A Transformation to Operation-On-Node Search Space

According to the description in Section 3.1, the cells in operation-on-edge search spaces (*e.g.*, the NAS-Bench-201 and DARTS search spaces) are transformed to the operation-on-node version, as illustrated in Figure 5.

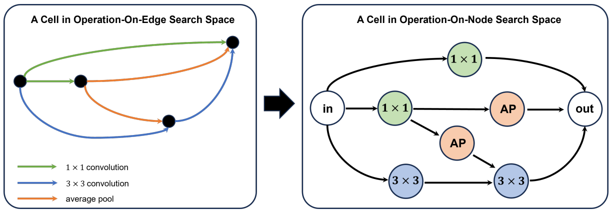


Figure 5: Transformation to the operation-on-node version.

B Searching RNNs in the PTB dataset

To show the generalization capability of CAP, we employ CAP to search for well-performing RNNs in the PTB dataset using NAS-Bench-NLP. According to Table 6, CAP can also help find promising architectures on the language modeling task. Compared with previous state-of-the-art predictor-based NAS methods, CAP can search for better architecture at the same query budget.

Method	Test PPL (log)	Query Num	Searching Strategy
BANANAS	4.64	100	predictor+BO
NPENAS	4.61	100	predictor+BO
CAP(ours)	**4.59**	100	predictor+RS

Table 6: Comparison with other predictor-based NAS methods on CIFAR-10 using the DARTS search space.

C More Results of Ablation Study

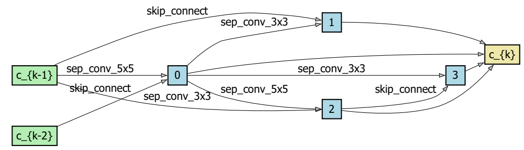
To further display the significance of the proposed context-aware pretext task, we compare the searching results of CAP with/without pre-training in Table 7. Obviously, the tailored pretext task helps the predictor-based NAS discover better architectures consistently in different search spaces.

Method	NB101	NB201-C10	NB201-C100	NB201-IMAGE16-120	DARTS
w/o pre-training	93.98	94.27	71.20	44.54	97.33
w/ pre-training	**94.18**	**94.34**	**73.41**	**46.44**	**97.58**

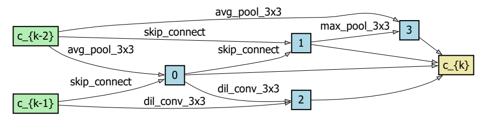
Table 7: The searching results of CAP with/without pre-training in NAS-Bench-101, NAS-Bench-201 and DARTS search spaces.

D The Best Searched Architecture on CIFAR-10 in DARTS

The best searched architecture on CIFAR-10 in DARTS is shown in Figure 6.



(a) Normal Cell



(b) Reduction Cell

Figure 6: The top-1 searched architecture on CIFAR-10.

E Experimental Settings

To perform the context-aware self-supervised task, a large portion of architectures are sampled randomly from the search space. For closed domain search spaces, 90% of the architectures are involved. Specifically, 381,262 architectures are used in NAS-Bench-101 and 14,063 architectures are utilized in NAS-Bench-201. For the DARTS search space, 1,000,000 architectures are randomly sampled for pre-training. The detailed settings are shown as follows.

Context-Aware Self-Supervised Task. Table 8 reports the hyperparameter settings of the context-aware self-supervised task in NAS-Bench-101, NAS-Bench-201 and DARTS search spaces. In general, the main GIN serves as a core component of the neural predictor while the auxiliary GIN is only used in the pre-training stage.

Hyperparameter	NB-101	NB-201	DARTS
Batch Size	1024	1024	1024
Epochs	300	300	300
K	1	1	2
R	2	3	3
Optimizer	AdamW	AdamW	AdamW
Model	GIN	GIN	GIN
Learning Rate (Main GIN)	0.001	0.001	0.001
Weight Decay (Main GIN)	0.0001	0.0001	0.0001
Dropout (Main GIN)	0.5	0.5	0.5
Layers (Main GIN)	3	3	3
Learning Rate (Aux GIN)	0.001	0.001	0.001
Weight Decay (Aux GIN)	0.0001	0.0001	0.0001
Dropout (Aux GIN)	0.5	0.5	0.5
Layers (Aux GIN)	2	2	2

Table 8: The hyperparameter settings of the context-aware self-supervised task in NAS-Bench-101, NAS-Bench-201 and DARTS search spaces. K and R determine the size of the central subgraph and context graphs in each architecture.

Performance Prediction. The hyperparameter settings of the proposed neural predictor in the performance prediction stage are shown in Table 9. Despite the various fine-tuning methods, their difference mainly lies in the decision of which

Hyperparameter	**NB-101**	**NB-201**	**DARTS**
Epochs	200	200	250
Optimizer	AdamW	AdamW	AdamW
Learning Rate	0.001	0.001	0.001
Weight Decay	0.001	0.001	0.001
Dropout	0.15	0.15	0.15
Encoder Layers	3	3	3
Model Dims	128	128	128
Model	GIN	GIN	GIN

Table 9: The hyperparameter settings of training the neural predictor in NAS-Bench-101, NAS-Bench-201 and DARTS search spaces.

encoder layers to freeze and other settings are the same. Consequently, we only report the settings of the partially fine-tuning method here.