Toward Edge-Efficient Dense Predictions with Synergistic Multi-Task Neural Architecture Search

Supplementary Material

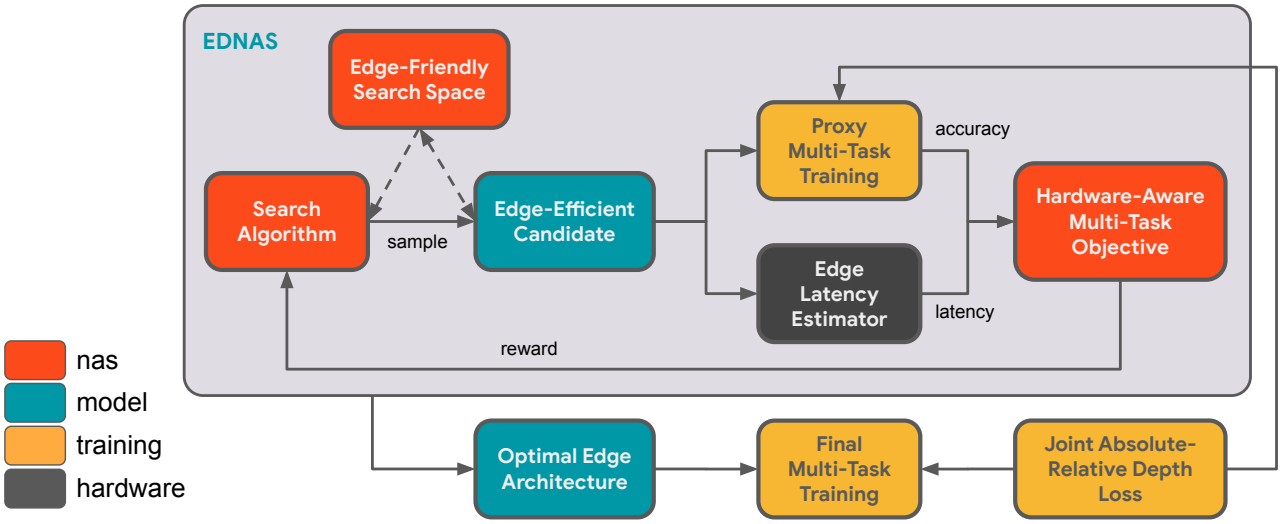


Figure 3: A system-level overview of our proposed methods. We leverage multi-objective, hardware-aware neural architecture search to discover optimal neural components suitable for multi-task dense predictions, while simultaneously ensuring efficient edge inference.

A. Experimental details

Hyperparameters of NAS. We use a Regularized Evolution controller with a population size of 50, random initialization, uniform mutator, and a tournament sample size of 10. We let the search run for about 2000 generations. These parameters were simply chosen to fit our computational budget and were not tuned. During the search, we train models for 5000 iterations as a proxy task to save computation. The final models are trained for 20000 iterations following AdaShare. For the β in the objective function in Eq. 5, we use (p=0.0) to set up a hard constraint function and (q=-0.07) to promote Pareto optimality, following MnasNet. We use $w_{i,j}$=1.0 to equally weight all evaluation metrics $M_{i,j}$ of any task T_i in Eq. 6 and Eq. 7. These can be adjusted to suit downstream applications. With 512 TPUv2 cores, our multi-trial search takes about 1.5 days for Cityscapes and 3.5 days for NYUv2. Since EDNAS is not constrained by the specific NAS algorithm, one can also use a one-shot search with weight sharing [6, 63] instead for better computational efficiency. Finally, Fig. 4 provides a visual comparison of IBN and Fused-IBN blocks.

Reference	Model	Seg $\mathcal{L}_{CE}$	Depth $\mathcal{L}_1$	Depth $\mathcal{L}_{RE}$	SN $\mathcal{L}_{ICS}$
Tab. 2: Cityscapes	MT edge baseline	0.4	1.000	—	—
	EDNAS	0.4	1.000	—	—
	EDNAS+JAReD	0.5	0.950	0.050	—
Tab. 3: NYUv2	MT edge baseline	1.0	1.000	—	40.0
	EDNAS	1.0	1.000	—	50.0
	EDNAS+JAReD	1.0	0.999	0.001	60.0

Table 8: Final loss weights. This table specifies the per-task loss weights for models trained on 2-task Cityscapes and 3-task NYUv2. "SN" stands for surface normal estimation.

