NEURAL INFORMATION PROCESSING SYSTEMS

NAR-Former V2: Rethinking Transformer for Universal Neural Network Representation Learning

Yun Yi[1], Haokui Zhang[*], Rong Xiao, Nannan Wang[*], Xiaoyu Wang

西安电子科技大学 XIDIAN UNIVERSITY
Xi'an, China

intellifusion 云天励飞
Shenzhen, China

西北工业大学 NORTHWESTERN POLYTECHNICAL UNIVERSITY
Xi'an, China

[*]Corresponding authors.

[1]This work was done while Yun Yi was an intern at Intellifusion.

Overview

NEURAL INFORMATION PROCESSING SYSTEMS

Modeling and learning the representation of neural networks

↓

predict networks' attributes
(without running the actual estimation procedures)

↓

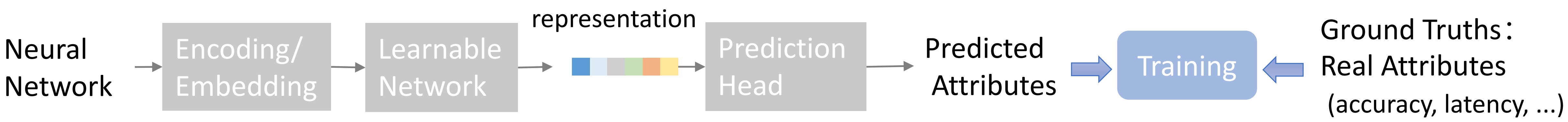
Improving the efficiency of network design and deployment

In this paper, we proposed NAR-Former V2,

- It can handle cell-structured networks as well as learn representations for the entire network
- We achieve this by incorporating graph-specific properties into the vanilla Transformer and introducing a graph-aided attention-based Transformer block.

Background

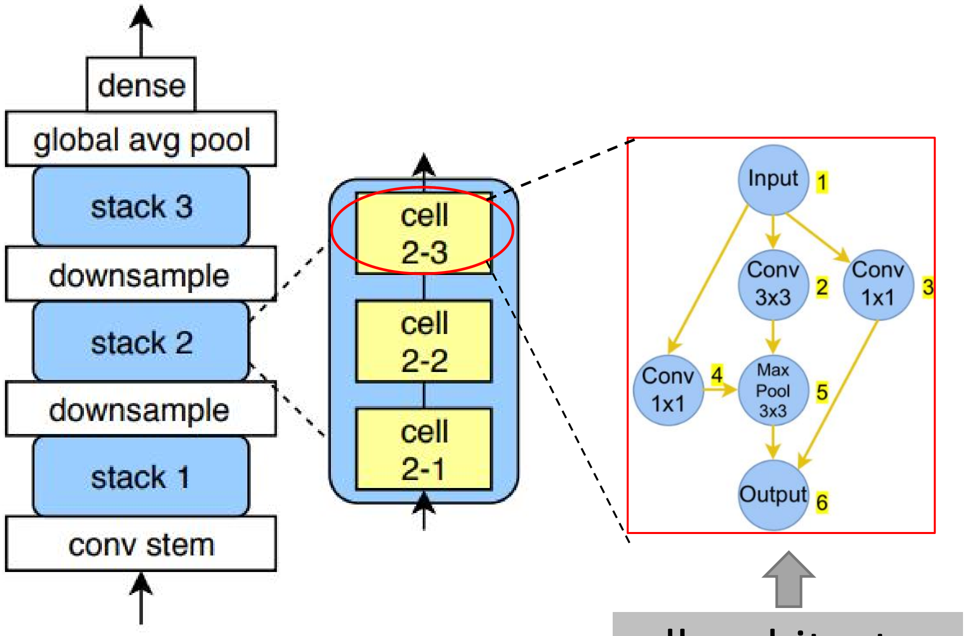
What is neural network representation learning



