ParZC: Parametric Zero-Cost Proxies for Efficient NAS

Peijie Dong[1*], Lujun Li[2*], Zhenheng Tang[2], Xiang Liu[1], Zimian Wei[3], Qiang Wang[4], Xiaowen Chu[1,2†]

[1]The Hong Kong University of Science and Technology (GuangZhou)
[2]The Hong Kong University of Science and Technology
[3]National University of Defense Technology
[4]Harbin Institute of Technology, Shenzhen
{pdong212, xliu886}@connect.hkust-gz.edu.cn, lilujunai@gmail.com, zhtang.ml@ust.hk, weizimian16@nudt.edu.cn, qiang.wang@hit.edu.cn, xwchu@hkust-gz.edu.cn

Abstract

Recent advancements in Zero-shot Neural Architecture Search (NAS) highlight the ability of zero-cost proxies in identifying superior architecture. However, we identify a critical issue with current zero-cost proxies: they aggregate node-wise zero-cost statistics without considering that not all nodes in a neural network equally impact performance estimation. Our observations reveal that node-wise zero-cost statistics significantly vary in their contributions to performance, with each node exhibiting a degree of uncertainty. Based on this insight, we introduce a novel method called Parametric Zero-Cost Proxies (ParZC) framework to enhance the adaptability of zero-cost proxies through parameterization. To address the node indiscrimination, we propose a Mixer Architecture with Bayesian Network (MABN) to explore the node-wise zero-cost statistics and estimate node-specific uncertainty. Moreover, we propose DiffKendall as a loss function to improve ranking consistency. Comprehensive experiments on NAS-Bench-101, 201, and NDS demonstrate the superiority of our proposed ParZC compared to existing zero-shot NAS methods. Additionally, we demonstrate the versatility and adaptability of ParZC on Vision Transformer search space.

Introduction

Neural Architecture Search (NAS) (He et al. 2023) has been proposed to search for optimal architectures automatically, which show better performance than the manually-designed networks. Despite its potential, NAS has been criticized for its substantial requirements on computational resources. For instance, NASNet (Zoph and Le 2017) necessitated 2k GPU hours to identify an architecture. To alleviate it, surrogate-based methods (Chen et al. 2019), one-shot NAS (Pham et al. 2018a; Liu, Simonyan, and Yang 2019; Hu et al. 2021), and zero-shot NAS (Mellor et al. 2021; Lin et al. 2021) are investigated to expedite the process. Among them, Zero-shot NAS is an innovative approach that leverages proxies or metrics to predict the performance of neural networks at initialization, significantly reducing the computational resources and time typically required for traditional NAS.

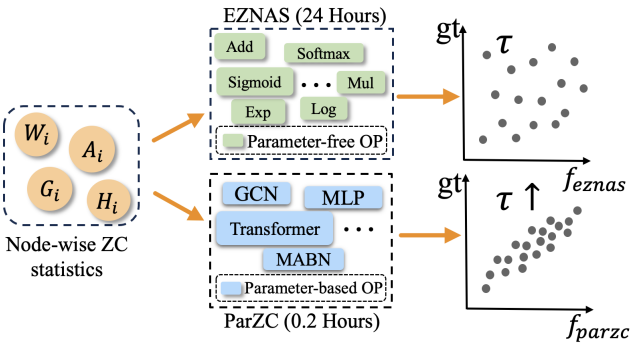


Figure 1: **Overview of ParZC and EZNAS (Akhauri et al. 2022) pipeline**. W: Weight, A: Activation, G: Gradient, H: Hessian Matrix.

Despite its high efficiency, ZC proxy exhibits several limitations: **(1) Homogeneity Assumption:** Previous methods (Mellor et al. 2021; Lin et al. 2021; Tanaka et al. 2020; Lee, Ajanthan, and Torr 2019; Abdelfattah et al. 2021) rely on the underlying assumption that each node has an equal influence on the ZC proxy calculation, which has been questionable (Cavagnero et al. 2023; Wang et al. 2023; Shu et al. 2022b). Here, a node represents a basic operation within the network, such as a 1x1, 3x3 convolution, or other basic computational units. **(2) High Data Demand:** Recent automated methods (Akhauri et al. 2022; Dong et al. 2023a; Li et al. 2023b, 2024) aiming to search for new proxies or integrate existing ones (Shu et al. 2022b) face challenges due to their significant data requirements. For example, EZNAS (Akhauri et al. 2022) first proposes a framework that evolves zero-cost proxies from scratch for neural architecture scoring, which requires 15,625 samples during searching. HNAS (Shu et al. 2022b) utilizes Bayesian optimization to combine the NTK (Jacot, Gabriel, and Hongler 2018) and existing proxies, which also demand 2,000 samples for optimization. Obtaining these samples (architecture-accuracy pairs) typically involves training each architecture from scratch, which is expensive and impractical.

For the homogeneity assumption, we conduct an intuitive experiment, as presented in Fig. 2. We collect the node-wise

*These authors contributed equally.
†Corresponding author

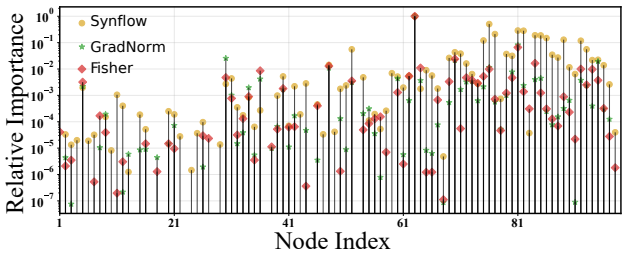


Figure 2: **Node-wise relative importance of ZC proxies** (Synflow (Tanaka et al. 2020), GradNorm (Abdelfattah et al. 2021), and Fisher (Turner et al. 2020)) based on GBDT impurity on NAS-Bench-201.

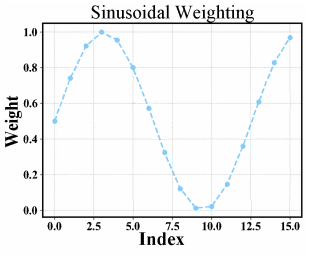


Figure 3: Layer-wise non-negative weighting schema for ParZC†.

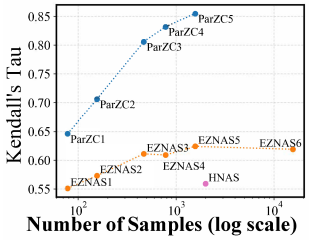


Figure 4: Comparison with HNAS and EZNAS with same number of samples.

ZC statistics of three ZC Proxies and employ GBDT (Friedman 2001) for regression analysis (see Supp. A.2 for GBDT details). Then, we visualize the relative importance of each node-wise ZC statistics using impurity in Fig. 2. In general, nodes in the deeper layer are more significant than shallower ones. This phenomenon reveals that node-wise ZC statistics significantly vary in their contributions to performance.

