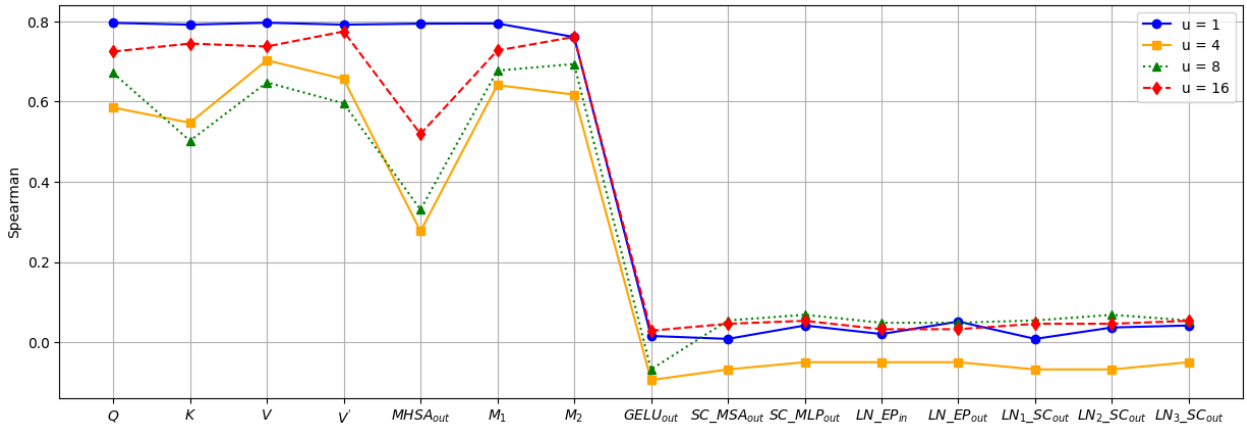

Figure 17. Comparison of Spearman's ρ correlation rankings of separate Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in token space for dimension reduction values $u = 1, 4, 8, 16$. **When $u = 1$, the correlation improves significantly, particularly for $MHSA_{out}$, reaching an improvement of up to 27.5% compared to $u = 16$.**

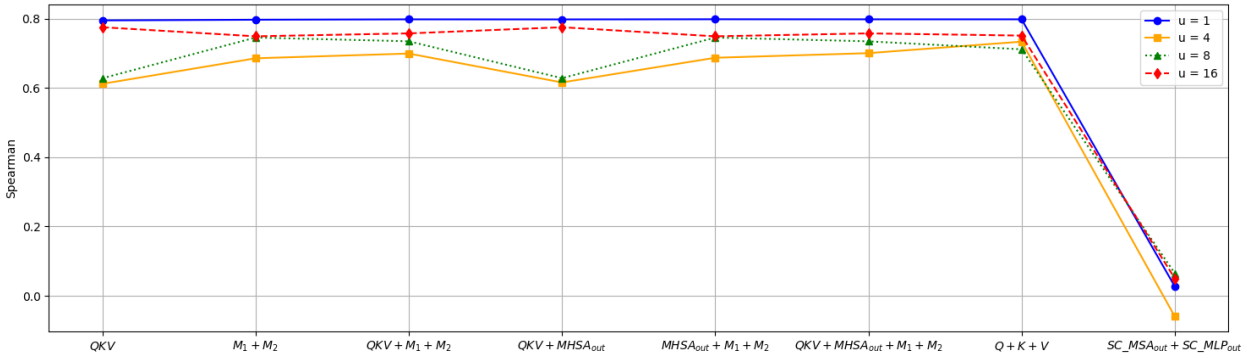

Figure 18. Comparison of Spearman's ρ correlation rankings of combined Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in token space for dimension reduction values $u = 1, 4, 8, 16$. **With $u = 1$, the correlation ranking improves by 5% to 10% for each meaningful indicator compared to $u = 4, 8, 16$.**

function, and the distillation training process. Additionally, these datasets have fewer classes and lower resolution compared to ImageNet-1K.

We compute the correlation rankings of the top-2 ViT features in both token and embedding spaces using entropy and minimum eigenvalue (with $u = 1$), and compare them with the CIFAR-100 distillation accuracy. The results are presented in Table 18. As shown, our proxies strongly correlate with the distillation accuracy of ViT architectures on CIFAR-100. Among them, the token-entropy proxies achieve the highest correlation. Using the same entropy measure, the embedding space shows slightly lower correlation. However, when evaluated with the minimum eigenvalue measure, the embedding space achieves higher correlation than the token space.

We select the top-2 proxies from Table 18 and compare them with previous ViT proxies in Table 19. As shown, our proxies achieve a 14.3% improvement in correlation ranking compared to the baseline. These results suggest that our proxies are useful indicators for ViT distillation training on CIFAR-100.

Similarly, for the Flowers dataset, we compute the Spearman correlation between the top-2 ViT features in both token and embedding spaces using entropy and minimum eigenvalue ($u = 1$) proxies, and compare them with the distillation accuracy across 100 ViT architectures. Table 20 shows that our proxies are strongly correlated with Flowers distillation accuracy. In Flowers, however, both the entropy and minimum eigenvalue proxies in the embedding space achieve higher correlation rankings than in the token space. This difference can be attributed to the characteristics of the dataset: Flowers has fewer classes and higher intra-class variability, making global embedding-level representations more informative than local token-level features. Consequently, ViT distillation performance is also affected by these dataset characteristics.