Motivation

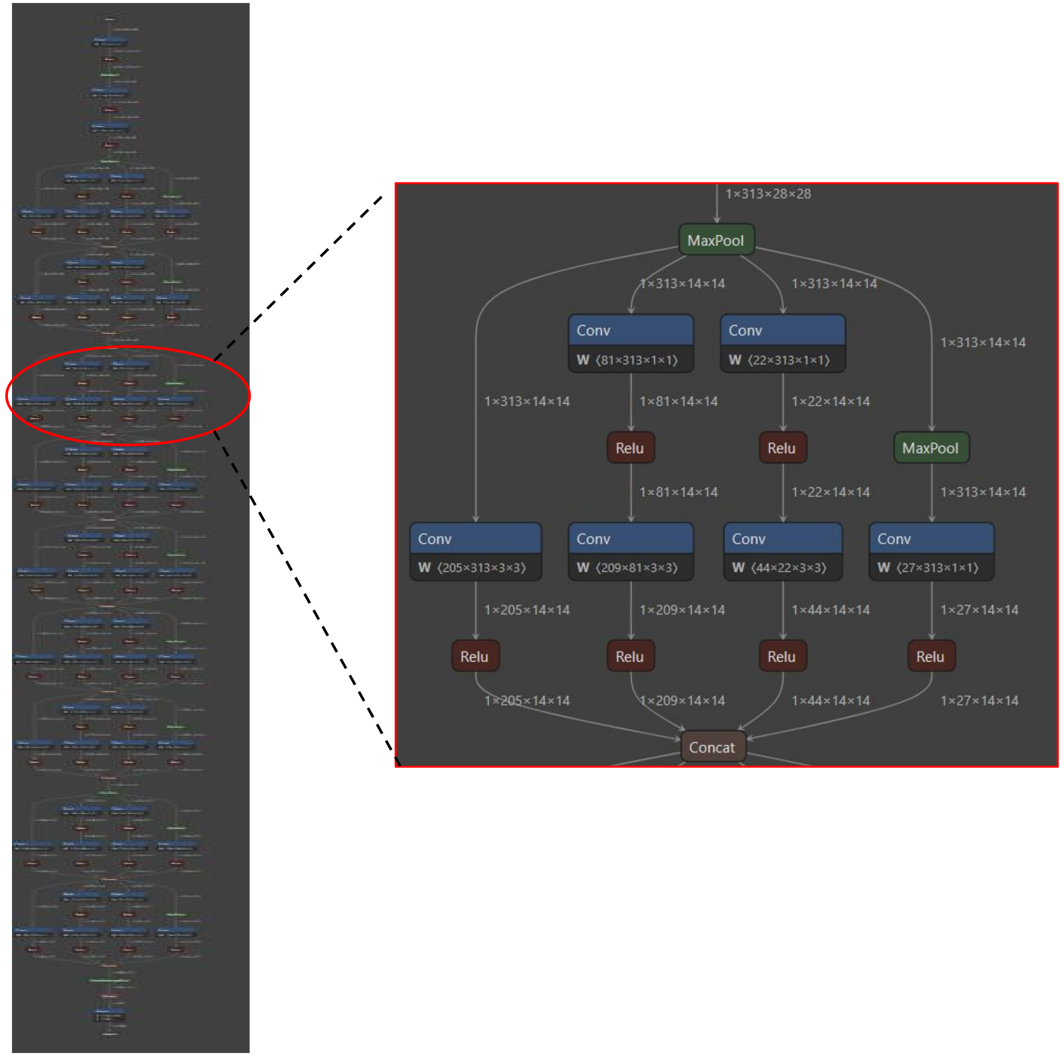
NEURAL INFORMATION PROCESSING SYSTEMS

Neural network forms that may need to be encoded in reality:



Architecure in NAS-Bench-101

cell architecture



entire deep neural network

Motivation

NEURAL INFORMATION PROCESSING SYSTEMS

Existing methods reached the SOTA only in specific scenario

Transformer based methods

achieve leading performance on encoding the architectures of cells.

