Aux-NAS: Exploiting Auxiliary Labels with Negligibly Extra Inference Cost

Yuan Gao, Weizhong Zhang, Wenhan Luo, Lin Ma, Jin-Gang Yu, Gui-Song Xia, Jiayi Ma

Motivation

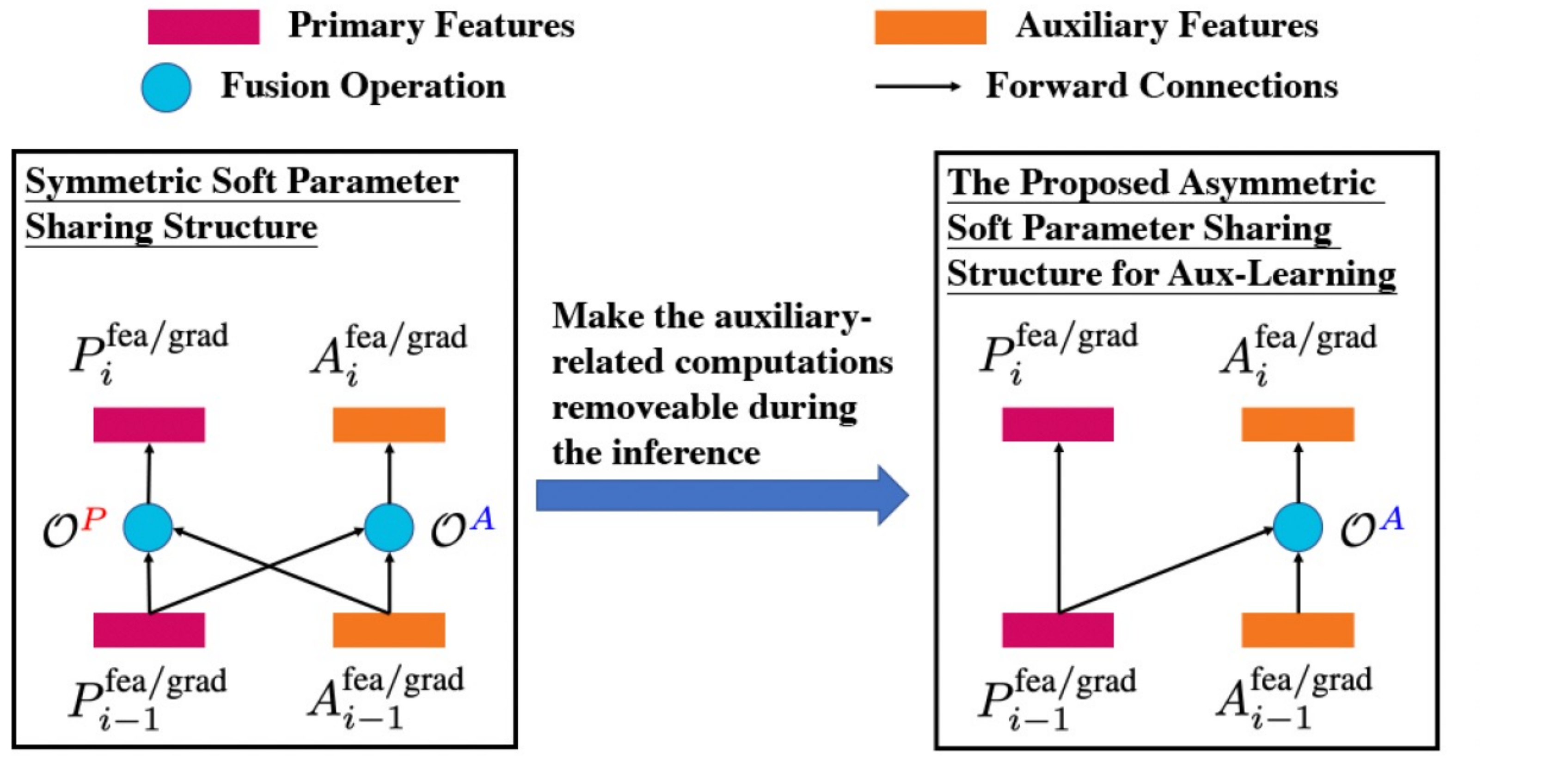
- Exploiting **auxiliary tasks** to boost the performance of the **primary task**;
- Preserving **a single task inference cost** of the primary task.