To address the challenge of non-homogeneous node handling, we propose the Parametric Zero-Cost Proxies (ParZC) framework as presented in Fig. 1. This framework introduces a novel approach to proxy design that leverages trainable parameters to adaptively learn and represent the unique characteristics of different nodes within a network. Specifically, we propose Mixer Architecture with Bayesian Network (MABN) to learn how to rank architectures within the search space. We incorporate a Bayesian Network into the Mixer Architecture to enhance the ability to measure uncertainty. Additionally, we directly optimize rank correlation by relaxing Kendall's Tau so that ParZC can effectively handle discrepancies in the ranking of architectures. ParZC significantly enhances both the rank correlation and the efficiency of the search process. Furthermore, we propose a training-free non-negative weighting scheme, denoted as ParZC† as an alternative of MABN. As illustrated in Fig. 3, this scheme employs sinusoidal weighting to assign differential importance to nodes based on their indices. It can potentially enhance ranking consistency in a complementary manner without requiring additional training. We present the overview of ParZC and EZNAS in Fig. 1 and the ranking correlation of HNAS (Shu et al. 2022b) and EZNAS (Akhauri et al. 2022) with ParZC w.r.t. the number of samples in Fig. 4. Our contributions are as follows:

- We introduce Parametric Zero-Cost Proxies (ParZC), a hybrid adaptable ZC proxy framework that leverages the uncertainty inherent in node-wise ZC proxies.
- We incorporate the Mixer Architecture with the Bayesian Network (MABN) to estimate uncertainty for node-wise ZC statistics. Additionally, we introduce DiffKendall, a novel approach designed to enhance ranking capabilities.
- We validate ParZC's superiority through comprehensive experiments conducted on NAS-Bench-101, 201, and NDS benchmarks.

Related Work

Training-free NAS (Lin et al. 2021; Mellor et al. 2021; Dong, Li, and Wei 2023; Tang et al. 2024a; Dong et al. 2024) employs zero-cost proxies as predictive indicators. This approach involves evaluating architectures using ZC proxies on randomly initialized weights, with just a limited number of forward and backward passes with a mini-batch of input data, thereby significantly enhancing efficiency. Zero-shot NAS (Ning et al. 2021; Wei et al. 2023; Li et al. 2025) can be categorized into two main types based on how to handle the neural network: (1) **Node-level zero-shot NAS** adopt from pruning literature including GradNorm (Abdelfattah et al. 2021), SNIP (Lee, Ajanthan, and Torr 2019), GraSP (Wang, Zhang, and Grosse 2020) Fisher (Turner et al. 2020), and Synflow (Tanaka et al. 2020). These ZC proxies are named after sensitivity indicators initially designed for fine-grained network pruning that measure the approximate loss change when certain parameters or activations are pruned. These ZC proxies (Abdelfattah et al. 2021) evaluate an architecture by summing up sensitivities of all nodes. (2) **Architecture-level zero-shot NAS** holistically assesses the architecture's discriminability by discerning variances among distinct input images. NWOT (Mellor et al. 2021) proposes a heuristic metric based on local Jacobian values to estimate the performance. ZenNAS (Lin et al. 2021) evaluates the candidate architectures using the gradient norm (Tang et al. 2024c) of the input image as a ranking score. HNAS (Shu et al. 2022b) reveals theoretical guarantees for zero-shot proxies and enhance the existing proxies (Abdelfattah et al. 2021) using Bayesian optimization. EZNAS (Akhauri et al. 2022) proposes to search for the zero-cost proxies by breaking down proxy into the combination of operations.

Predictor-based NAS streamlines the architecture search process by predicting the performance of neural architectures based on their encoded representations. These methods primarily focus on the encoding scheme and the design of the predictor model. Various predictors have been employed, including embedding matrices (Luo et al. 2020; Ning et al. 2020a; Tang et al. 2024b), GCN (Wen et al. 2020; Chen et al. 2021c), MLP (White, Neiswanger, and Savani 2021), and Transformers (Lu et al. 2021, 2023). One-shot NAS (Guo et al. 2019; Chu, Zhang, and Xu 2021; Tang et al. 2024b; Dong et al. 2023b; He et al. 2023) can also be considered predictor-based NAS, using the supernet as a perfor-

mance estimator. Several approaches have been proposed to enhance encoding techniques: NAO (Luo et al. 2018) incorporates LSTM-based autoencoders to generate latent representations, Arch2Vec (Yan et al. 2020) transforms architectures into a continuous vector space, GATES (Ning et al. 2020b; Shen et al. 2024) utilizes a graph-based scheme representing architectures as DAGs (Tang et al. 2024a). TA-GATES (Ning et al. 2022) further employ ZC to break the symmetry of architecture encoding. NP (Wen et al. 2020) employs adjacency matrices. CTNAS (Chen et al. 2021c) introduces a contrastive learning approach, encoding architectures based on their comparative performance. Recent works such as TNASP (Lu et al. 2021) and PINAT (Lu et al. 2023) leverage transformer architectures for encoding, further advancing the field of predictor-based NAS. In this paper, we investigate the potential in the node-wise ZC statistic, which is a fine-grained way to encode the architectures.

Parametric Zero-Cost Proxies

In this section, we present our ParZC framework (illustrated in Fig. 5), comprising a node-wise ZC encoding scheme and a Mixer Architecture with Bayesian Network. To improve the ranking correlation, we propose DiffKendall to optimize the parameters of ParZC directly. Finally, we propose a training-free alternative to ParZC, which utilize a non-negative weighting to improve the existing ZC proxies.

Node-wise ZC Encoding

To address the significant magnitude differences and variations among node-wise Zero-Cost (ZC) proxies, we employ min-max feature scaling σ. This technique mitigates large condition number issues and reduces variance from disparate feature scales. For the k-th ZC proxy $z_k : \mathcal{S}(N_i) \to \mathbb{R}$ applied to the m-th neural network $\mathcal{N}^{(m)}$ from the search space, the encoding is defined as:

$$\sigma(z_k(\mathcal{N}^{(m)})) := \frac{z_k(\mathcal{N}^{(m)}) - \min(z_k(\mathcal{N}^{(m)}))}{\max(z_k(\mathcal{N}^{(m)})) - \min(z_k(\mathcal{N}^{(m)}))}$$

where z_k represents the k-th node-wise ZC proxy, which is a function that maps the statistics of a node to a real number: $z_k : \mathcal{S}(N_i) \to \mathbb{R}$. $z_k(\mathcal{N}^{(m)}) \in \mathbb{R}^L$ denotes the application of z_k to all L nodes of $\mathcal{N}^{(m)}$. σ denotes the min-max feature scaling operation, which normalizes the ZC statistics to a common scale, facilitating more stable and effective learning in subsequent stages of the model. As illustrated in Fig. 5, just with one batch of data as input to perform forward and backward operations, we can gather the statistic $z_k(\mathcal{N}^{(m)}) \in \mathbb{R}^L$, where the label on the nodes denotes the order of processing. Refer to Supp. A.1 for details.