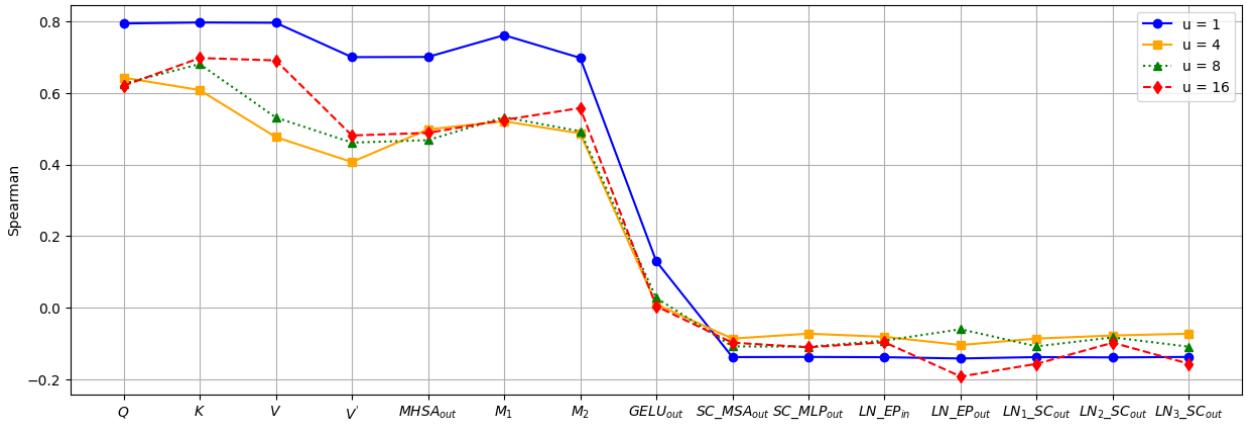

Figure 19. Comparison of Spearman's ρ correlation rankings of separate Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in embedding space for dimension reduction values $u = 1, 4, 8, 16$. **When $u = 1$, the correlation improves significantly, particularly for $MHSA_{out}$, reaching an improvement of up to 27.5% compared to $u = 16$.**

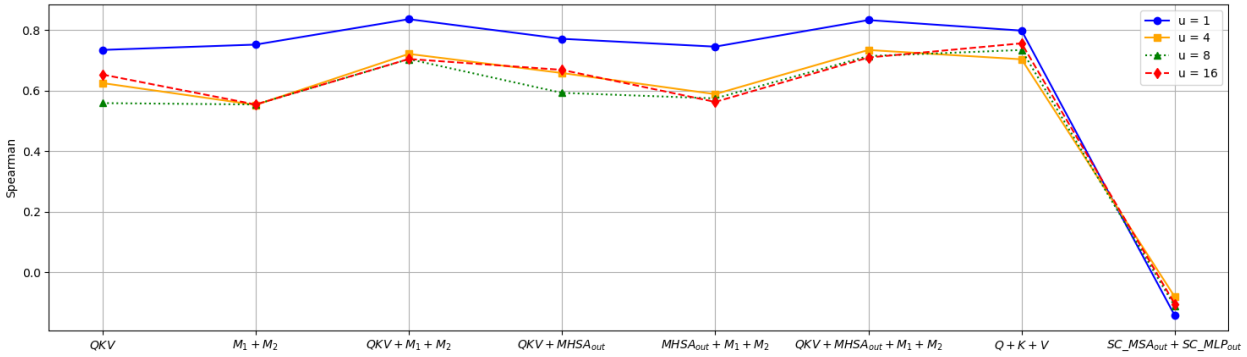

Figure 20. Comparison of Spearman's ρ correlation rankings of combined Minimum Eigenvalue ViT feature proxies across 1,000 ViT architectures (5-7M) in embedding space for dimension reduction values $u = 1, 4, 8, 16$. **For each meaningful indicator, the correlation ranking with $u = 1$ improves, with a maximum improvement of 19% (for $M_1 + M_2$) compared to $u = 4$, 8, and 16.**

We select the top-2 proxies from Table 20 and compare them with previous ViT proxies in Table 21 across 100 ViT architectures - AutoFormer. As shown, our proxies achieve a 4.3% improvement in correlation ranking compared to the previous proxies. These results demonstrate that our proxies are strong predictors of ViT performance in distillation training on the Flowers dataset.

PiT search space. We sample 100 ViT architectures from the PiT search space and collect their distillation accuracies on three datasets: CIFAR-100, Flowers, and Chaoyang, from ViT-Bench-101 [57]. In their setup, for each small dataset, a ResNet-50 teacher with low-resolution input (32×32) guides the student ViT architectures. The final distillation accuracy is obtained by combining the standard supervised loss computed on labeled data with a distillation loss. For their vanilla setup, the ViT architectures are trained without a teacher network or external knowledge.