Key Ideas

- To **avoid negative transfer** between primary and auxiliary tasks:
  - ➢ **Architecture-based methods with soft parameter sharing** is applied.
- To achieve **a single task inference cost** of the primary task:
  - ➢ We design **an asymmetric network architecture** that produces **switchable networks** between the training (more complex) and the inference (more efficient) phases

The Asymmetric Architecture

Soft Parameter Sharing for MTL — **The proposed Asymmetric Architecture**

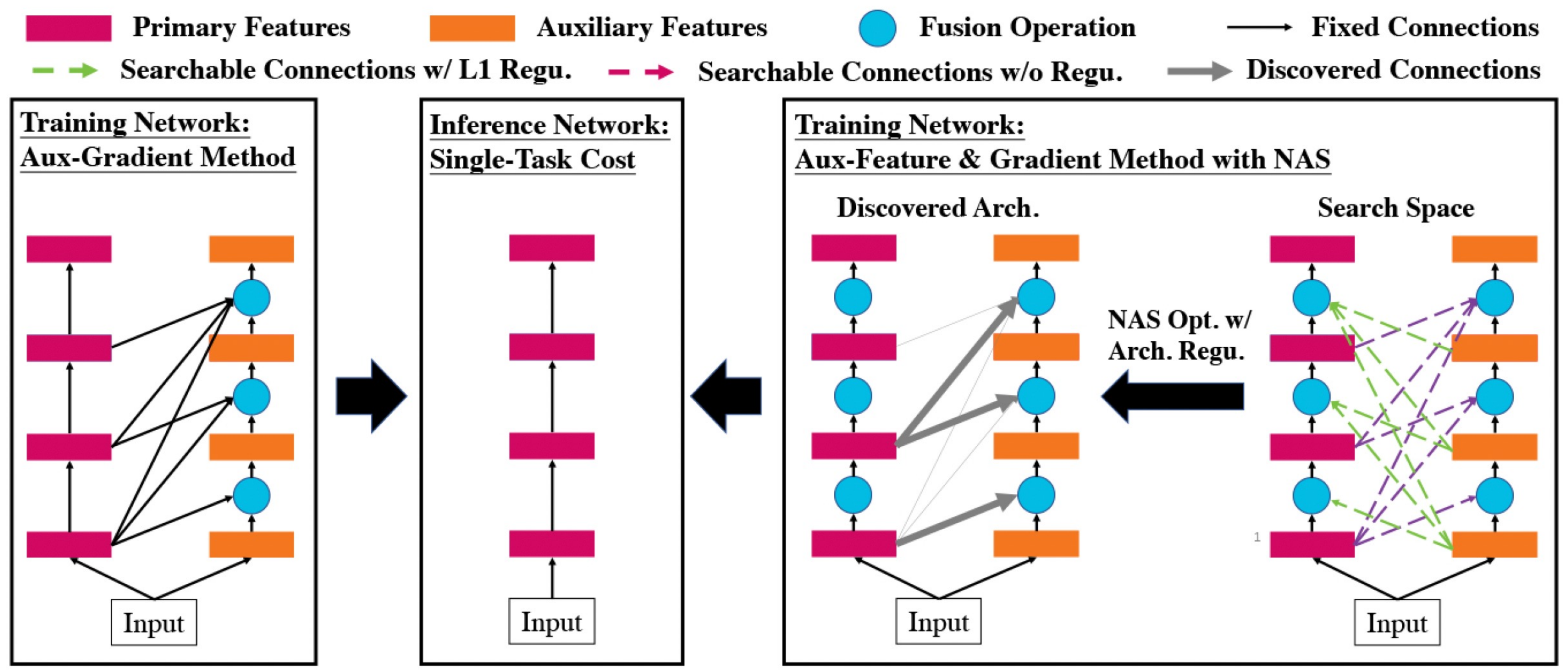


Our method follows the **Right** Subfigure:

- **Only exploit the auxiliary gradients (rather than features)** as **additional regularization for the primary task**.
- We can **remove the auxiliary computations** when **inferencing the primary task** (since the gradients are no longer required during the inference).

