InTrain: Intrinsic Trainability for Zero-Cost Neural Architecture Search

Supplementary Material

A. Geometric Capacity

The purpose of Eq. 1 in the main text is to quantify the effective dimensionality of the activation manifold at layer ℓ prior to training. We demonstrate that the Participation Ratio (PR) is mathematically equivalent to the exponential of the order-2 Rényi entropy of the normalized eigenspectrum.

Let $\mathbf{C}_\ell \in \mathbb{R}^{D\times D}$ be the covariance matrix of the post-activation features at layer ℓ. Let its eigenvalues be denoted as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D \geq 0$. We define the normalized spectral density as:

$$p_i = \frac{\lambda_i}{\sum_{j=1}^{D} \lambda_j} = \frac{\lambda_i}{\mathrm{Tr}(\mathbf{C}_\ell)}, \quad \text{where} \sum_{i=1}^{D} p_i = 1. \tag{1}$$

The continuous Participation Ratio is defined algebraically using the trace operator as:

$$\mathrm{PR}(\mathbf{C}_\ell) = \frac{(\mathrm{Tr}\,\mathbf{C}_\ell)^2}{\mathrm{Tr}(\mathbf{C}_\ell^2)} = \frac{(\sum_{i=1}^{D} \lambda_i)^2}{\sum_{i=1}^{D} \lambda_i^2} \tag{2}$$

By substituting the definition of p_i from Eq. 1 into the sum of squared probabilities, we obtain:

$$\sum_{i=1}^{D} p_i^2 = \sum_{i=1}^{D} \left(\frac{\lambda_i}{\mathrm{Tr}(\mathbf{C}_\ell)}\right)^2 = \frac{\sum_{i=1}^{D} \lambda_i^2}{(\mathrm{Tr}\,\mathbf{C}_\ell)^2} = \frac{\mathrm{Tr}(\mathbf{C}_\ell^2)}{(\mathrm{Tr}\,\mathbf{C}_\ell)^2} \tag{3}$$

Comparing Eq. 2 and Eq. 3, it immediately follows that:

$$\mathrm{PR}(\mathbf{C}_\ell) = \frac{1}{\sum_{i=1}^{D} p_i^2} \tag{4}$$

Recognizing that the order-2 Rényi entropy for a discrete probability distribution $\mathbf{p}$ is defined as $H_2(\mathbf{p}) = -\log(\sum_{i=1}^{D} p_i^2)$, we can rewrite the collision probability as $\sum p_i^2 = \exp(-H_2(\mathbf{p}))$. Substituting this yields the exact identity:

$$\mathrm{PR}(\mathbf{C}_\ell) = \exp(H_2(\mathbf{p})) \tag{5}$$

Eq. 5 proves that our capacity metric is a fundamental information-theoretic measure. Consequently, an architecture scoring a higher PR inherently exhibits a larger entropy in its covariance spectrum. This allows us to explicitly identify and select networks that naturally avoid dimensional collapse and possess a rich, isotropic state-space volume without requiring actual training.

B. Optimization Resilience

Eq. 4 in the main text assesses gradient health by computing the ratio of the gradient's standard deviation to its maximum absolute value ($\sigma(\nabla)/\max|\nabla|$). We show that under local linearization, this statistical ratio acts as a direct estimator for the square root of the Jacobian's stable rank, a key property reflecting dynamical stability.

Let θ represent the parameters of a specific layer, and $\mathcal{L}$ be the loss function. During backpropagation, the gradient $\nabla_\theta \mathcal{L}$ is heavily governed by the end-to-end Jacobian $\mathbf{J}$ mapping the layer's output to the network's final output. Under the local linearization assumption, we can express the gradient flow as:

$$\nabla_\theta \mathcal{L} \approx \mathbf{J}^T \mathbf{u} \tag{6}$$

where $\mathbf{u}$ is the upstream error signal. For the purpose of spectral analysis, we assume $\mathbf{u}$ is isotropic, i.e., $\mathbb{E}[\mathbf{u}\mathbf{u}^T] \propto \mathbf{I}$. Let the singular value decomposition (SVD) of $\mathbf{J}^T$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma} = \mathrm{diag}(s_1, s_2, \ldots, s_M)$ contains the singular values $\mathbf{s}$.

We establish the mapping between the gradient's statistical moments and the singular values of $\mathbf{J}$:

- The expected energy (variance) of the gradient is given by the trace of its covariance matrix:

$$\begin{aligned} \mathrm{Var}(\nabla_\theta \mathcal{L}) &\propto \mathbb{E}[\|\mathbf{J}^T\mathbf{u}\|_2^2] = \mathrm{Tr}(\mathbf{J}^T \mathbb{E}[\mathbf{u}\mathbf{u}^T]\mathbf{J}) \\ &\propto \mathrm{Tr}(\mathbf{J}^T\mathbf{J}) = \|\mathbf{s}\|_2^2 \end{aligned} \tag{7}$$

