Appendix for "Exploring Tradeoffs through Mode Connectivity for Multi-Task Learning"

Anonymous Author(s)
Affiliation
Address
email

Contents

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

A Implementation Details

A.1 Evaluation Protocol

For scene understanding benchmarks such as CityScapes and NYUv2, mainstream multi-task learning (MTL) approaches typically report the final performance by averaging results over the last 10 epochs, due to the absence of a validation set. However, this evaluation protocol is not directly applicable to our method, which generates a solution curve during training and evaluates sampled points along the curve to demonstrate superior trade-offs. Specifically, we save the trained model at the final epoch and evaluate MTL performance at $t = \frac{1}{2}$, which represents the midpoint of the curve.

For image classification benchmarks such as CelebA, we adopt the same training procedure as used for the scene understanding benchmarks. However, we utilize a validation set to select the optimal value of t and report the corresponding MTL performance.

A.2 Experimental Setting

In line with the implementation and training strategy of FairGrad [Ban and Ji, 2024], we construct our model using SegNet [Badrinarayanan et al., 2017], with MTAN [Liu et al., 2019] employed as the backbone. The model is trained using the Adam optimizer for 200 epochs. The initial learning rate is set to 1e-4 and decayed by a factor of 2 after 100 epochs. The batch size is set to 2 for NYUv2 and 8 for CityScapes. The control point numbers are 4 and 5 on CityScapes and NYUv2, respectively, with 2 and 3 trainable in the second stage.

For the CelebA dataset, we adopt a 9-layer convolutional neural network (CNN) as the backbone, with task-specific linear heads appended. The model is trained with the Adam optimizer for 15 epochs using an initial learning rate of 3e-4 and a batch size of 256. The control point number is 3, with 1 trainable in the second stage.

Regarding the MultiMNIST dataset, we follow the protocol described in PaMaL [Dimitriadis et al., 2023]. Each MultiMNIST image is formed by sampling (with replacement) two MNIST digits (28×28), which are placed at the top-left and bottom-right of a 36×36 grid. This composite image is then resized to 28×28 pixels. The resulting dataset comprises 60,000 training, 10,000 validation, and 10,000 test samples. The model uses a LeNet-style shared-bottom architecture: the encoder contains two convolutional layers with 10 and 20 channels (kernel size 5), each followed by max pooling and ReLU activation. The encoder outputs a 50-dimensional embedding. Each decoder consists of two fully connected layers, with the final output layer producing predictions over 10 classes. The model is trained using Adam with a learning rate of 0.001, no learning rate scheduler, a batch size of 256, and a total of 10 training epochs. The control point number is 3, with 1 trainable in the second stage.

B Additional Experiments

B.1 Ablation Study

Our system comprises multiple components, including the order-aware objective ($\mathcal{R}_o$), alignment objective ($\mathcal{L}_{\text{align}}$ in Eqn.7), and curve selection. To evaluate the effectiveness and rationale behind each component, we conduct an ablation study on the NYUv2 dataset, with the results presented in Table 1. As shown, without $\mathcal{L}_{\text{ast}}$, `EXTRA` still outperforms the baseline but falls short of the complete system's performance. This suggests that, while the model can minimize loss, it fails to properly calibrate the remaining tasks at $t = \frac{1}{2}$. Additionally, `EXTRA` shows slight improvements without $\mathcal{L}_o$, demonstrating the strong capability of NURBS in capturing diverse trade-offs. This observation is further corroborated by the comparison between NURBS and Bézier, which reveals a substantial performance gap in MTL.

B.2 Grouping Strategy Comparison

To further demonstrate the effectiveness of the proposed grouping strategy, we compare it with two alternative approaches: random grouping and K-means clustering. The corresponding results are shown in Figure 1. In this experiment, the 40 tasks are grouped into 3 clusters using each of the three strategies, followed by multi-task learning (MTL) training. As illustrated, our strategy achieves

Table 1: Ablation study of EXTRA on *NYUv2* (3 tasks).

$\mathcal{L}_o$	$\mathcal{L}_{align}$	Bézier	NURBS	Δm% ↓
				-4.66
✓			✓	-6.05
	✓		✓	-4.97
✓	✓	✓		-4.36
✓	✓		✓	-6.30

the best overall MTL performance, outperforming both random and K-means grouping. Notably, K-means clustering fails to deliver satisfactory results due to imbalanced cluster sizes. Specifically, we observe that two of the three K-means clusters contain only a single task, which hinders the exploration of trade-offs under our training framework.