We compute the correlation rankings between the top-2 ViT feature proxies in both token and embedding spaces using entropy and minimum eigenvalue ($u = 8$), and compare them with Flowers distillation accuracy across 100 ViT architectures - PiT. The results are presented in Table 22. As can be seen, our proxies strongly correlate with the distillation accuracy of ViT architectures on Flowers.

We select the top-2 strongest proxies from Table 22 and compare them with previous ViT proxies in terms of correlation with both vanilla and distillation accuracies on CIFAR-100, Flowers, and Chaoyang. Tables 23, 24, and 25 present the results for CIFAR-100, Flowers, and Chaoyang, respectively. As can be seen, on CIFAR-100, our proxies improve the correlation ranking with distillation accuracy by 1.7%. For the Flowers dataset, our proxies achieve a 3.7% improvement in correlation on vanilla accuracy and a 2.8% improvement on distillation accuracy. Furthermore, on the Chaoyang dataset, our proxies improve the correlation ranking by 1.6% for vanilla accuracy and 4.7% for

Table 11. Spearman's ρ and Kendall's τ correlations between test accuracy and each of the **entropy** proxies are computed for 1,000 ViT architectures (5–7M parameters) across 26 feature types. These proxies are calculated in both token and embedding spaces for individual and combined features, with dimension reductions $u = 1, 4, 8$ and 16. **Bold** values indicate correlations higher than those of previous proxies in Table 8.

Proxy		Token								Embedding							
		u=1		u=4		u=8		u=16		u=1		u=4		u=8		u=16	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Encoder	Q	**0.745**	**0.509**	**0.748**	**0.510**	**0.756**	**0.523**	**0.775**	**0.556**	**0.732**	**0.480**	**0.730**	**0.481**	**0.731**	**0.482**	**0.727**	**0.481**
	K	**0.733**	**0.485**	**0.739**	**0.493**	**0.731**	**0.484**	**0.724**	**0.477**	**0.738**	**0.495**	**0.735**	**0.492**	**0.732**	**0.487**	**0.722**	**0.473**
	V	**0.744**	**0.503**	**0.739**	**0.493**	**0.740**	**0.493**	**0.742**	**0.499**	**0.718**	**0.468**	**0.729**	**0.477**	**0.722**	**0.470**	**0.734**	**0.486**
	V'	**0.787**	**0.578**	**0.780**	**0.569**	**0.776**	**0.565**	**0.773**	**0.560**	0.666	0.453	0.672	0.462	0.672	0.462	0.673	0.462
	$MHSA_{out}$	**0.790**	**0.585**	**0.786**	**0.580**	**0.785**	**0.579**	**0.784**	**0.578**	0.657	0.439	0.672	0.459	**0.682**	**0.472**	**0.687**	**0.479**
	M_1	**0.844**	**0.660**	**0.846**	**0.663**	**0.847**	**0.663**	**0.846**	**0.663**	**0.762**	**0.571**	**0.760**	**0.569**	**0.760**	**0.570**	**0.762**	**0.572**
	M_2	**0.809**	**0.599**	**0.804**	**0.598**	**0.805**	**0.603**	**0.803**	**0.602**	**0.676**	**0.463**	**0.680**	**0.475**	**0.689**	**0.482**	**0.685**	**0.477**