Mixer Architecture with Bayesian Network

Stacked Multi-layer perceptions (MLPs) can approximate complex nonlinear functions but struggle in capturing intricate, higher-order interactions among input data with high uncertainty. We introduce a novel approach, the Mixer Architecture with Bayesian Network (MABN), as shown in Fig. 5, designed to explicitly model uncertainty by embedding probabilistic relationships within ZC proxy statistics. Given the inherent instability in ZC estimations, our method utilizes the Mixer Architecture to explore inter-segment relationships effectively. By incorporating Bayesian networks, we significantly enhance its ability to assess uncertainty in node-wise ZC proxies.

Bayesian Network. We introduce a Bayesian Network that employs probabilistic backpropagation (Hernández-Lobato and Adams 2015), which can enhance the estimation of the input uncertainty. Each Bayesian layer transforms the input $x \in \mathbb{R}^{N\times L\times P}$ to an output $y \in \mathbb{R}^{N\times L\times O}$ through a linear transformation using Bayesian weights $y = xW_b^T$. The weight matrix W_b is computed using the reparameterization trick:

$$W_b = \mu + \log(1 + e^{\rho}) \cdot \epsilon$$

where μ represents the mean and ρ represents the log distribution variance, and $\epsilon \sim \mathcal{N}(0, I)$ is a random variable sampled from a standard normal distribution. In the Bayesian network, the output Y given an input X and weights W_b is described by the conditional probability $P(Y|X, W_b)$, indicating the probability of observing Y for specified X and W_b. The weights are derived from a posterior distribution $P(W_b|X, Y)$. The process culminates in a probabilistic linear transformation $Y = XW_b^T$, where each forward pass involves integrating over potential linear transformations, weighted according to their posterior probabilities. This method aligns with Bayesian principles, effectively allowing the network to incorporate uncertainty into its predictions. As illustrated in Fig. 5, we incorporate Bayesian Networks before and after the Mixer Architecture to enhance uncertainty modeling of node-wise ZC proxies.

Mixer Architecture. We first apply a linear layer to project the input into a higher-dimensional space X'. We further segment $X' \in \mathbb{R}^{N\times(S\times L)}$ by splitting input into S segments with length of L then we have $X' \in \mathbb{R}^{N\times S\times L}$. In response to improved estimations of ZC proxies, we introduce the Mixer architecture, a concise approach designed to model the complex and nonlinear mapping of node-wise ZC statistics by exploiting inter-segment relationships. The Mixer architecture leverages a segment mixer to achieve these goals. The process begins with a preprocessing phase where a Bayesian Network (BN) assesses segment uncertainty $X_b = XW_b^T$. A layer normalization step follows BN, expressed as $X_b' = \text{LayerNorm}(X_b)$, where X_b denotes the transposed input segments. Subsequently, a Feedforward Network (FFN) is applied to these normalized segments, enhancing cross-segment interaction. This operation is represented as $X_{\text{seg}} = X_b'^T + \text{FFN}\left(X_b'^T\right)$. The segment X_{seg} is then transposed once more and processed through r combination of Linear, ReLU, and Dropout. Following a pooling operation, the segment dimensions are transformed from $\mathbb{R}^{N\times S\times L}$ to $\mathbb{R}^{N\times S}$. A Bayesian Network processes the output in the final stage, enabling precise estimation of the architecture's characteristics.

This methodology within each mixer block significantly bolsters the model's ability to discern and interpret complex inter-segment relations and patterns within the input data. The Bayesian MLP Mixer architecture represents a sophisticated blend of Bayesian inference principles and structured

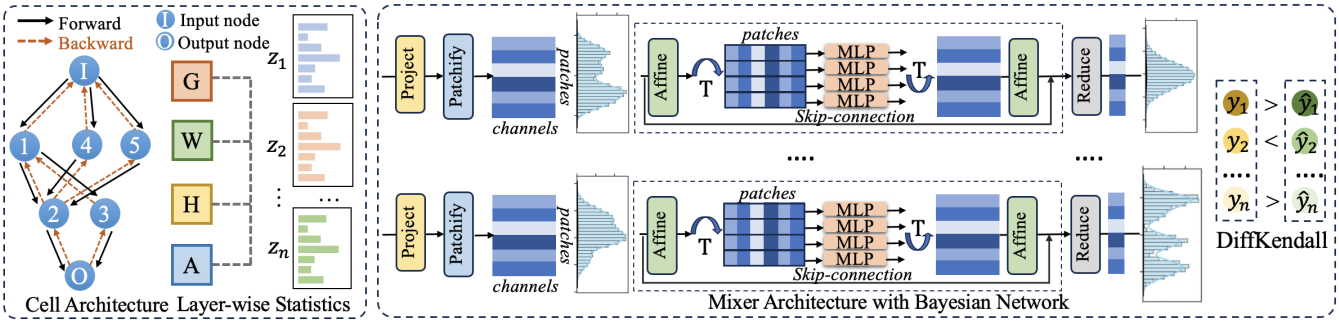


Figure 5: **Framework of ParZC.** Left: Illustration of node-wise ZC proxies. It can extract gradient (G), weight (W), hessian (H), or activation (A) from different nodes. ParZC utilizes these node-wise ZC from different proxies as input. Right: Bayesian network and mixer architecture enhance uncertainty measurement and inter-channel information extraction, with DiffKendall loss monitoring architectural relations.

segment mixing. This innovative approach marks a leap forward from traditional MLP architectures, offering enhanced capabilities in processing and understanding intricate data structures. Our structure exhibits a resemblance to that of MLP-Mixer (Tolstikhin et al. 2021). However, a notable disparity lies in our input methodology. Unlike MLP-Mixer, which splits images into multiple patches, our mixer architecture exclusively relies on probability as its input source.

Differentiable Ranking Optimization

We employ Kendall's tau (Zheng, Zhang, and Huang 2023) to assess the correlation between the rankings produced by zero-shot estimations and the ground truth. However, the standard form of Kendall's tau is not differentiable, complicating its use in gradient-based optimization. To make Kendall's tau differentiable, we proposed DiffKendall, which introduces a sigmoid-based transformation characterized by parameters α, encapsulated in the function $\sigma_\alpha(\Delta) = \text{sigmoid}(\alpha\Delta) - \text{sigmoid}(-\alpha\Delta)$. This transformation smooths the non-differentiable sign function inherent in the original Kendall's Tau computation. The approximation of Kendall's Tau τ_d is:

$$\tau_d = -\frac{1}{\binom{L}{2}} \sum_{i \neq j} \sigma_\alpha(\Delta x_{ij}) \cdot \sigma_\alpha(\Delta y_{ij})$$

where $\binom{L}{2}$ represents the total number of unique element pairs and $\Delta x_{ij} = x_i - x_j, \Delta y_{ij} = y_i - y_j$. This expression encapsulates the concordance and discordance between the ranks of elements in the sequences x and y while maintaining differentiability. In contrast to the pairwise rank loss (Xu et al. 2021b), which relies on the quality of the pairs selected for training, the proposed τ_d offers a broader view of rank correlation by considering both concordant and discordant pairs in sequences.