GNN-based methods

perform better when dealing with complete DNNs, or when the depth of the input data is unseen during training.

- We need to reconsider the two representation learning models
- propose an unified method

Motivation

Comparison

	Transformer	GNN
Information Aggregation	global	neighbouring
Feed-Forward Network	✔	-

Analyses

Why Transformer performs poor when encoding entire DNNs?

global modeling based on self-attention

↓

excessive sensitivity

↓

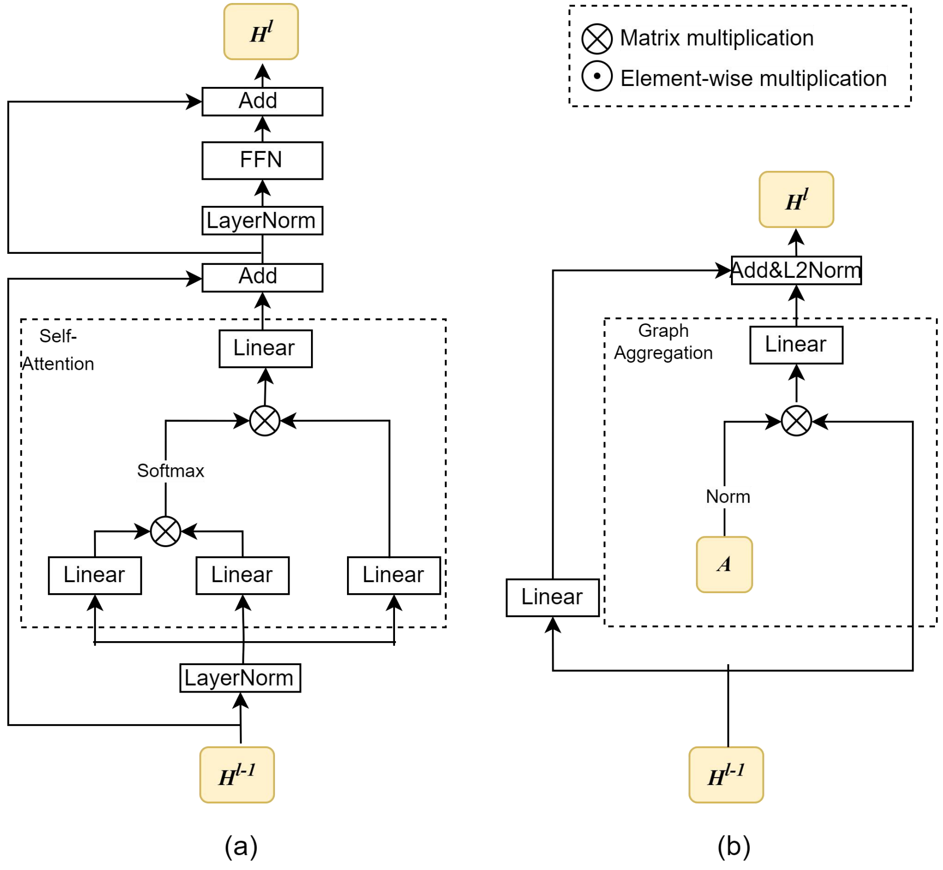
subtle variation can affect the whole representation