	$GELU_{out}$	0.137	0.104	0.134	0.103	0.134	0.103	0.133	0.102	0.131	0.100	0.129	0.099	0.129	0.099	0.129	0.098
	SC_MSA_{out}	-0.039	-0.023	-0.047	-0.029	-0.059	-0.037	-0.073	-0.047	-0.056	-0.038	-0.045	-0.031	-0.047	-0.031	-0.067	-0.043
	SC_MLP_{out}	-0.023	-0.014	-0.053	-0.034	-0.063	-0.040	-0.074	-0.048	-0.043	-0.029	-0.044	-0.029	-0.052	-0.034	-0.058	-0.037
	LN_EP_{in}	-0.024	-0.014	-0.059	-0.038	-0.070	-0.046	-0.078	-0.051	-0.050	-0.033	-0.045	-0.031	-0.057	-0.038	-0.066	-0.042
	LN_EP_{out}	-0.031	-0.019	-0.062	-0.040	-0.068	-0.045	-0.079	-0.052	-0.071	-0.047	-0.061	-0.039	-0.054	-0.035	-0.045	-0.030
	$LN_1\_SC_{out}$	-0.039	-0.023	-0.047	-0.029	-0.059	-0.037	-0.073	-0.047	-0.056	-0.038	-0.045	-0.031	-0.049	-0.033	-0.067	-0.043
	$LN_2\_SC_{out}$	-0.029	-0.017	-0.049	-0.031	-0.059	-0.038	-0.074	-0.049	-0.055	-0.038	-0.058	-0.037	-0.053	-0.034	-0.045	-0.030
	$LN_3\_SC_{out}$	-0.023	-0.014	-0.053	-0.034	-0.063	-0.040	-0.074	-0.048	-0.043	-0.028	-0.044	-0.029	-0.054	-0.035	-0.058	-0.037
Patches	EP	-0.192	-0.158	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157
MLP Head	E	-0.055	-0.034	0.016	0.011	0.053	0.033	0.098	0.060	-0.088	-0.059	-0.113	-0.072	-0.168	-0.106	-0.193	-0.117
	LN_E_{out}	-0.011	-0.006	0.059	0.038	0.087	0.055	0.125	0.078	0.216	0.133	0.271	0.170	0.262	0.168	0.234	0.150
QKV		**0.773**	**0.544**	**0.775**	**0.546**	**0.777**	**0.550**	**0.778**	**0.551**	**0.720**	**0.496**	**0.714**	**0.491**	**0.713**	**0.489**	**0.712**	**0.489**
$M_1 + M_2$		**0.849**	**0.665**	**0.848**	**0.665**	**0.848**	**0.665**	**0.848**	**0.665**	**0.752**	**0.562**	**0.751**	**0.562**	**0.752**	**0.564**	**0.753**	**0.566**
$QKV + M_1 + M_2$		**0.838**	**0.652**	**0.838**	**0.651**	**0.823**	**0.630**	**0.837**	**0.652**	**0.837**	**0.649**	**0.836**	**0.648**	**0.836**	**0.648**	**0.836**	**0.648**
$QKV + MHSA_{out}$		**0.805**	**0.604**	**0.805**	**0.604**	**0.793**	**0.580**	**0.805**	**0.604**	**0.773**	**0.553**	**0.770**	**0.549**	**0.772**	**0.550**	**0.772**	**0.551**
$MHSA_{out} + M_1 + M_2$		**0.851**	**0.669**	**0.850**	**0.667**	**0.849**	**0.667**	**0.849**	**0.666**	**0.744**	**0.556**	**0.744**	**0.556**	**0.745**	**0.558**	**0.746**	**0.560**
$QKV + MHSA_{out} + M_1 + M_2$		**0.856**	**0.676**	**0.849**	**0.668**	**0.824**	**0.632**	**0.824**	**0.632**	**0.759**	**0.574**	**0.818**	**0.627**	**0.833**	**0.644**	**0.820**	**0.623**
$Q + K + V$		**0.740**	**0.499**	**0.749**	**0.505**	**0.750**	**0.507**	**0.745**	**0.502**	**0.728**	**0.477**	**0.731**	**0.482**	**0.728**	**0.481**	**0.727**	**0.480**
$SC\_MSA_{out} + SC\_MLP_{out}$		-0.040	-0.024	-0.053	-0.034	-0.060	-0.039	-0.065	-0.042	-0.056	-0.037	-0.046	-0.031	-0.052	-0.034	-0.058	-0.037