The Network Architecture & Optimization

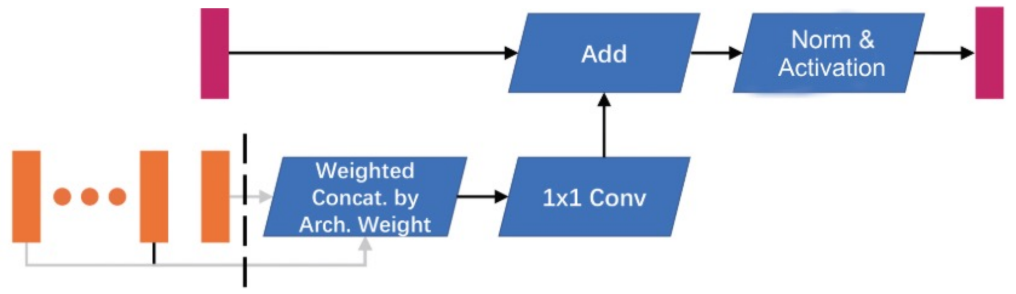
- Based on **the asymmetric architecture design**, we implement our model with two methods, which differ from:
  - ➢ **What auxiliary information we exploit;**
  - ➢ **how we optimize the network architecture.**



- **Method 1: Aux-G (Basic, Left)**
  - ➢ Directly using **the auxiliary gradients** to train the primary task.
  - ➢ Directly exploiting **the asymmetric architecture**.
- **Method 2: Aux-NAS (Advanced, Right)**
  - ➢ Using both **the auxiliary gradients and features**.
  - ➢ Using **Neural Arch Search (NAS)** to optimize the network so that it converges to **an asymmetric architecture with only primary-to-auxiliary connections.**
- Both of our methods converge such that **auxiliary computations can be safely removed**, leading to an inference architecture depicted in the **Middle**.
- For **Method 2: Aux-NAS**, we implement **regularized NAS** with ***L1 Norm* on all the aux-to-prim architecture weights α^P** to gradually prune them out.

$$\min_{\alpha^P,\alpha^A,w} \mathcal{L}^P(\mathbf{P}(\alpha^P,w)) + \mathcal{L}^A(\mathbf{A}(\alpha^A,w)) + \mathcal{R}(\alpha^P), \quad \text{with} \quad \mathcal{R}(\alpha^P) = \lambda\|\alpha^P\|_1,$$

- **The proposed fusion operator**. The above regularized NAS objective enables to cut off through the dash line



Features

- Our method is **general** w.r.t.:
  1. ***Task Combinations***, i.e.,
     Pixel Labeling Tasks: Semantic Seg., Normal & Disp. Pred.
     Image Level Tasks: Object & Scene Classification.
  2. ***Networks***, **CNNs**: VGG & ResNet; **Transformers**: ViT-Base.
  3. ***Datasets***: NYUv2, CityScapes, Taskonomy.
- Our method can be integrated with existing ***Multi-Task Optimizations*** methods, e.g., PCGrad, DWA, etc.
- Our method scales to more auxiliary tasks ***Linearly***.

Experiments

All the experiments demonstrate significant improvements w.r.t. SOTAs.

VGG-16, Seg (Aux) + Disparity (Aux) Tasks

CityScapes Primary: Seg	(%) (↑) mIoU	PAcc
Single	68.3	94.5
Aux-Head	70.0	94.6
Adashare	70.3	94.7
Adashare-Aux	70.1	94.8
Aux-G-Stage	70.1	94.8
Aux-G-Layer	70.2	94.8
Aux-NAS	**71.1**	**95.0**

ResNet-50, Obj. Cls. (Prim) + Scene Cls. (Aux)

Taskonomy Primary: Object Cls.	(%) (↑) Top-1	Top-5
Single	34.3	65.9
Aux-Head	34.7	66.6
Adashare	35.9	67.1
Adashare-Aux	36.3	67.7
Aux-G-Stage	37.4	67.9
Aux-G-Layer	37.2	68.3
Aux-NAS	**39.8**	**70.7**

ViT-Base Normal (Prim) + Seg (Aux) Tasks

NYU v2, Primary: Normal	Err (↓) Mean	Med.	RMSE	Within t° (%) (↑) 11.25	22.5
Single	14.6	12.9	17.7	43.2	80.8
Aux-Head	14.8	13.2	17.9	41.9	80.1
Adashare	13.2	11.4	16.8	49.7	82.2
Adashare-Aux	12.9	11.0	16.7	51.9	85.5
Aux-G-Layer	12.6	10.7	15.7	52.3	**85.9**
Aux-NAS	**12.5**	**10.3**	**15.6**	**53.8**	**85.9**

Ablation Analysis

Gradient	Feature	NAS	Seg. (%) (↑) mIoU	PAcc
✓			35.4	65.9
✓		✓	35.7	66.0
✓	✓	✓	**36.0**	**66.1**

Our Code is Released!

https://github.com/ethanygao/Aux-NAS