AIO-P: Expanding Neural Performance Predictors Beyond Image Classification

Keith G. Mills[1], Di Niu[1], Mohammad Salameh[2], Weichen Qiu[1], Fred X. Han[2], Puyuan Liu[2], Jialin Zhang[3], Wei Lu[2] and Shangling Jui[3]

[1]University of Alberta [2]Huawei Technologies Canada [3]Huawei Kirin Solution, Shanghai, China

Open-source link: https://github.com/Ascend-Research/AIO-P

Neural Architecture Search (NAS) for Computer Vision (CV) tasks primarily target Classification.

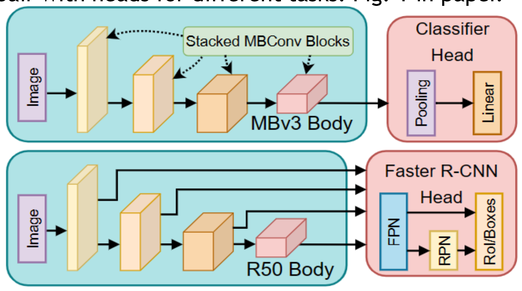
However, there are other tasks such as Pose Estimation, Object Detection, Segmentation, etc.

These tasks are more complex and resource-intensive than Classification as they have larger network heads and deal with larger images.

We propose AIO-P, or **All-In-One Predictors** for multi-task performance evaluation featuring cross-task predictions transferable across search spaces.

- Apply *K*-Adapters to GNN predictors, which allows us to infuse task-specific information.
- Propose a pseudo-labeling sampling technique to build training data for the *K*-Adapters.
- Use scaling techniques, which make use of FLOPs, to augment predictor labels.
- Verify efficacy of AIO-P for cross-task prediction on Human Pose Estimation, Object Detection, and Instance/Semantic/Panoptic Segmentation.
- Demonstrate transferability on model zoos for Classification and Semantic Segmentation.

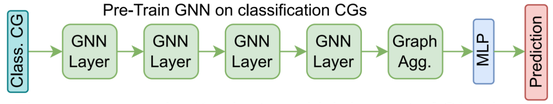
Network bodies from classification search spaces can pair with heads for different tasks. Fig. 1 in paper.



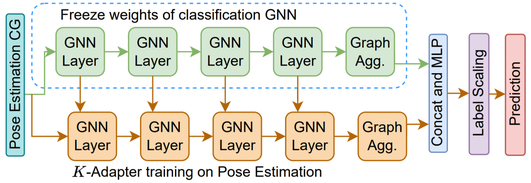
K-Adapters for Knowledge Infusion

We use *K*-Adapters, originally introduced in Natural Language Processing tasks, to instill task-specific information into AIO-P.

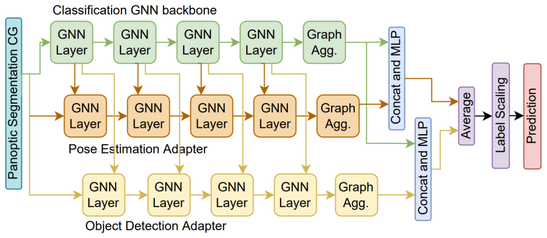
1. We start with a GNN predictor pre-trained on classification:



2. Then, we append a *K*-Adapter, which is a set of GNN layers and aggregation mechanism, to the existing predictor. The *K*-Adapter is trained on networks from a new task. The weights of the original GNN are frozen. Here, our task is Pose Estimation.

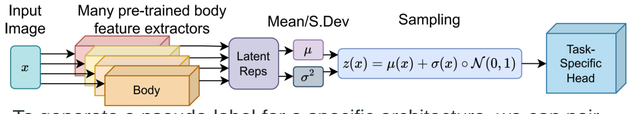


3. For downstream inference on a target task, e.g., something we do not have a lot of data on, we can combine the knowledge of all *K*-Adapters by averaging their predictions.



Pseudo-Labeling to Train *K*-Adapters

Training networks on larger CV tasks, like those involving MS-COCO, is costly in terms of time and computational power. Instead, we train a shared head to cover an entire search space.



To generate a pseudo-label for a specific architecture, we can pair the shared head with that architecture and perform fine-tuning.

Experimental Setup and Results

We train AIO-P using *K*-Adapters for Pose Estimation and Object Detection, then use it to estimate performance for several tasks:

- Human Pose Estimation (LSP and MPII datasets).
- Object Detection (OD).
- Instance (IS), Semantic (SS), and Panoptic Segmentation (PS).