↓

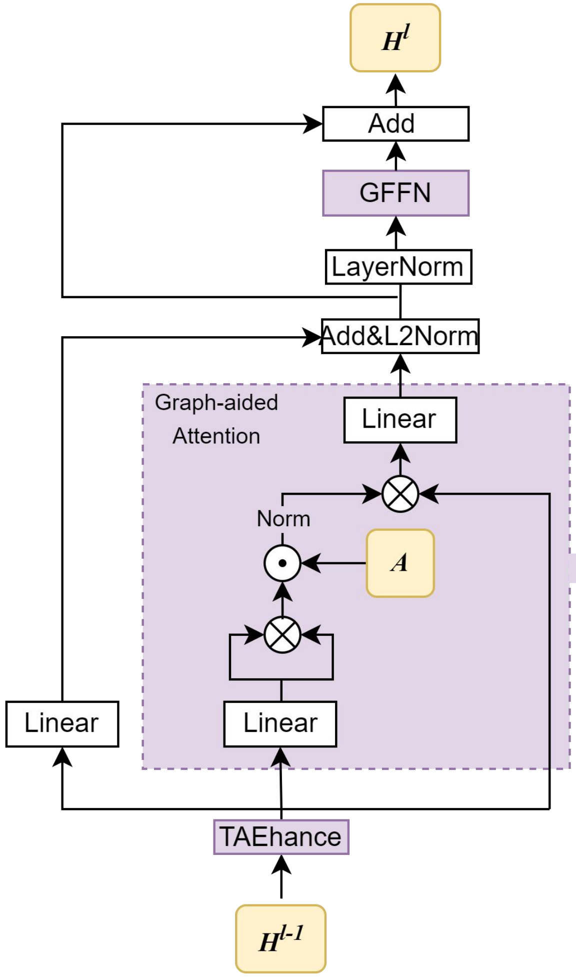
baised toward fitting the training data

↓

poor performance on unseen data



Transformer block vs. GNN block



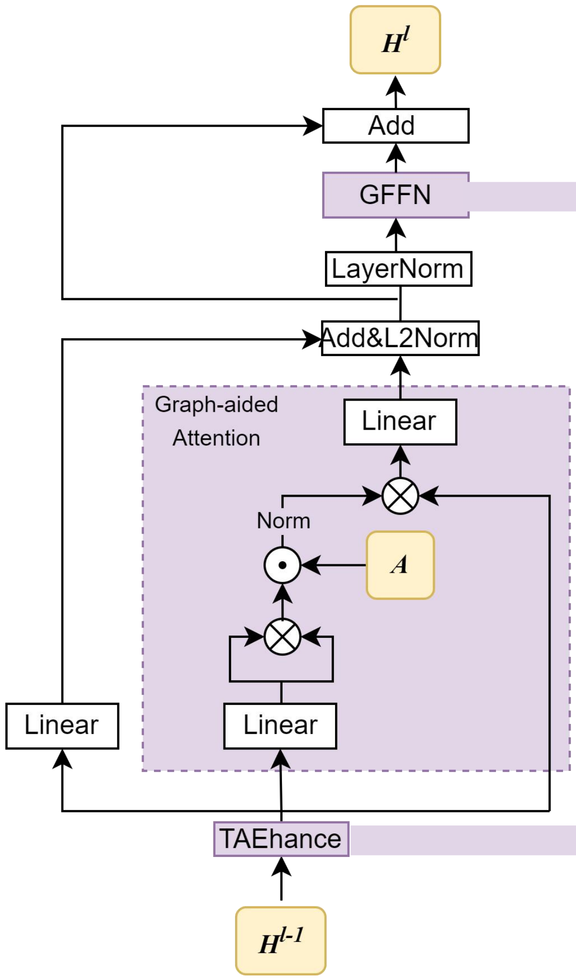
(c)

Graph-aided attention

Employ the adjacency matrix
to govern the attention calculation range

$$X^l = \mathrm{Sigmoid}(W_q^l \widetilde{H}^l + b_q^l),$$
$$S^l = (X^l X^{lT} / \sqrt{d}) \boxed{\odot A,}$$
$$Z^l = W_a^l(\mathrm{Norm}(S^l)\widetilde{H}^l) + b_a^l.$$

Proposed Method: NAR-Former-V2



(c)

Grouped Feed-Forward Network

Introduce group linear transformation into the original FFN to reduce the parameters and futher avoid overfitting problem.