Non-negative Weighting

Inspired by the distribution patterns learned from MABN, we derive a training-free weighting approach termed ParZC†. This method can be directly applied to various zero-cost (ZC) proxies, resulting in performance enhancements. Specifically, we employ the Sinusoidal Weighting to avoid the instabilities caused by negative weights, maintaining a consistent and interpretable weighting mechanism. This approach allows for a smooth and periodic variation in the weights, which can capture potential patterns or trends in the node importance. The sinusoidal function ensures a continuous weighting scheme, avoiding abrupt changes that could lead to instability or inconsistencies in the ranking process. The weighting scheme is defined as follows:

$$w_i = \sin\left(\frac{0.5 \times i}{2}\right) + 1 \tag{1}$$

where w_i represents the weight assigned to the i-th node. The Sinusoidal function is ranging from 0 to 1. It is worth noting that the proposed weighting scheme is training-free, meaning that it does not require any additional optimization.

Experiments

We conduct a series of experiments across three established NAS benchmarks on CNN and ViT search spaces. Our primary objective is to evaluate the ranking ability of ParZC in comparison with eleven zero-cost proxies and fourteen predictor-based NAS methods. Additionally, we present an in-depth ablation study for key factors.

Datasets and Implementation Details

Datasets. We conduct experiments on various NAS benchmarks with extensive search space including NAS-Bench-101 (NB101) (Ying et al. 2019), NAS-Bench-201 (NB201) (Dong and Yang 2020) and Network Design Spaces (NDS) (Radosavovic et al. 2019) with DARTS (Liu, Simonyan, and Yang 2019)/NASNet (Zoph and Le 2017)/ENAS (Pham et al. 2018a), spanning CIFAR-10 (Krizhevsky, Nair, and Hinton 2014), CIFAR-100 (Krizhevsky 2009), and ImageNet16-120 datasets (Chrabaszcz, Loshchilov, and Hutter 2017). To verify the adaptability of ParZC, we extend the experiment to ViT search space, a.k.a. Autoformer (Chen et al. 2021a), on ImageNet-1k.

Implementation Details For each architecture in the training set, we aggregate their node-wise ZC statistics with

	NB101-CF10		NB201-CF10		NB201-CF100		NB201-IMG16		NDS-DARTS		NDS-NASNet		NDS-ENAS	
	SP	KD	SP	KD	SP	KD	SP	KD	SP	KD	SP	KD	SP	KD
Params	37.0	25.0	72.0	54.0	73.0	55.0	69.0	52.0	67.0	50.0	50.5	36.1	41.0	32.0
FLOPs	36.0	25.0	69.0	50.0	71.0	52.0	67.0	48.0	67.6	50.7	48.1	34.5	41.0	32.0
Fisher (Turner et al. 2020)	-28.0	-20.0	50.0	37.0	54.0	40.0	48.0	36.0	33.7	22.7	-9.2	-4.8	-5.9	-4.1
GradNorm (Abdelfattah et al. 2021)	-25.0	-17.0	58.0	42.0	-63.0	47.0	57.0	42.0	37.5	26.0	-7.1	-3.9	-0.4	-0.1
GraSP (Wang, Zhang, and Grosse 2020)	27.0	18.0	51.0	35.0	54.0	38.0	55.0	39.0	-20.8	-14.7	14.2	8.6	18.4	12.3
SNIP (Lee, Ajanthan, and Torr 2019)	-19.0	-14.0	58.0	43.0	-63.0	47.0	57.0	42.0	42.3	30.0	-0.7	0.9	2.8	2.6
Synflow (Tanaka et al. 2020)	31.0	21.0	73.0	54.0	76.0	57.0	75.0	56.0	49.9	36.4	7.5	5.3	6.3	4.0
NWOT (Mellor et al. 2021)	31.0	21.0	77.0	58.0	80.0	62.0	77.0	59.0	66.3	48.9	44.9	31.7	38.0	28.0
Zen (Lin et al. 2021)	59.0	42.0	35.0	27.0	35.0	28.0	39.0	29.0	49.0	36.1	13.2	10.2	13.5	10.4
ZiCo (Li et al. 2023a)	63.0	46.0	74.0	54.0	78.0	58.0	79.0	60.0	49.5	34.9	22.4	16.7	17.3	12.0
$\text{ParZC}^{\dagger}_{\text{GradNorm}}$	-32.8	-21.8	65.9	48.0	-65.5	46.3	65.5	48.1	45.2	31.8	-12.0	-7.0	-12.5	-8.4
$\text{ParZC}^{\dagger}_{\text{SNIP}}$	-29.2	-21.0	68.1	50.6	-67.3	50.4	67.4	49.5	48.2	33.9	-1.7	0.2	8.7	5.6
$\text{ParZC}^{\dagger}_{\text{Synflow}}$	43.1	30.0	75.6	57.5	75.8	57.7	75.9	56.9	49.9	36.3	7.7	5.4	9.9	7.3
EZNAS (Akhauri et al. 2022)	6.8	4.5	83.0	65.0	82.0	65.0	78.0	61.0	67.0	**56.0**	50.0	**44.0**	63.0	**52.0**
ParZC	**83.2**	**63.7**	**90.4**	**70.6**	**91.1**	**74.3**	**87.9**	**69.9**	**67.8**	50.3	**54.9**	38.5	**69.0**	50.6

Table 1: **Spearman (SP) and Kendall's Tau (KD) coefficients (%) of various ZC proxies** across NAS benchmarks NAS-Bench-101 (NB101), NAS-Bench-201 (NB201), and NDS for CIFAR-10, CIFAR-100, and ImageNet16-120 datasets.

Train Samples	S_{100}	S_{172}	S_{424}	S_{4236}
Train Ratio	0.02%	0.04%	0.1%	1%
Test Samples	all	all	100	all
SPOS (Guo et al. 2019)	-	-	19.6	-
FairNAS (Chu, Zhang, and Xu 2021)	-	-	23.2	-
NAO (Luo et al. 2018)	50.1	56.6	70.4	77.5
NP (Wen et al. 2020)	39.1	54.5	71.0	76.9
Arch2Vec (Yan et al. 2020)	43.5	51.1	56.1	59.6
GATES (Ning et al. 2020b)	60.5	65.9	66.6	82.2
ReNAS (Xu et al. 2021b)	-	-	63.4	81.6
CTNAS (Chen et al. 2021c)	-	-	75.1	-
TNASP (Lu et al. 2021)	60.0	66.9	75.2	82.0
TA-GATES (Ning et al. 2022)	-	-	-	66.8
PINAT (Lu et al. 2023)	67.9	71.5	**80.1**	84.6
ParZC	**69.3**	**71.7**	79.7	**85.3**

Table 2: Kendall's Tau (%) for Predictor-based NAS Algorithms on CIFAR-10, evaluated on NAS-Bench-101.