Table 12. Spearman's ρ and Kendall's τ correlations between test accuracy and each of the **minimum eigenvalue** proxies are computed for 1,000 ViT architectures (5–7M parameters) across 26 feature types. These proxies are calculated in both token and embedding spaces for individual and combined features, with dimension reductions $u = 1, 4, 8$ and 16. **Bold** values indicate correlations higher than those of previous proxies in Table 8.

Proxy		Token								Embedding							
		u=1		u=4		u=8		u=16		u=1		u=4		u=8		u=16	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Encoder	Q	**0.796**	**0.636**	0.585	0.408	0.671	0.462	**0.725**	**0.501**	**0.794**	**0.633**	0.643	0.433	0.626	0.421	0.620	0.422
	K	**0.792**	**0.630**	0.547	0.377	0.502	0.320	**0.745**	**0.528**	**0.797**	**0.637**	0.608	0.408	**0.681**	**0.453**	**0.697**	**0.489**
	V	**0.797**	**0.637**	**0.703**	**0.478**	**0.648**	**0.441**	**0.737**	**0.495**	**0.796**	**0.637**	0.477	0.317	0.531	0.356	**0.691**	**0.456**
	V'	**0.792**	**0.628**	0.656	0.468	0.596	0.389	**0.775**	**0.559**	**0.699**	**0.540**	0.407	0.263	0.461	0.304	0.481	0.305
	$MHSA_{out}$	**0.794**	**0.633**	0.277	0.184	0.331	0.214	0.519	0.345	**0.701**	**0.541**	0.498	0.340	0.468	0.321	0.488	0.333
	M_1	**0.795**	**0.633**	0.641	0.449	**0.677**	0.470	**0.727**	**0.493**	**0.761**	**0.585**	0.521	0.350	0.532	0.356	0.525	0.349
	M_2	**0.761**	**0.589**	0.617	0.430	**0.694**	**0.484**	**0.761**	**0.531**	**0.698**	**0.537**	0.487	0.325	0.493	0.328	0.558	0.375
	$GELU_{out}$	0.016	0.013	-0.094	-0.063	-0.068	-0.045	0.029	0.019	0.129	0.104	0.009	0.006	0.027	0.020	0.004	0.007
	SC_MSA_{out}	0.008	0.007	-0.068	-0.044	0.054	0.036	0.046	0.031	-0.138	-0.109	-0.086	-0.057	-0.108	-0.071	-0.098	-0.068
	SC_MLP_{out}	0.042	0.034	-0.050	-0.032	0.069	0.044	0.053	0.036	-0.138	-0.109	-0.073	-0.048	-0.109	-0.073	-0.111	-0.075
	LN_EP_{in}	0.021	0.017	-0.050	-0.033	0.049	0.031	0.033	0.023	-0.138	-0.109	-0.082	-0.054	-0.092	-0.062	-0.097	-0.066
	LN_EP_{out}	0.052	0.042	-0.050	-0.033	0.049	0.031	0.033	0.023	-0.142	-0.113	-0.104	-0.068	-0.060	-0.039	-0.192	-0.127
	$LN_1\_SC_{out}$	0.008	0.007	-0.068	-0.044	0.054	0.036	0.046	0.031	-0.138	-0.109	-0.086	-0.057	-0.108	-0.071	-0.156	-0.107
	$LN_2\_SC_{out}$	0.037	0.029	-0.068	-0.044	0.069	0.044	0.046	0.031	-0.139	-0.110	-0.078	-0.051	-0.083	-0.055	-0.098	-0.068
	$LN_3\_SC_{out}$	0.042	0.034	-0.050	-0.032	0.054	0.036	0.054	0.036	-0.138	-0.109	-0.073	-0.048	-0.109	-0.073	-0.157	-0.107
Patches	EP	NA	NA	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	-0.192	-0.157	0.192	0.157	-0.192	-0.157
MLP Head	E	0.076	0.061	0.028	0.010	-0.035	-0.020	0.371	0.246	-0.126	-0.096	-0.016	-0.009	0.019	0.013	0.052	0.036
	LN_E_{out}	0.091	0.074	0.028	0.010	-0.034	-0.020	0.371	0.246	-0.086	-0.067	-0.068	-0.050	0.031	0.018	-0.111	-0.075
QKV		**0.795**	**0.634**	0.612	0.432	0.628	0.413	**0.775**	**0.560**	**0.734**	**0.529**	0.625	0.425	0.558	0.371	0.653	0.449
$M_1 + M_2$		**0.797**	**0.639**	**0.685**	**0.484**	**0.745**	**0.525**	**0.749**	**0.514**	**0.752**	**0.576**	0.552	0.374	0.554	0.372	0.554	0.373
$QKV + M_1 + M_2$		**0.798**	**0.640**	**0.699**	**0.498**	**0.734**	**0.503**	**0.757**	**0.527**	**0.836**	**0.654**	**0.721**	**0.521**	**0.704**	**0.504**	**0.704**	**0.505**
$QKV + MHSA_{out}$		**0.798**	**0.639**	0.616	0.436	0.628	0.413	**0.775**	**0.560**	**0.771**	**0.567**	0.658	0.453	0.592	0.395	0.668	0.461
$MHSA_{out} + M_1 + M_2$		**0.798**	**0.640**	**0.687**	**0.485**	**0.745**	**0.525**	**0.749**	**0.514**	**0.745**	**0.570**	0.588	0.404	0.574	0.387	0.562	0.379
$QKV + MHSA_{out} + M_1 + M_2$		**0.798**	**0.640**	**0.700**	**0.500**	**0.734**	**0.503**	**0.757**	**0.527**	**0.831**	**0.650**	**0.734**	**0.532**	**0.714**	**0.513**	**0.709**	**0.510**
$Q + K + V$		**0.798**	**0.640**	**0.733**	**0.528**	**0.711**	**0.483**	**0.751**	**0.516**	**0.798**	**0.639**	**0.703**	**0.490**	**0.734**	**0.512**	**0.756**	**0.532**
$SC\_MSA_{out} + SC\_MLP_{out}$		0.027	0.022	-0.060	-0.039	0.065	0.042	0.050	0.033	-0.143	-0.114	-0.081	-0.054	-0.113	-0.075	-0.105	-0.072

distillation accuracy. These results demonstrate that our proxies are effective predictors for ViT distillation training across CIFAR-100, Flowers, and Chaoyang datasets.

Table 13. Spearman's ρ and Kendall's τ correlations between test accuracy and each of the **entropy** proxies in token space with dimension reduction $u = 1$, computed for 1,000 ViT architectures (15–19M parameters) and 1,000 ViT architectures (45–47M parameters) across 26 feature types, compared with baselines.