Type-Aware enhancement module

Use the number of connected layers in each layer to assist the model in learning the type of layer.

$$\mathrm{TAEnhance}(H^{l-1}, D) = \mathrm{Sigmoid}(W_d^l \boxed{D} + b_d^l) \odot H^{l-1}.$$

Proposed Method: NAR-Former-V2

Be similar to that in NAR-Former

Be built upon the proposed improved transformer block

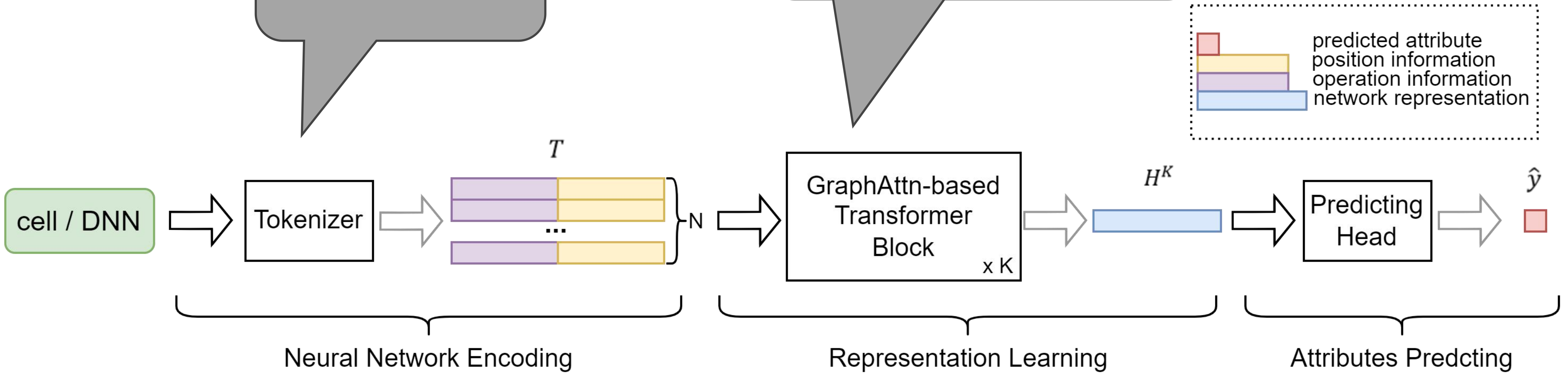


Table 1: Latency prediction on NNLQP [21]. Training and test sets have the same distribution.

Test Model	MAPE↓			Acc(10%)↑		
	NAR-Former [47]	NNLP [21] avg / best	Ours avg / best	NAR-Former [47]	NNLP [21] avg / best	Ours avg / best
All	22.37%	3.47% / 3.44%	3.07% / 3.00%	35.00%	95.25% / 95.50%	96.41% / 96.30%
AlexNet	26.25%	6.37% / 6.21%	6.18% / 5.97%	27.00%	81.75% / 84.50%	81.90% / 84.00%
EfficientNet	13.91%	3.04% / 2.82%	2.34% / 2.22%	45.50%	98.00% / 97.00%	98.50% / 100.0%
GoogleNet	16.00%	4.18% / 4.12%	3.63% / 3.46%	39.00%	93.70% / 93.50%	95.95% / 95.50%
MnasNet	15.76%	2.60% / 2.46%	1.80% / 1.70%	33.00%	97.70% / 98.50%	99.70% / 100.0%
MobileNetV2	15.19%	2.47% / 2.37%	1.83% / 1.72%	39.00%	99.30% / 99.50%	99.90% / 100.0%
MobileNetV3	16.88%	3.50% / 3.43%	3.12% / 2.98%	36.00%	95.35% / 96.00%	96.75% / 98.00%
NasBench201	43.53%	1.46% / 1.31%	1.82% / 1.18%	55.50%	100.0% / 100.0%	100.0% / 100.0%
SqueezeNet	24.33%	4.03% / 3.97%	3.54% / 3.34%	23.00%	93.25% / 93.00%	95.95% / 96.50%
VGG	23.64%	3.73% / 3.63%	3.51% / 3.29%	26.50%	95.25% / 96.50%	95.85% / 96.00%
ResNet	28.18%	3.34% / 3.25%	3.11% / 2.89%	25.50%	98.40% / 98.50%	98.55% / 99.00%