Task Loss and Weighting. Following AdaShare [53], we use Cross-Entropy loss $\mathcal{L}_{CE}$ to train semantic segmentation, $\mathcal{L}_1$ loss for the base training of monocular depth estimation, and the inverse of cosine similarity loss $\mathcal{L}_{ICS}$ for surface normal prediction. Our JAReD loss also includes a weighted mean relative error component $\mathcal{L}_{RE}$ as specified in Eq. 8 We manually tune the loss weights to avoid ineffective weighting interfering with the evaluation of NAS-found architectures, using two guidelines: (1) We set task weights so that our *MT edge baseline* best matches

Method	Loss weight			Seg		Depth		Avg		
	$\mathcal{L}_{CE}$	$\mathcal{L}_1$	$\mathcal{L}_{RE}$	mIoU	PAcc	AbsE	RelE	ΔT_S	ΔT_D	ΔT
Single-task seg	1.000	0.000	0.000	40.04	88.68	—	—	—	—	—
Single-task depth	0.000	1.000	0.000	—	—	0.0157	0.340	—	—	—
Multi-task seg-depth	0.500	1.000	0.000	38.64	88.49	0.0171	0.354	-1.9	-6.3	-4.1
Multi-task seg-depth	0.500	0.999	0.001	46.78	90.62	0.0149	0.323	+9.5	+5.1	+7.3
Multi-task seg-depth	0.500	0.990	0.010	46.83	90.56	0.0144	0.304	+9.5	+9.5	+9.5
Multi-task seg-depth	0.500	0.950	0.050	46.11	90.47	0.0143	0.281	+8.6	+13.3	+10.9
Multi-task seg-depth	0.500	0.900	0.010	46.41	90.56	0.0146	0.300	+9.0	+9.5	+9.3

Table 9: Impact of loss weighting

AdaShare's (Sec. 4.1), then use similar weights for *EDNAS*. (2) For *EDNAS+JAReD*, we keep the λ in Eq. 8 small to avoid overwhelming the $\mathcal{L}_1$ and other tasks such as segmentation. Tab. 8 details the final weights of our main models, as presented in Tab. 2 and Tab. 3. In addition, Tab. 9 illustrates the impact of different loss weighting strategies on the multi-task performance of segmentation and depth prediction.

Δ Metrics for MTL Evaluation. Following the standard metrics for evaluating multi-task learning [40, 53, 59], we calculate the scores of multi-task learning relative to the single-task performance. Specifically, given a *multi-task* model a for evaluation, let $T_i \in T$ be a task of interest (e.g. semantic segmentation) and $m_{ij} \in M_i$ be an evaluation metric for task T_i (e.g. mIoU). Let $\hat{m}_{ij}$ be the *baseline* score of a corresponding *singe-task* model (e.g. single-task segmentation mIoU). We define the per-metric relative score Δm_{ij} (e.g. ΔmIoU) of the multi-task model a with regard to its baseline $\hat{m}_{ij}$ as followed:

$$\Delta m_{ij} = (-1)^{l_j} \frac{m_{ij} - \hat{m}_{ij}}{\hat{m}_{ij}} * 100\% \tag{9}$$

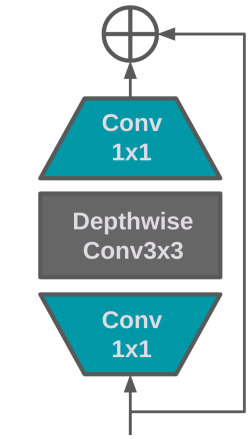
$$\text{with } l_j = \begin{cases} 1 \text{ if lower is better for metric } M_j \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

We then define the per-task relative score ΔT_i (e.g. ΔSeg) of any task T_i and the overall multi-task score ΔT of model a respectively as:

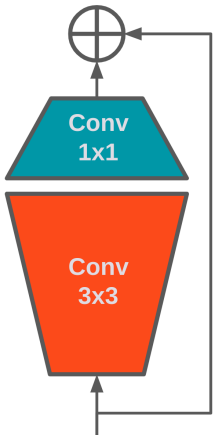
$$\Delta T_i = \frac{1}{|M_i|} \sum_{j=1}^{|M_i|} \Delta m_{ij} \tag{11}$$

$$\Delta T = \frac{1}{|T|} \sum_{i=1}^{|T|} \Delta T_i \tag{12}$$

with $|M_i|$ and $|T|$ being the cardinality of the corresponding metric set and task set respectively.