Proxy		15-19M		45-47M	
		ρ	τ	ρ	τ
SNIP [30]		0.280	0.190	0.056	0.037
GraSP [50]		-0.064	-0.043	0.023	0.015
TE-score [7]		-0.084	-0.057	-0.057	-0.106
NASWOT [36]		0.232	0.162	0.243	0.171
DSS [68]		0.468	0.315	-0.119	-0.079
AutoProxA [57]		0.446	0.299	-0.126	-0.084
DSS++ [69]		0.450	0.302	-0.140	-0.094
Encoder	Q	**0.544**	**0.369**	**0.003**	**0.002**
	K	**0.556**	**0.378**	**0.085**	**0.057**
	V	**0.512**	**0.344**	**0.03**	**0.02**
	V'	**0.582**	**0.401**	-0.024	-0.016
	$MHSA_{out}$	**0.529**	**0.356**	-0.038	-0.025
	M_1	**0.720**	**0.533**	**0.380**	**0.261**
	M_2	**0.530**	**0.355**	**0.003**	**0.002**
	$GELU_{out}$	0.413	0.290	**0.293**	**0.202**
	SC_MSA_{out}	0.145	0.097	**0.011**	**0.008**
	SC_MLP_{out}	0.166	0.111	-0.026	-0.018
	LN_EP_{in}	0.178	0.119	-0.015	-0.010
	LN_EP_{out}	0.185	0.124	-0.041	-0.027
	$LN_1\_SC_{out}$	0.145	0.097	**0.011**	**0.008**
	$LN_2\_SC_{out}$	0.153	0.103	**0.004**	**0.004**
	$LN_3\_SC_{out}$	0.166	0.111	-0.026	-0.018
Patches	EP	0.051	0.042	NA	NA
MLP Head	E	0.031	0.021	-0.055	-0.037
	LN_E_{out}	0.024	0.016	-0.050	-0.034
QKV		**0.559**	**0.380**	-0.178	-0.119
$M_1 + M_2$		**0.718**	**0.529**	**0.368**	**0.251**
$QKV + M_1 + M_2$		**0.675**	**0.484**	**0.232**	**0.158**
$QKV + MHSA_{out}$		**0.560**	**0.380**	-0.170	-0.114
$MHSA_{out} + M_1 + M_2$		**0.717**	**0.529**	**0.360**	**0.246**
$QKV + MHSA_{out} + M_1 + M_2$		**0.720**	**0.533**	**0.350**	**0.239**
$Q + K + V$		**0.541**	**0.367**	**0.084**	**0.056**
$SC\_MSA_{out} + SC\_MLP_{out}$		0.148	0.098	**0.004**	**0.003**

Table 14. Spearman's ρ and Kendall's τ correlation rankings of the top-3 Token-Entropy proxies ($u = 1$) across 1,000 ViT architectures (5–7M) using: 1) feature map 0, 2) feature map 10, and 3) a random feature map for each chosen module. **Comparable results across these sampling strategies demonstrate that our proxies are independent of the specific feature map selection.**

Proxy	Index 0		Index 10		Random dynamic index	
	ρ	τ	ρ	τ	ρ	τ
$M_1 + M_2$	0.871	0.687	0.875	0.693	0.870	0.685
$MHSA_{out} + M_1 + M_2$	0.874	0.692	0.874	0.692	0.868	0.682
$QKV + MHSA_{out} + M_1 + M_2$	0.864	0.676	0.864	0.677	0.863	0.676

Table 15. Spearman's ρ and Kendall's τ correlation rankings of the top-3 Token-Entropy proxies ($u = 1$) across 1,000 ViT architectures (5–7M) using original and Gaussian inputs. **Comparable results across inputs suggest that our proxies are data-independent.**

Proxy	Real Input		Gaussian Input	
	ρ	τ	ρ	τ
$M_1 + M_2$	0.849	0.665	0.847	0.663
$MHSA_{out} + M_1 + M_2$	0.851	0.669	0.846	0.662
$QKV + MHSA_{out} + M_1 + M_2$	0.856	0.676	0.837	0.650

K. Experiment Setup and Additional Results for Efficient ViT Design

To enable efficient ViT design, we employ our proxy in a one-shot Neural Architecture Search (NAS) framework to

Table 16. Spearman's ρ and Kendall's τ correlation rankings of the top-3 Token-Entropy proxies ($u = 1$) across 1,000 ViT architectures (5–7M) using original and noise-added input. **Comparable results across inputs suggest that our proxies are robust.**

Proxy	Original Data		Gaussian Noise	
	ρ	τ	ρ	τ
$M_1 + M_2$	0.849	0.665	0.849	0.665
$MHSA_{out} + M_1 + M_2$	0.851	0.669	0.844	0.659
$QKV + MHSA_{out} + M_1 + M_2$	0.856	0.676	0.836	0.649

Table 17. Spearman's ρ and Kendall's τ correlation rankings between the Token-Entropy proxy ($u = 1$) values and OoD accuracy across 1,000 ViT architectures (5–7M parameters). **Our proxy demonstrates promising performance, achieving results comparable to those of previous ViT proxies.**

Proxy	ρ	τ
DSS [68]	0.647	0.454
AutoProxA [57]	0.633	0.440
DSS++ [69]	0.636	0.443
$MHSA_{out} + M_1 + M_2$	**0.664**	**0.482**

Table 18. Spearman's ρ correlation rankings between the top-2 features proxies with entropy, minimum eigenvalue in token and embedding space ($u = 1$), and CIFAR-100 distillation accuracy, across 100 ViT architectures - AutoFormer. Bold indicates the top-2 strongest proxies. **Our proxies demonstrate that it is strong indicators for ViT architecture distillation training on CIFAR-100.**

Proxy	Entropy		Minimum Eigenvalue	
	Token	Embedding	Token	Embedding
$MHSA_{out} + M_1 + M_2$	**0.894**	0.811	0.793	0.811
$QKV + MHSA_{out} + M_1 + M_2$	**0.890**	0.867	0.793	0.868

Table 19. Spearman's ρ and Kendall's τ correlation rankings between the Token-Entropy proxy ($u = 1$) values and CIFAR-100 distillation accuracy across 100 ViT architectures - AutoFormer. **Our proxies achieved higher correlation than baselines, demonstrating that it is strong indicators for ViT architecture in CIFAR-100 distillation training.**