NEURAL INFORMATION PROCESSING SYSTEMS

Table 2: Latency prediction on NNLQP [21]. "Test Model = AlexNet" means that only AlexNet models are used for testing, and the data from the other 9 model families are used for training. The best results refer to the lowest MAPE and corresponding ACC (10%) in 10 independent experiments. *: obtained based on the released code without using its fine-tuning step.

Metric	Test Model	FLOPs	FLOPs +MAC	nn-Meter [55]	TPU [17]	BRP-NAS [12]	NAR-Former [47]*	NNLP [21] (avg / best)	Ours (avg / best)
MAPE↓	AlexNet	44.65%	15.45%	7.20%	10.55%	31.68%	46.28%	10.64% / 9.71%	24.28% / 18.29%
	EfficientNet	58.36%	53.96%	18.93%	16.74%	51.97%	29.34%	21.46% / 18.72%	13.20% / 11.37%
	GoogleNet	30.76%	32.54%	11.71%	8.10%	25.48%	24.71%	13.28% / 10.90%	6.61% / 6.15%
	MnasNet	40.31%	35.96%	10.69%	11.61%	17.26%	26.70%	12.07% / 10.86%	7.16% / 5.93%
	MobileNetV2	37.42%	35.27%	6.43%	12.68%	20.42%	25.74%	8.87% / 7.34%	6.73% / 5.65%
	MobileNetV3	64.64%	57.13%	35.27%	9.97%	58.13%	33.99%	14.57% / 13.17%	9.06% / 8.72%
	NasBench201	80.41%	33.52%	9.57%	58.94%	13.28%	105.71%	9.60% / 8.19%	9.21% / 7.89%
	ResNet	21.18%	18.91%	15.58%	20.05%	15.84%	40.37%	7.54% / 7.12%	6.80% / 6.44%
	SqueezeNet	29.89%	23.19%	18.69%	24.60%	42.55%	74.59%	9.84% / 9.52%	7.08% / 6.56%
	VGG	69.34%	66.63%	19.47%	38.73%	30.95%	44.26%	7.60% / 7.17%	15.40% / 14.26%
	Average	47.70%	37.26%	15.35%	21.20%	30.76%	45.17%	11.55% / 10.27%	10.55% / 9.13%
Acc(10%)↑	AlexNet	6.55%	40.50%	75.45%	57.10%	15.20%	7.60%	59.07% / 64.40%	24.65% / 28.60%
	EfficientNet	0.05%	0.05%	23.40%	17.00%	0.10%	15.15%	25.37% / 28.80%	44.01% / 50.20%
	GoogleNet	12.75%	9.80%	47.40%	69.00%	12.55%	24.35%	36.30% / 48.75%	80.10% / 83.35%
	MnasNet	6.20%	9.80%	60.95%	44.65%	34.30%	20.90%	55.89% / 61.25%	73.46% / 81.60%
	MobileNetV2	6.90%	8.05%	80.75%	33.95%	29.05%	20.70%	63.03% / 72.50%	78.45% / 83.80%
	MobileNetV3	0.05%	0.05%	23.45%	64.25%	13.85%	16.05%	43.26% / 49.65%	68.43% / 70.50%
	NasBench201	0.00%	10.55%	60.65%	2.50%	43.45%	0.00%	60.70% / 70.60%	63.13% / 71.70%
	ResNet	26.50%	29.80%	39.45%	27.30%	39.80%	13.25%	72.88% / 76.40%	77.24% / 79.70%
	SqueezeNet	16.10%	21.35%	36.20%	25.65%	11.85%	11.40%	58.69% / 60.40%	75.01% / 79.25%
	VGG	4.80%	2.10%	26.50%	2.60%	13.20%	11.45%	71.04% / 73.75%	45.21% / 45.30%
	Average	7.99%	13.20%	47.42%	34.40%	21.34%	14.09%	54.62% / 60.65%	62.70% / 67.40%

Experiments

Table 3: Accuracy prediction on NAS-Bench-101 [48]. "SE" denotes the self-evolution strategy proposed by TNASP [26].