Train Samples	S'_{78}	S'_{156}	S'_{469}	S'_{781}	S'_{1563}
Train Ratio	0.05%	1%	3%	5%	10%
Test Samples	all	all	all	all	all
NAO (Luo et al. 2018)	46.7	49.3	47.0	52.2	52.6
NP (Wen et al. 2020)	34.3	41.3	58.4	63.4	64.6
Arch2Vec (Yan et al. 2020)	54.2	57.3	60.1	60.6	60.5
GraphTrans	40.9	55.0	59.4	58.8	67.3
Graphormer	50.5	63.0	68.0	71.9	77.6
TNASP (Lu et al. 2021)	53.9	58.9	64.0	68.9	72.4
NASBOWL (Ru et al. 2021)	**66.7**	59.1	-	72.6	76.2
EZNAS (Akhauri et al. 2022)	55.1	57.3	61.1	60.9	62.4
PINAT (Lu et al. 2023)	54.9	63.1	70.6	76.1	78.4
ParZC	64.6	**70.6**	**80.6**	**83.2**	**85.5**

Table 3: Kendall's Tau (%) for Predictor-based NAS Algorithms on CIFAR-10, evaluated on NAS-Bench-201.

Synflow (Tanaka et al. 2020), SNIP (Lee, Ajanthan, and Torr 2019), GradNorm (Abdelfattah et al. 2021), etc. We adopt Kendall's Tau (KD) and Spearman (SP) to measure the rank correlation between predicted and actual accuracy. For NB101 and NB201, we utilize Adam optimizer with a learning rate 1e-4 and weight decay of 1e-3. The training batch size is 10, and the evaluation batch size is 50. The training epochs on NB101, NB201, and NDS are 150, 200, and 296, respectively. Specifically for NDS, we mainly conduct experiments on NASNet, DARTS, and ENAS search spaces to verify the ranking ability of ParZC. DiffKendall is a loss function when training ParZC with $\alpha = 0.5$. We detail the training settings in the Supp. A.3 for different search spaces. All of the experiments are conducted on RTX 4090Ti and PyTorch (Paszke et al. 2019) framework. The hyper-parameters of our proposed MABN, such as hidden size, dropout rate, and embedding dimension, are finely tuned using Bayesian optimization with Optuna (Akiba et al. 2019) (For more details, please refer to the Supp. A.4).

Comparison with Zero-cost Proxies

We report the rank correlation with SP and KD on three NAS benchmarks in Tab. 1, including NB101, NB201 and NDS. The results on NB101 and NB201 are obtained from previous methods (Abdelfattah et al. 2021; Akhauri et al. 2022; Zheng et al. 2024), while the results on NDS are evaluated using the official implementation.

Results of ParZC. We compare our ParZC with three kinds of zero-shot NAS methods: size-based, pruning-based, and theory-based proxies. The size-based proxies serve as the baseline, encompassing FLOPs and Params, achieving competitive performance. Pruning-based proxies are inspired by pruning metrics like Fisher (Turner et al. 2020), GradNorm (Abdelfattah et al. 2021), GraSP (Wang, Zhang, and Grosse 2020), L2Norm (Abdelfattah et al. 2021), Plain (Mozer and Smolensky 1988), SNIP (Lee, Ajanthan, and Torr 2019), Synflow (Tanaka et al. 2020), which also achieve relatively good performance but most of them still fail to outperform the baseline. Theory-based proxies such as NWOT (Mellor et al. 2021), Zen (Lin et al. 2021), and ZiCo (Li et al. 2023a), generally achieve better performance than pruning-based proxies but also show poor correlation on challenging search space such as NASNet and ENAS.

Algorithm	Test Accuracy (%)			Cost (GPU Sec.)	Method	Applicable Space
	CIFAR-10	CIFAR-100	ImageNet16-120			
ResNet (He et al. 2016)	93.97	70.86	43.63	-	manual	-
ENAS (Pham et al. 2018b)	93.76±0.00	71.11±0.00	41.44±0.00	15120	reinforce	C
GDAS (Dong and Yang 2019)	93.44±0.06	70.61±0.21	42.23±0.25	8640	gradient	C
DrNAS (Chen et al. 2021b)	93.98±0.58	72.31±1.70	44.02±3.24	14887	gradient	C
NWOT (Mellor et al. 2021)	92.96±0.81	69.98±1.22	44.44±2.10	306	training-free	C & D
TE-NAS (Chen, Gong, and Wang 2021)	93.90±0.47	71.24±0.56	42.38±0.46	1558	training-free	C
KNAS (Xu et al. 2021a)	93.05	68.91	34.11	4200	training-free	C & D
NASI (Shu et al. 2022a)	93.55±0.10	71.20±0.14	44.84±1.41	120	training-free	C
GradSign (Zhang and Jia 2022)	93.31±0.47	70.33±1.28	42.42±2.81	-	training-free	C & D
HNAS ($\mathcal{M}_{\text{SNIP}}$) (Shu et al. 2022b)	93.94±0.02	71.49±0.11	46.07±0.14	2976	hybrid	C & D
HNAS ($\mathcal{M}_{\text{GraSP}}$) (Shu et al. 2022b)	94.13±0.13	72.59±0.82	46.24±0.38	3148	hybrid	C & D
HNAS ($\mathcal{M}_{\text{Trace}}$) (Shu et al. 2022b)	94.07±0.10	72.30±0.70	45.93±0.37	3006	hybrid	C & D
EZNAS (Akhauri et al. 2022)	93.63±0.12	69.82±0.16	43.47±0.20	86400	hybrid	D
ParZC	**94.36**±0.01	**73.49**±0.02	**46.34**±0.04	3000	hybrid	C & D

Table 4: **Comparison of NAS algorithms on NAS-Bench-201**. The result of ParZC is reported with mean and standard deviation of 3 independent runs. "C" and "D" denotes continuous and discrete search space.

Overall, EZNAS (Akhauri et al. 2022) demonstrate its superiority for ranking ability among all search space. Our proposed ParZC surpasses the baseline by a large margin and achieves competitive results across all search spaces.

Results of ParZC†. As presented in Tab. 1, ParZC† demonstrates quantifiable improvements in Spearman (SP) and Kendall's Tau (KD) correlation coefficients over GradNorm, SNIP, and Synflow across various NAS benchmarks. The implementation of a non-negative weighting schema has yielded statistically significant enhancements in performance metrics, indicating a more efficient prediction of network performance during the architecture search process. Besides Sinusoidal Weighting (SIN), we provide the results of other strategies including Linear Decrease (LD), Linear Increase (LI), and Exponential Decrease (ED). Our analysis reveals that while each strategy uniquely influences ranking changes, the Sinusoidal Weighting (SIN) consistently outperforms the others in enhancing model performance.