Proxy	ρ	τ
DSS [68]	0.715	0.530
AutoProxA [57]	0.750	0.558
DSS++ [69]	0.751	0.558
$MHSA_{out} + M_1 + M_2$	**0.894**	**0.724**
$QKV + MHSA_{out} + M_1 + M_2$	**0.890**	**0.715**

Table 20. Spearman's ρ correlation rankings between the top-2 features, using entropy and minimum eigenvalue proxies ($u = 1$) in token and embedding space, with Flowers distillation accuracy, across 100 ViT architectures - AutoFormer. Bold indicates the top-2 strongest proxies. **Our proxies demonstrate that they are strong indicators for ViT architecture distillation training on Flowers.**

Proxy	Entropy		Minimum Eigenvalue	
	Token	Embedding	Token	Embedding
$MHSA_{out} + M_1 + M_2$	0.682	**0.877**	0.465	**0.874**
$QKV + MHSA_{out} + M_1 + M_2$	0.672	0.827	0.465	0.828

Table 21. Spearman's ρ and Kendall's τ correlation rankings between the entropy, minimum eigenvalue proxies ($u = 1$) in embedding space and Flowers distillation accuracy across 100 ViT architectures - AutoFormer. **Our proxies achieve higher correlations than the baselines, demonstrating that they are strong indicators for ViT architecture performance in Flowers distillation training.**

Proxy	ρ	τ
DSS [68]	0.819	0.655
AutoProxA [57]	0.823	0.657
DSS++ [69]	0.834	0.679
$MHSA_{out}$ + M_1 + M_2 (Entropy)	**0.877**	**0.723**
$MHSA_{out}$ + M_1 + M_2 (MinEigen)	**0.874**	**0.728**

Table 22. Spearman's ρ correlation rankings between the top-2 features in token and embedding space, using entropy and minimum eigenvalue proxies ($u = 8$), and Flowers distillation accuracy across 100 ViT architectures - PiT. Bold indicates the top-2 strongest proxies. **Our proxies demonstrate that they are strong indicators for ViT architecture distillation training on Flowers in the PiT search space.**

Proxy	Entropy		Minimum Eigenvalue	
	Token	Embedding	Token	Embedding
$MHSA_{out} + M_1 + M_2$	0.618	0.900	0.606	0.911
$QKV + MHSA_{out} + M_1 + M_2$	0.605	**0.953**	0.585	**0.964**

Table 23. Spearman's ρ and Kendall's τ correlation rankings between the entropy, minimum eigenvalue proxies ($u = 8$) in embedding space and CIFAR-100 distillation accuracy across 100 ViT architectures - PiT. **Our proxies achieve higher correlations than the baselines, demonstrating that they are strong indicators for ViT architecture performance in CIFAR-100 distillation training.**

Proxy	Vanilla		Distillation	
	ρ	τ	ρ	τ
DSS [68]	0.831	0.656	0.823	0.640
AutoProxA [57]	0.827	0.640	0.907	0.743
DSS++ [69]	0.879	0.708	0.865	0.684
$QKV + MHSA_{out}$ + M_1 + M_2 (Entropy)	**0.861**	**0.670**	**0.910**	**0.744**
$QKV + MHSA_{out}$ + M_1 + M_2 (MinEigen)	**0.863**	**0.677**	**0.924**	**0.763**

Table 24. Spearman's ρ and Kendall's τ correlation rankings between the entropy, minimum eigenvalue proxies ($u = 8$) in embedding space, and Flowers distillation accuracy across 100 ViT architectures - PiT. **Our proxies achieve higher correlations than the baselines, demonstrating that they are strong indicators for ViT architecture performance in Flowers distillation training.**

Proxy	Vanilla		Distillation	
	ρ	τ	ρ	τ
DSS [68]	0.858	0.691	0.850	0.680
AutoProxA [57]	0.886	0.706	0.936	0.792
DSS++ [69]	0.893	0.729	0.891	0.722
$QKV + MHSA_{out}$ + M_1 + M_2 (Entropy)	**0.919**	**0.750**	**0.953**	**0.817**
$QKV + MHSA_{out}$ + M_1 + M_2 (MinEigen)	**0.930**	**0.774**	**0.964**	**0.840**

identify the optimal architecture that achieves the highest accuracy under computational constraints. The process consists of two stages. In the first stage, we integrate our proxy into the searching method to predict the optimal ViT architecture within the search space. In the second stage, the selected ViT architecture is retrained from scratch to achieve its final performance.

Table 25. Spearman's ρ and Kendall's τ correlation rankings between the entropy, minimum eigenvalue proxies ($u = 8$) in embedding space and Chaoyang distillation accuracy across 100 ViT architectures - PiT. **Our proxies achieve higher correlations than the baselines, demonstrating that they are strong indicators for ViT architecture performance in Chaoyang distillation training.**

Proxy	Vanilla		Distillation	
	ρ	τ	ρ	τ
DSS [68]	0.653	0.472	0.670	0.492
AutoProxA [57]	0.713	0.523	0.722	0.546
DSS++ [69]	0.765	0.572	0.740	0.560
$QKV + MHSA_{out}$ + M_1 + M_2 (Entropy)	**0.781**	**0.589**	**0.787**	**0.610**
$QKV + MHSA_{out}$ + M_1 + M_2 (MinEigen)	**0.772**	**0.580**	0.782	**0.601**

Because our proxy can compute values for thousands of architectures in a short time, it significantly reduces the search time compared to traditional NAS methods and previous proxies. Moreover, in practice, when using our proxy as an indicator within a one-shot NAS framework, it is unnecessary to train the large Supernet to estimate the performance of all architectures. For other types of NAS, it eliminates the need to train thousands of sub-architectures to evaluate their performance. Therefore, our method substantially reduces the time required for training, making the design of efficient ViTs more practical.