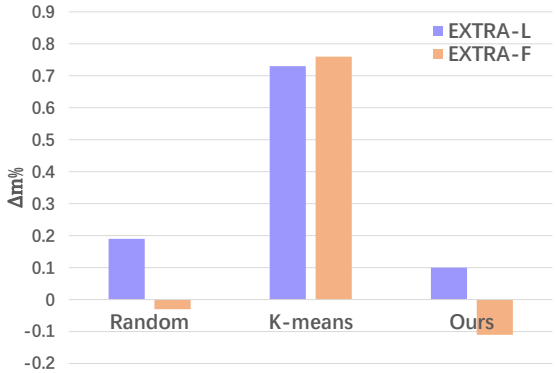


Figure 1: Comparison of grouping strategies on CelebA. 'Random' represents the uniform division for 40 tasks, while K-means represents leveraging K-means to cluster the warmup gradients of 40 tasks.

B.3 Memory Cost Analysis

To evaluate the efficiency and scalability of our method compared to endpoint-based approaches, we statistically analyze their memory consumption during training across various MTL benchmarks, as shown in Figure 2. As depicted, in the 2-task (CityScapes) and 3-task (NYUv2) settings, EXTRA incurs slightly higher memory usage due to its two-stage training paradigm. However, in the large-scale scenario with 40 tasks (CelebA), endpoint-based methods such as PaMaL require substantial memory resources, resulting in an out-of-memory (OOM) issue during training. In contrast, EXTRA maintains a manageable memory footprint, demonstrating better scalability.

B.4 Plug-and-Play Verification

In addition to LS and FairGrad, we further incorporate another mainstream MTL approach, CAGrad, to demonstrate the plug-and-play capability of EXTRA, with the corresponding results shown in Table 2. As illustrated, EXTRA also provides significant improvements when applied to CAGrad, following the same trend observed with LS and FairGrad. Specifically, EXTRA not only enhances overall MTL performance but also improves each individual metric, thereby verifying its plug-and-play effectiveness.

B.5 Analysis on Control Point Number

We further analyze the effect of the number of bends in the NURBS representation on the CityScapes dataset, with results presented in Figure 4. As shown, EXTRA-L achieves the best performance in terms of Δm% when the number of bends is set to 4. Notably, reducing the number of bends to 3 leads to a substantial drop in performance, likely due to the limited expressive capacity of the

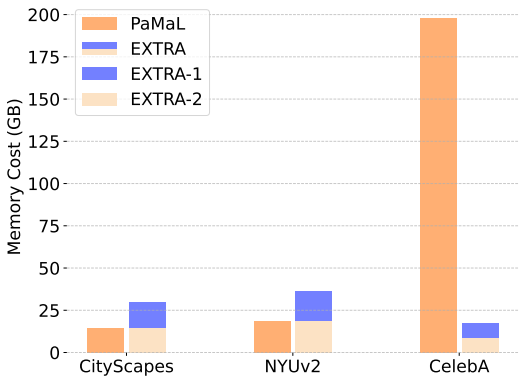
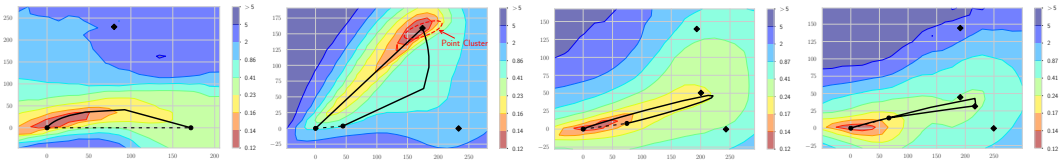


Figure 2: Memory Cost Comparison. 'EXTRA-1' and 'EXTRA-2' represents the first and second training stage of EXTRA. Note that the memory consumption of PaMaL on the CelebA dataset is estimated, as its actual training raises an out-of-memory (OOM) issue under our experimental settings.

Table 2: **Scene understanding** (*CityScapes*, 2 tasks). ▲/▼ indicates outperforms/underperforms their vanilla versions. 'EXTRA-L', 'EXTRA-C', and 'EXTRA-F' are EXTRA augmented LS, CAGrad, and FairGrad versions.