  Thus, the standard deviation scales with the ℓ_2-norm of the singular values: $\sigma(\nabla_\theta \mathcal{L}) \propto \|\mathbf{s}\|_2$.
- The maximum absolute response in the gradient vector is bounded by the dominant spectral direction of the operator $\mathbf{J}$. Mathematically, this corresponds to the operator norm induced by the largest singular value:

$$\max|\nabla_{\theta,i}| \sim O(\|\mathbf{J}^T\|_2) = s_{\max} = \|\mathbf{s}\|_\infty \tag{8}$$

Taking the ratio of these two quantities, we derive our proposed statistic:

$$\frac{\sigma(\nabla_\theta \mathcal{L})}{\max|\nabla_{\theta,i}|} \propto \frac{\|\mathbf{s}\|_2}{\|\mathbf{s}\|_\infty} \tag{9}$$

In numerical linear algebra and dynamical systems, the stable rank of a matrix $\mathbf{J}$ is defined as $r(\mathbf{J}) =$

$\|\mathbf{J}\|_F^2/\|\mathbf{J}\|_2^2 = \|\mathbf{s}\|_2^2/\|\mathbf{s}\|_\infty^2$. Comparing this definition with Eq. 9, it is evident that our metric estimates the square root of the stable rank:

$$\frac{\sigma(\nabla_\theta \mathcal{L})}{\max|\nabla_{\theta,i}|} \approx \sqrt{r(\mathbf{J})} \tag{10}$$

A high stable rank indicates that the singular values are uniformly distributed, whereas a low stable rank indicates the dominance of a single singular value (predicting exploding gradients along one direction and vanishing along others). Therefore, Eq. (4) correlates with Jacobian spectral spread, a quantity that influences dynamical amplification and hence is related to notions of Lyapunov-type instability in continuous limits.

C. InTrain-NAS Algorithm Details

Algorithm 1 outlines the evolutionary search framework of InTrain-NAS, which employs a history-based elitist selection strategy. The algorithm takes four parameters: the search space $\mathcal{Z}$, maximum iterations T, computational budget B, and elite set size k. The search begins by initializing a random architecture that satisfies the budget constraint. We maintain history sets for architectures and their InTrain scores. During each iteration, the current architecture is evaluated using the InTrain metric, and the results are stored in the history sets. A new candidate architecture is then generated by mutating one of the top-k performing architectures from the history. After T iterations, the algorithm returns the architecture with the highest InTrain score from the history set. This approach efficiently explores the search space by continually refining high-scoring architectures while maintaining diversity through mutation operations.

D. More Implementation Details

Hardware Environment. All experiments are conducted using NVIDIA RTX 4090 GPUs. The NAS search experiments are distributed across multiple GPUs for model training, while proxy evaluations (InTrain scoring) are performed on a single GPU. The reported search costs in GPU days correspond to single-GPU equivalence for fair comparison.

Hyperparameter Settings. For proxy evaluation, we use batch size $B = 64$, input resolution $R = 64$, and synthetic inputs $\mathbf{X} \sim \mathcal{N}(0, 1)$. During evolutionary search, we follow Algorithm 1 with maximum iterations $T = 100000$ and mutation elite set size $k = 10$ for parent selection. These configurations follow standard NAS practices [3, 5, 9].

Algorithm 1 The evolutionary search framework of InTrain-NAS

Input: Max Iterations T; search space $\mathcal{Z}$; inference budget B; mutation elite size k.
Output: Searched Optimal architecture $\mathcal{A}^*$.
Randomly select an architecture $\mathcal{A}_1$ that meets the inference budget B.
Initialize history sets $\mathbb{A} = \{\}$ and proxy scores set $\mathcal{S} = \{\}$.
for $i = 1$ to T **do**
    Compute the InTrain score s_i of the architecture $\mathcal{A}_i$.
    Append the architecture $\mathcal{A}_i$ to the history set $\mathbb{A}$.
    Append the score s_i to the proxy scores set $\mathcal{S}$.
    Select one of the top-k architectures from $\mathbb{A}$ based on InTrain scores.
    Mutate the selected architecture to generate $\mathcal{A}_{i+1} \in \mathcal{Z}$, ensuring it meets the budget B.
end for
$\mathcal{A}^*$ = the architecture of the highest InTrain score in $\mathbb{A}$.

E. Additional Experimental Results

Searched Results on NAS-Bench-201. To provide a more comprehensive evaluation of our proposed InTrain-NAS method, we conduct additional experiments on the NAS-Bench-201 benchmark using an extended sample size of 5000 architectures. As shown in Table 1, our method consistently achieves state-of-the-art performance across all three datasets, outperforming all existing zero-cost NAS approaches.

Notably, InTrain-NAS achieves the highest accuracy on CIFAR-10 (93.72%), CIFAR-100 (71.05%), and ImageNet16-120 (45.85%), with remarkably low standard deviations, indicating superior stability and robustness. Compared to the strongest baseline AZ-NAS, our method demonstrates improvements of +0.07%, +0.13%, and +0.17% on the three datasets respectively. Furthermore, the performance gap between our method and the ground truth upper bound is significantly reduced, particularly on the more challenging ImageNet16-120 dataset where we achieve 45.85% compared to the ground truth of 47.18%. The extended evaluation with a larger sample size of 5000 architectures validates the scalability and effectiveness of our approach, confirming that the performance advantages observed in the main experiments are consistent and statistically significant across a broader range of architectural samples.