	ProxylessNAS		
Task	GNN	+Eqs. 4 & 5	AIO-P
LSP	27.27 ± 0.39%	0.72 ± 0.14%	**0.70** ± 0.23%
MPII	8.10 ± 0.38%	**0.34** ± 0.07%	0.42 ± 0.13%
OD	59.53 ± 0.41%	1.15 ± 0.46%	**0.63** ± 0.09%
IS	62.00 ± 0.34%	0.93 ± 0.18%	**0.52** ± 0.14%
SS	53.07 ± 0.37%	0.71 ± 0.22%	**0.50** ± 0.06%
PS	56.19 ± 0.35%	0.76 ± 0.07%	**0.50** ± 0.10%

We measure the performance of AIO-P in terms of Mean Absolute Error (MAE; above) and SRCC (below). We can obtain less than 1% MAE on all tasks, going as low as 0.5% on SS and PS, while achieving strong SRCC above 0.65 without any fine-tuning.

	ProxylessNAS		
Task	GNN	+Eqs. 4 & 5	AIO-P
LSP	0.593 ± 0.02	0.561 ± 0.05	**0.698** ± 0.01
MPII	0.711 ± 0.01	**0.767** ± 0.02	0.753 ± 0.01
OD	0.558 ± 0.06	0.471 ± 0.11	**0.781** ± 0.03
IS	0.599 ± 0.07	0.211 ± 0.10	**0.831** ± 0.02
SS	0.487 ± 0.03	0.262 ± 0.18	**0.735** ± 0.02
PS	0.562 ± 0.00	0.119 ± 0.12	**0.732** ± 0.03

Label Scaling for Task Transferability

Performance is measured differently across CV tasks, e.g., accuracy for Classification, mAP for Object Detection, mIoU for Semantic Segmentation and PCK for 2D Human Pose Estimation.

AIO-P is designed for cross-task prediction, while each *K*-Adapter learns knowledge and labels from a specific task. To overcome this, we use standardization to learn a unitless understanding of architecture performance. Furthermore, FLOPs are related to model and task complexity, so we use a FLOPs-based transform to further augment the labels:

$$y_F = y \cdot (\mathrm{Log}_{10}(F+1)+1)^{-1}$$

Transferability using Open-Source Model Zoos

To demonstrate the applicability of AIO-P, we consider several open-source model zoos for Semantic Segmentation and Image Classification. These are small sets of neural networks, like Inception or EfficientNets.

Model Zoo	#Archs	AIO-P w/o Eq. 5	AIO-P
DeepLab-ADE20k	5	0.127 ± 0.255	**0.991** ± 0.016
DeepLab-Pascal	6	0.392 ± 0.088	**0.939** ± 0.035
DeepLab-Cityscapes	8	0.572 ± 0.031	**0.925** ± 0.024
Slim-ResNets	6	-0.577 ± 0.183	**0.920** ± 0.106
Slim-Inception	5	-0.700 ± 0.316	**0.980** ± 0.040
Slim-MobileNets	5	-0.500 ± 0.000	**0.400** ± 0.535
Slim-EfficientNets	8	**1.000** ± 0.000	**1.000** ± 0.000

Even though AIO-P has not been trained on any of these networks, using our FLOPs-based target transform, we can achieve near perfect rank correlation (SRCC) on DeepLab architectures, perfect SRCC=1.0 for EfficientNets and high SRCC for other classification nets.

Application to NAS

Furthermore, we can pair AIO-P with a search algorithm and use it to optimize neural network architectures.

We consider a simple algorithm that edits the primitive operations of a network, such as convolution kernel size, channels, etc., through 1-edit mutations. We apply this search algorithm and AIO-P to a proprietary mobile Facial Recognition (FR) network.

	Full	Simple	Lighted	Dark	FLOPs
Base Model Pr	**96.3%**	**98.7%**	**97.9%**	96.5%	563M
AIO-P Search Pr	96.1%	**98.7%**	**97.9%**	**96.7%**	486M
Base Model Rc	**91.9%**	**98.3%**	**96.8%**	92.6%	563M
AIO-P Search Rc	91.1%	98.2%	96.6%	**93.2%**	486M

Using AIO-P, we can find an architecture that is smaller with over 13% FLOPs reduction, yet still maintains or exceeds the original precision (Pr) and recall (Rc).

HUAWEI HISILICON

UNIVERSITY OF ALBERTA QUAECUMQUE VERA