Comparison with Predictor-based NAS

To make a fair comparison, we present the KD on NB101 and NB201 compared to training-based NAS with the same data-splitting settings. Tab. 2 and 3 present the Kendall's Tau obtained from the same data splits, denoted as $S_{\#samples}$ for the NB101 benchmark, utilizing 0.02% to 1% of the entire search space, and $S'_{\#samples}$ for the NB201 benchmark, utilizing 0.05% to 10% of the entire search space. We compare our ParZC with one-shot (Guo et al. 2019; Chu, Zhang, and Xu 2021) and predictor-based NAS (Wen et al. 2020; Lu et al. 2023, 2021). Note that we incorporate the operation encoding and adjacency matrix following PINAT (Lu et al. 2023) into ParZC to compare. For NB101, results demonstrate that our proposed ParZC exhibits a remarkable ability in ranking architectures on NB101, which not only outperforms the one-shot based NAS like SPOS (Guo et al. 2019) and FairNAS (Chu, Zhang, and Xu 2021) but also surpasses the SOTA transformer-based predictors (Chen et al. 2021c; Lu et al. 2021, 2023). For NB201, our ParZC increases the SP by around 10%. With only 78 samples (0.05% of search space), our ParZC can achieve better performance than PINAT (Lu et al. 2023) with 156 samples, which denotes that our ParZC contains additional information over the architecture and is complementary to existing predictor-based methods.

Search Results on NAS-Bench-201

We present a thorough evaluation of various NAS algorithms, focusing on their performance on the test set on CIFAR-10/100 and ImageNet16-120 in NB201, as detailed in Tab. 4. To substantiate the effectiveness and efficiency of our proposed ParZC, we conduct comparative analyses with several baseline approaches, including optimization-based (Dong and Yang 2020), one-shot (Pham et al. 2018b; Dong and Yang 2019; Chen et al. 2021b), zero-shot (Mellor et al. 2021; Chen, Gong, and Wang 2021; Xu et al. 2021a; Shu et al. 2022a; Zhang and Jia 2022) and automatic designed proxies (Akhauri et al. 2022; Dong et al. 2024). We categorize various NAS methods into five types: evolution, random search, reinforcement, gradient, and training-free. Hybrid denotes a combination of these types. For example, EZNAS belongs to training-free and evolution categories. Our ParZC uniquely integrates gradient and training-free approaches. As shown in Tab. 4, ParZC outperforms training-based and training-free baselines by consistently selecting superior architectures. ParZC requires only 50 GPU minutes for its training process, as it estimates performance in batches, which iscomparable to even zero-shot NAS methods like KNAS (Xu et al. 2021a). Furthermore, our ParZC attains SOTA performance on NB201 with minimal variance, showcasing its efficiency and effectiveness.

Experiments on Vision Transformer

We present the performance of the searched Vision Transformer architecture on the ImageNet-1k dataset in Tab. 5.

Algorithms	Param (M)	Top-1 (%)	GPU Days
Deit-Ti (Touvron et al. 2020)	5.7	72.2	-
TNT-Ti (Han et al. 2021)	6.1	73.9	-
ViT-Ti (Dosovitskiy et al. 2021)	5.7	74.5	-
PVT-Tiny (Wang et al. 2021)	13.2	75.1	-
ViTAS-C (Su et al. 2021)	5.6	74.7	32
AutoFormer-Ti (Chen et al. 2021a)	5.7	74.7	24
TF-TAS-Ti (Zhou et al. 2022)	5.9	75.3	0.5
ParZC	6.1	**75.5**	**0.05**

Table 5: Comparison with ViT on ImageNet-1k.

	LD		LI		ED		SIN	
	SP	KD	SP	KD	SP	KD	SP	KD
GradNorm	-9.2	-5.1	-11.4	-7.3	-8.3	-6.1	7.9	6
SNIP	-14.7	-9.3	-12.7	-7.4	-8.9	-6.5	10.1	7.6
Synflow	1.7	1.2	0.5	0.8	0.9	0.3	2.6	3.5
L2Norm	-1.2	-3.1	0.1	0.7	-0.9	-2.1	2	1.5

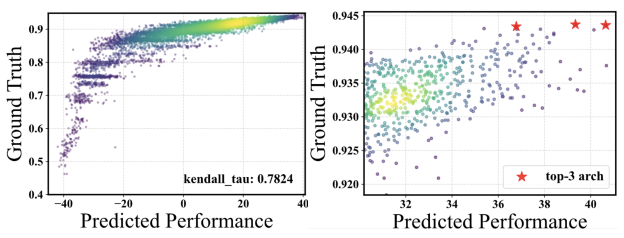
Table 6: Ranking Changes with weighting strategy.

Following TF-TAS (Zhou et al. 2022), we utilize the ground-truth labels from AutoFormer (Chen et al. 2021a; Wei et al. 2023). Utilizing a relatively small dataset comprising 1,000 samples, our ParZC algorithm demonstrates the capability to identify a high-performance architecture within an impressively short span of 0.05 GPU days. This efficiency level aligns with that of leading one-shot NAS methods. The results in the table show that the architecture identified by ParZC not only competes with but also exceeds the performance of the SOTA TF-TAS-Ti model (Zhou et al. 2022) while maintaining a comparable number of parameters.

Ablation Study

Design Choices. We dissect the contributions of components in ParZC using KD and SP on NB201 in Tab. 7. The results are reported with mean and std of 3 runs with different seeds. Integrating all components yields optimal KD and SP coefficients of 69.98% and 87.90%, respectively. Only mixer architecture is a competitive baseline with 62.29% KD and 80.19% SP. Compared with baseline MLP, mixer architecture can achieve 7.65% higher in KD and 6.69% higher in SP. We also observe that NP (Wen et al. 2020) can further increase the predictive capability(1.2%↑ KD).

Effectiveness of DiffKendall. We present an ablation study evaluating the impact of Mean Squared Error (MSE) Loss, Rank Loss, and Differentiable Kendall's Tau (DiffKendall)



(a) All architectures (b) Top architectures

Figure 6: Rank correlation between ParZC and ground truth

NP	Mixer	BN	MLP	KD(%)	SP(%)
✓	-	-	-	34.29±0.42	45.61±12.89
-	✓	-	-	62.29±3.31	80.19±1.45
-	-	-	✓	54.64±8.60	73.50±13.78
-	-	✓	-	50.12±5.35	69.09±8.86
✓	✓	-	-	59.79±2.41	78.84±4.61
✓	-	✓	-	54.81±1.22	74.24±1.51
-	✓	✓	-	67.69±4.21	86.03±3.53
✓	✓	✓	-	**68.89**±1.40	**87.17**±0.64

Table 7: **Ablation study of design choices** in ParZC using 78 samples on NB201. NP: Neural Predictor (Wen et al. 2020), Mixer: Mixer Architecture, BN: Bayesian Network, MLP: Multi-layer Perceptron.