Backbone	Method	Training Samples 0.1% (424)	Training Samples 0.1% (424)	Training Samples 1% (4236)
		Test Samples 100	Test Samples all	Test Samples all
CNN	ReNAS [46]	0.634	0.657	0.816
LSTM	NAO [27]	0.704	0.666	0.775
	NAO+SE	0.732	0.680	0.787
GNN	NP [43]	0.710	0.679	0.769
	NP + SE	0.713	0.684	0.773
	CTNAS [3]	0.751	-	-
Transformer	TNASP [26]	0.752	0.705	0.820
	TNASP + SE	0.754	0.722	0.820
	NAR-Former [47]	0.801	0.765	**0.871**
	NAR-Former V2	**0.802**	**0.773**	0.861

Table 4: Accuracy prediction on NAS-Bench-201 [10]. "SE" denotes the self-evolution strategy proposed by TNASP [26].

Backbone	Model	Training Samples (781) 5%	Training Samples (1563) 10%
LSTM	NAO [27]	0.522	0.526
	NAO + SE	0.529	0.528
GNN	NP [43]	0.634	0.646
	NP + SE	0.652	0.649
Transformer	TNASP [26]	0.689	0.724
	TNASP + SE	0.690	0.726
	NAR-Former [47]	0.849	**0.901**
	NAR-Former V2	**0.874**	0.888

Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

Table 5: Ablation studies on NNLQP [21]. "PE" denotes position encoding.

Row	Structure	Op Type	Op Attributes	Graph-Attn	GFFN	TA-Enhance	MAPE↓	Acc(10%)↑	Acc(5%)↑
1(Baseline)	GNN	One-hot	Real Num	-	-	-	3.48	95.26	77.80
2	GNN	PE	PE	-	-	-	3.43(-0.05)	95.11(-0.15)	79.58(+1.78)
3	GNN	One-hot	PE	-	-	-	3.33(-0.15)	95.57(+0.31)	80.19(+2.39)
4	Transformer	One-hot	PE	✓	-	-	3.20(-0.28)	96.00(+0.74)	81.86(+4.06)
5	Transformer	One-hot	PE	✓	✓	-	3.20(-0.28)	96.06(+0.80)	81.76(+3.96)
6	Transformer	One-hot	PE	✓	✓	✓	3.07(-0.41)	96.41(+1.15)	82.71(+4.91)

Conclusion

Overview:
we combine the strengths of Transformer and GNN to develop a universal neural network representation learning model, which is capable of effectively processing models of varying scales, ranging from several layers to hundreds of layers.

Experiments:
(1) Complete DNNs encoding & latency prediciton: our proposed method surpasses the GNN-based method NNLP by a significant margin on the NNLQP dataset.
(2) Cell encoding & accuracy prediciton: our method achieves highly comparable performance to other state-of-the-art methods on NASBench101 and NASBench201 datasets.

Future work:
We will focus on optimizing the design of the representation learning framework and applying it to a broader range of practical applications. Such as using the proposed model to search for the best mixed precision model inference strategies.

西安電子科技大學 XIDIAN UNIVERSITY intellifusion 云天励飞 西北工業大學 NORTHWESTERN POLYTECHNICAL UNIVERSITY NEURAL INFORMATION PROCESSING SYSTEMS

Thanks for your listening!

IIP Lab: https://iip-xdu.github.io

Intellifusion: https://www.intellif.com/

Codes link: https://github.com/yuny220/NAR-Former-V2