Effect of Batch Size. We analyze the impact of batch size on InTrain's performance, evaluating the trade-off between correlation accuracy and computational latency. We test batch sizes from $B = 1$ to $B = 128$ on NAS-Bench-201

Table 1. Top-1 Accuracy Comparison on 5000 architectures randomly sampled from NAS-Bench-201. The table compares the average Top-1 accuracy of the architectures found by various zero-cost NAS methods on the NAS-Bench-201 search space. The results are averaged over 5 random search runs and reported with standard deviation.

Method	**CIFAR-10 Top-1 Acc. (%)**	**CIFAR-100 Top-1 Acc. (%)**	**ImageNet16-120 Top-1 Acc. (%)**
#Params	93.58 ± 0.15	70.75 ± 0.25	42.15 ± 0.65
FLOPs	93.58 ± 0.15	70.75 ± 0.25	42.15 ± 0.65
GradNorm [1]	90.25 ± 0.85	63.18 ± 1.20	25.12 ± 4.25
Snip [4]	89.65 ± 1.45	62.05 ± 2.65	27.65 ± 4.50
Grasp [10]	90.48 ± 1.25	63.68 ± 2.30	29.15 ± 5.30
Synflow [8]	93.22 ± 0.75	69.48 ± 1.85	40.85 ± 5.25
NASWOT [7]	92.82 ± 0.20	70.15 ± 0.23	44.68 ± 0.75
TE-NAS [2]	92.45 ± 0.28	68.12 ± 0.95	40.75 ± 1.95
ZenNAS [6]	90.45 ± 0.32	67.95 ± 0.90	39.80 ± 1.15
ZiCo [5]	93.58 ± 0.15	70.78 ± 0.22	42.28 ± 0.75
AZ-NAS [3]	93.65 ± 0.12	70.92 ± 0.42	45.68 ± 0.25
VKDNW$_{single}$ [9]	93.63 ± 0.14	70.88 ± 0.27	45.41 ± 0.31
InTrain (Ours)	$\mathbf{93.72 \pm 0.11}$	$\mathbf{71.05 \pm 0.38}$	$\mathbf{45.85 \pm 0.22}$
Ground Truth	94.35 ± 0.14	73.17 ± 0.23	47.18 ± 0.26

Table 2. Batch-size sensitivity study on NAS-Bench-201 (CIFAR-10). For each batch size we report Kendall's τ, Spearman's ρ (mean $\pm$ std over S seeds), and the single-run forward+backward latency (ms) averaged over architectures.

Batch size	Kendall's τ	Spearman's ρ	Latency (ms)
B = 1	0.643	0.839	25.73
B = 8	0.651	0.848	28.56
B = 16	0.659	0.850	32.95
B = 32	0.662	0.852	34.41
B = 64	0.669	0.857	37.29
B = 128	0.671	0.858	44.65

(CIFAR-10), with results presented in Table 2. As shown in the table, the quality of the proxy score, measured by both Kendall's τ and Spearman's ρ, consistently improves with larger batch sizes. The correlation (KT) increases from 0.643 at $B = 1$ to 0.669 at $B = 64$. However, we observe diminishing returns beyond this point—increasing to $B = 128$ yields minimal correlation gain (KT: $0.669 \rightarrow 0.671$) while increasing latency by $\approx 20\%$ (37.29ms $\rightarrow$ 44.65ms). This analysis confirms that InTrain is robust (*i.e.*, does not collapse) even at small batch sizes (*e.g.*, $B = 8, \tau = 0.651$), but achieves its optimal balance of high correlation and low latency at $B = 64$. Therefore, we use $B = 64$ as the default for all other experiments in this paper.

References

[1] Mohamed S Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas D Lane. Zero-cost proxies for lightweight nas. In *International Conference on Learning Representations*, 2021. 3

[2] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021. 3

[3] Junghyup Lee and Bumsub Ham. Az-nas: Assembling zero-cost proxies for network architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5893–5903, 2024. 2, 3

[4] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. 3

[5] Guihong Li, Yuedong Yang, Kartikeya Bhardwaj, and Radu Marculescu. Zico: Zero-shot nas via inverse coefficient of variation on gradients. 2023. 2, 3

[6] Ming Lin, Pichao Wang, Zhenhong Sun, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Zen-nas: A zero-shot nas for high-performance image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 347–356, 2021. 3

[7] 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. 3

[8] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iter-

atively conserving synaptic flow. *Advances in neural information processing systems*, 33:6377–6389, 2020. 3

[9] Ondrej Tybl and Lukas Neumann. Training-free neural architecture search through variance of knowledge of deep network weights. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14881–14890, 2025. 2, 3

[10] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. 3