K.1. AutoFormer Search Space

Searching setup: We use the Token-Entropy proxy $MHSA_{out} + M_1 + M_2$ ($u = 8$) to search for the optimal subnet within the AutoFormer Supernet NAS architecture for ImageNet1K [11]. In AutoFormer-Tiny, we constrain the parameter range to 4-6 million and perform evolutionary search over 2,000 ViT architectures in 8.67 hours. In AutoFormer-Small, we search for the optimal subnet within the 20-23 million parameter range using 1,400 ViT architectures in 4.12 hours. For AutoFormer-Base, the parameter range is constrained to 51-54 million, and the search is conducted over 600 ViT architectures in 1.48 hours. The configuration details for evolutionary subnet search are provided in Table 26. We use one GPU, with 10 workers and a batch size of 64. The input resolution is set to 224×224, and the images are normalized using the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225].

Table 26. Searching setting when using Token-Entropy $MHSA_{out} + M_1 + M_2$ ($u = 8$) Proxy in Autoformer search space.

Size	Param (M)	Search time (h)	Population
Tiny	4-6	8.67	2000
Small	20-23	4.12	1400
Base	51-54	1.48	600

Training setup. We retrain the three searched ViT architectures on ImageNet-1K [11], employing the same image resolution and preprocessing as in the search phase to ensure consistency. All models are trained for 500 epochs, including 20 warm-up epochs. The warm-up learning rate is set to 1×10^{-6}, the minimum learning rate to 1×10^{-5}, and the base learning rate is 0.0005, scheduled using a cosine scheduler. The optimization is performed with the AdamW optimizer, with a weight decay of 0.05 and a momentum of 0.9. Training is conducted on 2 GPUs with 12 workers.

- Tiny ViT Architecture: Batch size of 1024.
- Small ViT Architecture: Batch size of 512.
- Base ViT Architecture: Batch size of 256.

Metrics. The evaluation results are reported using Top-1 and Top-5 accuracy.

K.2. Transfer Learning

Experiment setup: We fine-tune the searched architecture $MHSA_{out} + M_1 + M_2$ - S at an input resolution of 384×384 on CIFAR-10 and CIFAR-100 [28]. The warm-up learning rate is 1×10^{-6}, the minimum learning rate is 1×10^{-5}, and the base learning rate is 0.0005, scheduled with a cosine scheduler. We use the AdamW optimizer with a weight decay of 0.05 and a momentum of 0.9. The number of workers is set to 12.

- **CIFAR-10.** Batch size of 64, trained on 2 GPUs for 385 epochs, with 50 warm-up epochs.
- **CIFAR-100.** Batch size of 32, trained on 4 GPUs for 500 epochs, with 50 warm-up epochs.

Table 27. Transfer learning results on CIFAR-10 and CIFAR-100, reported in terms of Top-1 accuracy and the number of parameters.

Model	Param	CIFAR-10	CIFAR-100
ViT-B/16 [14]	86M	98.1	87.1
DeiT-B [48]	86M	99.1	90.8
Autoformer-S [5]	23M	99.1	91.1
TF-TAS-S [68]	23M	99.1	91.2
T-Razor-S [69]	23M	99.1	91.3
$MHSA_{out}$ + M_1 + M_2 - S	**23M**	**99.1**	**91.3**

Results: As shown in Table 27, the optimal ViT architecture identified by our proposed proxy achieves transfer learning performance comparable to AutoFormer-S [5], TF-TAS-S [68], and T-Razor-S [69].

K.3. PiT Search Space

Searching setup. We use our Token-Entropy proxy $MHSA_{out} + M_1 + M_2$ $(u = 8)$ to search for the optimal ViT architecture within the PiT search space for ImageNet-1K [11]. We constrain the parameter range to 1–5 million and perform the search over 500 ViT architectures. The total search time is 0.166 hours. The search is conducted using one GPU, with 10 workers and a batch size of 64. We use an input resolution of 224×224, and images are normalized using the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225].

Training setup. We train the searched ViT architectures on ImageNet-1K [11] and use the same image resolution and preprocessing as in the search setup. We use a batch size of 256 and train for 500 epochs. The learning rate is set to 0.0005 with a cosine scheduler. We adopt the AdamW optimizer with a weight decay of 0.05 and a momentum parameter of 0.9. Training is conducted using 2 GPUs with 10 workers.

Metrics. We present the results in terms of Top-1 and Top-5 accuracy.

L. Limitation

Our evaluation is comprehensive and consistent with prior work. However, examining a larger search space in future studies could provide additional confirmation of the accuracy improvements achieved by our proxies.