MSE	Rank	DiffKendall	KD(%)	SP(%)
✓	-	-	65.69	85.04
-	✓	-	65.64	84.89
✓	✓	-	64.62	83.92
✓	-	✓	65.56	84.75
-	✓	✓	64.41	83.85
✓	✓	✓	66.36	85.51
-	-	✓	**66.83**	**85.97**

Table 8: **Ablation study of loss functions** using 178 Samples on NB101. MSE: Mean Squared Error Loss, Rank: Ranking-Based Loss Function (Xu et al. 2021b).

on NB101, as shown in Tab. 8. The results show that employing DiffKendall as the single loss function achieves the best rank correlation with 66.83% KD and 85.97% SP.

Sample Size. Tab. 3 and Fig. 4 illustrate the Kendall's Tau correlation coefficients obtained with varying sample sizes. Our proposed ParZC method outperforms the SOTA counterparts, EZNAS (Akhauri et al. 2022) and HNAS (Shu et al. 2022b), by a large margin.

Visual Inspection. In Fig. 6, we visualize the rank correlation of ParZC on NB201. The fig. (a) displays the correlation across the entire search space, exhibiting a remarkable Kendall Tau of 78.24%. We investigate the top-tier architecture within the search space in Fig. (b). Notably, we mark the top architectures with a star symbol, which validates our ParZC's effectiveness in identifying architectures with superior performance. Additionally, we provide the visualization of Bayesian Network in Fig.9 of Supp. E.1.

Conclusion

We present a Parametric Zero-Cost Proxies (ParZC) framework to address the critical issue of homogeneous treatment of node-wise ZC statistics. Specifically, we propose Mixer Architecture with Bayesian Network (MABN) to explore and quantify the inherent uncertainties in the node-wise ZC statistics. To enhance the ranking capabilities of ParZC, we further introduce DiffKendall to handle the discrepancy in ranking architectures. Extensive experiments on various NAS benchmarks and Vision Transformer demonstrate that our ParZC can outperform ZC proxies and predictor-based NAS methods. We aspire for our work to catalyze the design and development of ZC proxies, thereby fostering innovation and progress within the research community.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 62272122, the Guangzhou Municipal Joint Funding Project with Universities and Enterprises under Grant No. 2024A03J0616, the Hong Kong RIF grant under Grant No. R6021-20, and Hong Kong CRF grants under Grant No. C7004-22G and C6015-23G.

References

Abdelfattah, M. S.; Mehrotra, A.; Dudziak, Ł.; et al. 2021. Zero-Cost Proxies for Lightweight NAS. In *ICLR*.

Akhauri, Y.; Munoz, J. P.; Jain, N.; et al. 2022. EZNAS: Evolving Zero-Cost Proxies For Neural Architecture Scoring. In *NeurIPS*.

Akiba, T.; Sano, S.; Yanase, T.; et al. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *SIGKDD*.

Cavagnero, N.; Robbiano, L.; Caputo, B.; et al. 2023. FreeREA: Training-Free Evolution-Based Architecture Search. In *WACV*, 1493–1502.

Chen, M.; Peng, H.; Fu, J.; et al. 2021a. Autoformer: Searching transformers for visual recognition. In *ICCV*, 12270–12280.

Chen, W.; Gong, X.; and Wang, Z. 2021. Neural Architecture Search on ImageNet in Four GPU Hours: A Theoretically Inspired Perspective. In *ICLR*.

Chen, X.; Wang, R.; Cheng, M.; et al. 2021b. DrNAS: Dirichlet Neural Architecture Search. In *ICLR*.

Chen, X.; Xie, L.; Wu, J.; et al. 2019. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, 1294–1303.

Chen, Y.; Guo, Y.; Chen, Q.; et al. 2021c. Contrastive neural architecture search with neural architecture comparators. In *CVPR*, 9502–9511.

Chrabaszcz, P.; Loshchilov, I.; and Hutter, F. 2017. A Downsampled Variant of ImageNet as an Alternative to the CIFAR datasets. *ArXiv*, abs/1707.08819.

Chu, X.; Zhang, B.; and Xu, R. 2021. FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search. In *ICCV*.

Dong, P.; Li, L.; Tang, Z.; Liu, X.; Pan, X.; Wang, Q.; and Chu, X. 2024. Pruner-Zero: Evolving Symbolic Pruning Metric from Scratch for Large Language Models. In *ICML*.

Dong, P.; Li, L.; and Wei, Z. 2023. DisWOT: Student Architecture Search for Distillation WithOut Training. In *CVPR*.

Dong, P.; Li, L.; Wei, Z.; et al. 2023a. EMQ: Evolving Training-free Proxies for Automated Mixed Precision Quantization. In *ICCV*, 17076–17086.

Dong, P.; Niu, X.; Li, L.; et al. 2023b. RD-NAS: Enhancing One-shot Supernet Ranking Ability via Ranking Distillation from Zero-cost Proxies. *ICASSP*.

Dong, X.; and Yang, Y. 2019. Searching for a Robust Neural Architecture in Four GPU Hours. In *CVPR*.

Dong, X.; and Yang, Y. 2020. NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search. In *ICLR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29: 1189–1232.

Guo, Z.; Zhang, X.; Mu, H.; et al. 2019. Single Path One-Shot Neural Architecture Search with Uniform Sampling. In *ECCV*, 544–560.

Han, K.; Xiao, A.; Wu, E.; et al. 2021. Transformer in Transformer. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; et al. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

He, X.; Yao, J.; Wang, Y.; Tang, Z.; Cheung, K. C.; See, S.; Han, B.; and Chu, X. 2023. Nas-lid: Efficient neural architecture search with local intrinsic dimension. In *AAAI*.

Hernández-Lobato, J. M.; and Adams, R. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *ICML*, 1861–1869. PMLR.

Hu, Y.; Wang, X.; Li, L.; et al. 2021. Improving One-Shot NAS with Shrinking-and-Expanding Supernet. *Pattern Recognition*.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. In *CVPR*.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2014. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*.

Lee, N.; Ajanthan, T.; and Torr, P. H. 2019. SNIP: Single-shot network pruning based on connection sensitivity. In *ICLR*.

Li, G.; Yang, Y.; Bhardwaj, K.; et al. 2023a. ZiCo: Zero-shot NAS via inverse Coefficient of Variation on Gradients. In *ICLR*.

Li, L.; Dong, P.; Wei, Z.; and Yang, Y. 2023b. Automated knowledge distillation via monte carlo tree search. In *ICCV*, 17413–17424.

Li, L.; Sun, H.; Li, S.; Dong, P.; Luo, W.; Xue, W.; Liu, Q.; and Guo, Y. 2025. Auto-GAS: automated proxy discovery for training-free generative architecture search. In *ECCV*.