Method	Segmentation ↑		Depth ↓		$\Delta\mathbf{m}\%$ ↓
	mIoU	Pix. Acc.	Abs. Err.	Rel. Err.	
Independent	74.01	93.16	0.0125	27.77	-
LS	75.18	93.49	0.0155	46.77	22.60
EXTRA-L	75.53 ▲	93.63 ▲	0.0127 ▲	33.45 ▲	4.93 ▲
CAGrad	75.16	93.48	0.0141	37.60	11.58
EXTRA-C	75.50 ▲	93.55 ▲	0.0135 ▲	35.73 ▲	8.61 ▲
FairGrad	75.72	93.68	0.0134	32.25	5.18
EXTRA-F	76.11 ▲	93.58 ▼	0.0126 ▲	30.20 ▲	1.63 ▲

NURBS curve. Conversely, increasing the number of bends to 5 or 6 does not yield performance gains, which is somewhat counterintuitive. To investigate this further, we visualize the corresponding loss landscapes in Figure 3. As illustrated, EXTRA-3 degenerates into a Bézier-like curve, exhibiting excessive smoothness. Meanwhile, the curves produced by EXTRA-5 and EXTRA-6 appear irregular and less stable, likely due to the increased difficulty in optimization, which in turns highlights the superiority of choosing curve-based rather than endpoint-based mode connectivity for MTL.



(a) 3 (b) 4 (c) 5 (d) 6

Figure 3: The loss landscape of employing NURBS with different control point number (including endpoints). We abbreviate them as EXTRA-3, EXTRA-4, EXTRA-5, and EXTRA-6.

B.6 Hyper-parameter Analysis

We evaluate the impact of the hyperparameter α on the final performance and present the results in Figure 5. As shown, the performance remains relatively stable across different values of α, with the highest average performance achieved at $\alpha = 0.3$. However, $\alpha = 0.5$ offers a better trade-off between average performance and variance. Therefore, we adopt $\alpha = 0.5$ as the default setting in our experiments.

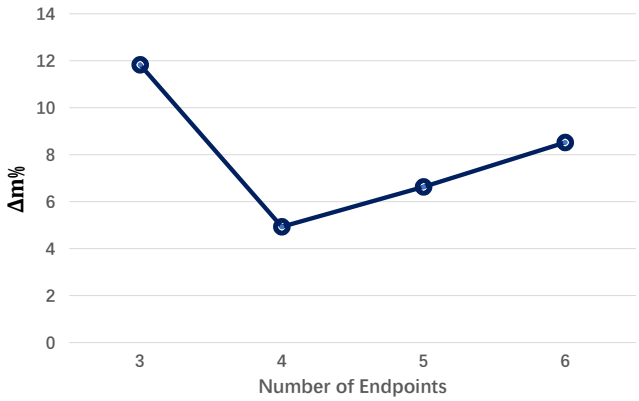


Figure 4: Analysis of number of bends on *CityScapes*, 2 tasks.

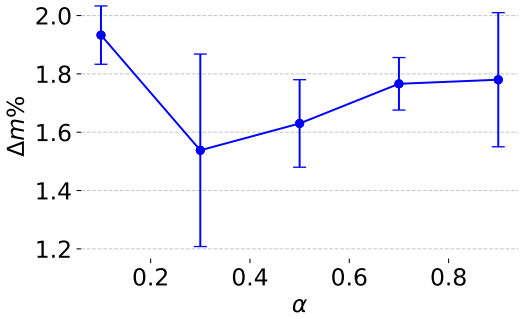


Figure 5: Analysis on α.

C Limitation and Discussion

Although EXTRA achieves state-of-the-art performance on mainstream MTL benchmarks, it faces challenges in delivering user-preference MTL results when the number of tasks exceeds two. While a Bézier surface can align mode connectivity with the Pareto front for three tasks, this approach becomes progressively more difficult as the number of tasks increases, which represents a key limitation of our method. Addressing this limitation will be a focal point of future work. Additionally, one might raise concerns regarding the fairness of our evaluation, given that EXTRA utilizes multiple points (endpoints and control points) to construct the curve, thereby increasing the model's capacity. While we acknowledge this concern, we emphasize that our work introduces a novel perspective to MTL, potentially offering an alternative to gradient-based MTL approaches, which have been debated for their effectiveness [Xin et al., 2022].

References

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Hao Ban and Kaiyi Ji. Fair resource allocation in multi-task learning. *arXiv preprint arXiv:2402.15638*, 2024.

Nikolaos Dimitriadis, Pascal Frossard, and François Fleuret. Pareto manifold learning: Tackling multiple tasks via ensembles of single-task models. In *International Conference on Machine Learning*, pages 8015–8052. PMLR, 2023.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. Do current multi-task optimization methods in deep learning even help? *Advances in neural information processing systems*, 35:13597–13609, 2022.