(a) Inverted Bottleneck (IBN)



(b) Fused-IBN

Figure 4: A visual comparison of the Inverted Bottleneck (IBN) [48] and Fused-IBN [64, 67, 56] blocks.

B. Qualitative Results

Figure 5 presents some qualitative results of semantic segmentation and depth estimation on CityScapes dataset

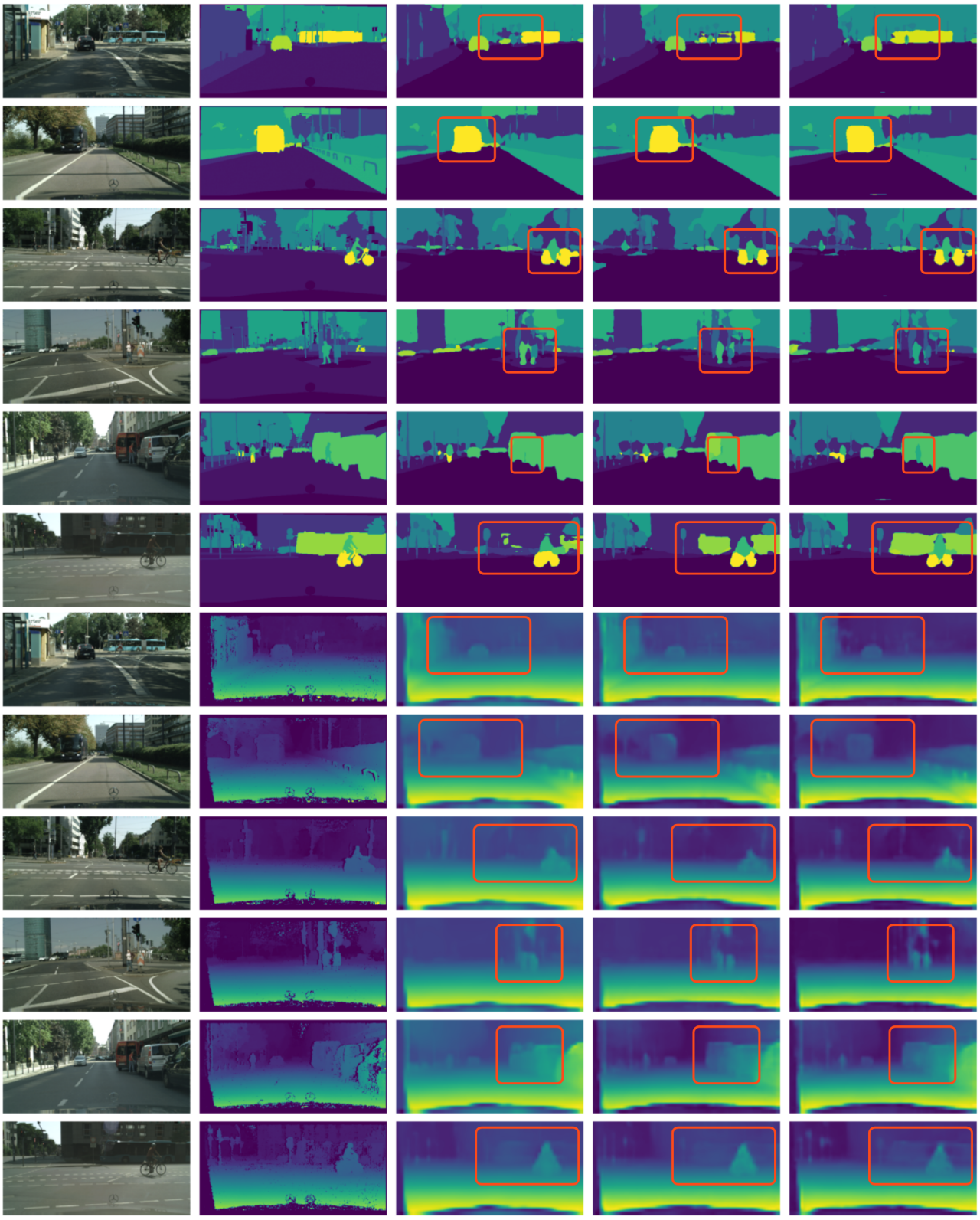


Figure 5: Qualitative results for semantic segmentation (top) and depth estimation (bottom) on CityScapes dataset