Li, L.; Wei, Z.; Dong, P.; Luo, W.; Xue, W.; fei Liu, Q.; and Guo, Y.-T. 2024. AttnZero: Efficient Attention Discovery for Vision Transformers. In *ECCV*.

Lin, M.; Wang, P.; Sun, Z.; et al. 2021. Zen-NAS: A Zero-Shot NAS for High-Performance Image Recognition. In *ICCV*.

Liu, H.; Simonyan, K.; and Yang, Y. 2019. DARTS: Differentiable Architecture Search. In *ICLR*.

Lu, S.; Hu, Y.; Wang, P.; et al. 2023. PINAT: A Permutation INvariance Augmented Transformer for NAS Predictor. *AAAI*.

Lu, S.; Li, J.; Tan, J.; et al. 2021. TNASP: A Transformer-based NAS Predictor with a Self-evolution Framework. *NeurIPS*.

Luo, R.; Tan, X.; Wang, R.; et al. 2020. Semi-supervised neural architecture search. *NeurIPS*, 33: 10547–10557.

Luo, R.; Tian, F.; Qin, T.; et al. 2018. Neural architecture optimization. In *NeurIPS*.

Mellor, J.; Turner, J.; Storkey, A.; et al. 2021. Neural Architecture Search without Training. In *ICML*.

Mozer, M. C.; and Smolensky, P. 1988. Skeletonization: A Technique for Trimming the Fat from a Network via Relevance Assessment. In *NeurIPS*.

Ning, X.; Tang, C.; Li, W.; et al. 2021. Evaluating Efficient Performance Estimators of Neural Architectures. In *NeurIPS*.

Ning, X.; Zheng, Y.; Zhao, T.; et al. 2020a. A generic graph-based neural architecture encoding scheme for predictor-based nas. In *ECCV*.

Ning, X.; Zheng, Y.; Zhao, T.; et al. 2020b. A generic graph-based neural architecture encoding scheme for predictor-based nas. In *ECCV*, 189–204.

Ning, X.; et al. 2022. TA-GATES: An Encoding Scheme for Neural Network Architectures. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *NeurIPS*.

Paszke, A.; Gross, S.; Massa, F.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*.

Pham, H.; Guan, M.; Zoph, B.; et al. 2018a. Efficient Neural Architecture Search via Parameters Sharing. In *ICML*.

Pham, H.; Guan, M. Y.; Zoph, B.; et al. 2018b. Efficient Neural Architecture Search via Parameter Sharing. In *ICML*.

Radosavovic, I.; Johnson, J.; Xie, S.; et al. 2019. On Network Design Spaces for Visual Recognition. In *ICCV*.

Ru, B.; Wan, X.; Dong, X.; et al. 2021. Interpretable Neural Architecture Search via Bayesian Optimisation with Weisfeiler-Lehman Kernels. In *ICLR*.

Shen, L.; Tang, Z.; Wu, L.; et al. 2024. Hot Pluggable Federated Learning. In *FL@FM-NeurIPS workshop*.

Shu, Y.; Cai, S.; Dai, Z.; et al. 2022a. NASI: Label- and Data-agnostic Neural Architecture Search at Initialization. In *ICLR*.

Shu, Y.; Dai, Z.; Wu, Z.; et al. 2022b. Unifying and Boosting Gradient-Based Training-Free Neural Architecture Search. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *NeurIPS*.

Su, X.; You, S.; Xie, J.; et al. 2021. ViTAS: Vision Transformer Architecture Search. In *ECCV*.

Tanaka, H.; Kunin, D.; Yamins, D. L.; et al. 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. *NeurIPS*.

Tang, Z.; Kang, X.; Yin, Y.; et al. 2024a. FusionLLM: A Decentralized LLM Training System on Geo-distributed GPUs with Adaptive Compression. arXiv:2410.12707.

Tang, Z.; Zhang, Y.; Dong, P.; et al. 2024b. FuseFL: One-Shot Federated Learning through the Lens of Causality with Progressive Model Fusion. In *NeurIPS*.

Tang, Z.; Zhang, Y.; Shi, S.; et al. 2024c. FedImpro: Measuring and Improving Client Update in Federated Learning. In *ICLR*.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*.

Touvron, H.; Cord, M.; Douze, M.; et al. 2020. Training data-efficient image transformers & distillation through attention. In *ICML*.

Turner, J.; Crowley, E. J.; O'Boyle, M.; et al. 2020. BlockSwap: Fisher-guided Block Substitution for Network Compression on a Budget. In *ICLR*.

Wang, C.; Zhang, G.; and Grosse, R. 2020. Picking Winning Tickets Before Training by Preserving Gradient Flow. In *International Conference on Learning Representations*.

Wang, H.; Ge, C.; Chen, H.; et al. 2023. PreNAS: Preferred One-Shot Learning Towards Efficient Neural Architecture Search. In *ICML*.

Wang, W.; Xie, E.; Li, X.; et al. 2021. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *ICCV*, 548–558.

Wei, Z.; Li, L.; Dong, P.; et al. 2023. Auto-Prox: Training-Free Vision Transformer Architecture Search via Automatic Proxy Discovery. *ArXiv*, abs/2312.09059.

Wen, W.; Liu, H.; Chen, Y.; et al. 2020. Neural predictor for neural architecture search. In *ECCV*.

White, C.; Neiswanger, W.; and Savani, Y. 2021. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *AAAI*, volume 35, 10293–10301.

Xu, J.; Zhao, L.; Lin, J.; et al. 2021a. KNAS: Green Neural Architecture Search. In *ICML*.

Xu, Y.; Wang, Y.; Han, K.; et al. 2021b. Renas: Relativistic evaluation of neural architecture search. In *CVPR*, 4411–4420.

Yan, S.; Zheng, Y.; Ao, W.; et al. 2020. Does unsupervised architecture representation learning help neural architecture search? *NeurIPS*, 33: 12486–12498.

Ying, C.; Klein, A.; Christiansen, E.; et al. 2019. NAS-Bench-101: Towards Reproducible Neural Architecture Search. In *ICML*.

Zhang, Z.; and Jia, Z. 2022. GradSign: Model Performance Inference with Theoretical Insights. In *ICLR*.

Zheng, H.; Liu, K.-H.; Fedorov, I.; et al. 2024. SiGeo: Sub-One-Shot NAS via Information Theory and Geometry of Loss Landscape. In *KDD*.

Zheng, K.; Zhang, H.; and Huang, W. 2023. DiffKendall: a novel approach for few-shot learning with differentiable kendall's rank correlation. *NeurIPS*, 36: 49403–49415.

Zhou, Q.; Sheng, K.; Zheng, X.; et al. 2022. Training-free Transformer Architecture Search. In *CVPR*, 10894–10903.

Zoph, B.; and Le, Q. V. 2017. Neural architecture search with reinforcement learning. In *ICLR*.