Supplementary Material for Multi-Task Dense Prediction via Mixture of Low-Rank Experts

Yuqi Yang[1,2*] Peng-Tao Jiang[2*] Qibin Hou[1,3†] Hao Zhang[2] Jinwei Chen[2] Bo Li[2]
[1]VCIP, CS, Nankai University [2]vivo Mobile Communication Co., Ltd [3]NKIARI, Shenzhen Futian
yangyq2000@mail.nankai.edu.cn, pt.jiang@vivo.com, andrewhoux@gmail.com

1. More Implementation Details

1.1. MLoRE Module Stacking

In our method, we stack two MLoRE modules after each selected backbone layer. In the first MLoRE module, the lightweight task-specific 1×1 convolutions are utilized to project the backbone feature to different task features. In the second MLoRE module, as the task-specific features have been split already, we utilize 1×1 convolutions directly to deal with task-specific features. Moreover, as the MLoRE is a linear module, we add a task-specific non-linear block between the two MLoRE modules to introduce the non-linearity into our decoder. Each non-linear block is composed of a GELU-BatchNorm-Linear structure.

1.2. MoE Optimization

Following previous MoE-based MTL methods [1,3], we utilize the noising gating and load-balancing loss proposed by Shazeer *et al*. [5], which is a common practice in sparse MoE training [4,5].

One may concern that without the load-balancing loss, it will have a higher possibility for an expert to be activated by all the tasks on the same sample. However, we can't discard the load-balancing loss to construct the global relationships across all the tasks in one expert for two reasons. Firstly, discarding the load-balancing loss will weaken MoE's ability to dynamically choose different experts for different samples, which is opposite to our motivation for using MoE. Secondly, without the load-balancing loss, most experts will be less or never activated, which will hurt the capacity of MoE. On the contrary, the proposed task-sharing generic path will not harm the ability of dynamic routing and the capacity of MoE. We will prove the necessity of load-balancing loss in the following experiments.

In addition, our MLoRE is trained with the top-k constraint. When training MoE without top-k constraint, we found each expert would be shared by all the tasks. However, we empirically found that this might make the optimization process difficult and harm MoE's ability to build relationships in a subset of tasks. As a result, although it can build global task relationships, the performance is highly influenced without the top-k constraint, as described in the left figure of Fig. 5 in the main paper. It can be seen that the performance of the setting without top-k constraint is lower than the top-k settings. On the contrary, our proposed task-sharing generic path can explicitly build global task relationships while MoE still builds relationships in a subset of tasks.

1.3. Re-parameterization During Training

Since the re-parameterization can speed up the forward propagation, it is natural to ask if we can extend the re-parameterization to the training phase for better training efficiency. However, in our MLoRE module, the re-parameterization can only be performed at the inference stage. The re-parameterization in training will largely influence the training-time behaviour and the reason is the BatchNorm layer in our MLoRE module. We follow RepVGG [2] to set BatchNorm in our task-sharing low-rank expert path, which is important for the re-parameterization-based method as stated in Sec. 4.2 of RepVGG. When a BatchNorm layer merges with a convolution layer in training, the feature statistics for this BatchNorm layer will be hard to perform.

2. Additional Study on MLoRE Module

2.1. Number of MLoRE Module

The number of the MLoRE module at one scale would also influence the performance. We conduct a series of experiments on it, and the results are shown in Tab. 1. It can be seen that when increasing the number of the MLoRE modules from 1 to 2, the MTL gain is increased from -1.25 to -0.58. When further stacking 1 MLoRE module, we do not observe obvious performance gain (-0.74 *v.s.* -0.58). Thus, in our paper, we set the number of the MLoRE module at

*The first two authors contributed equally to this paper. Work was done when Yuqi Yang was an intern at vivo.
†Qibin Hou is the corresponding author.

Table 1. Ablation on the number of MLoRE module on PASCAL-Context dataset.

Number	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Δ_m ↑
1	78.80	66.83	85.15	13.57	73.40	-1.25
2	79.26	67.82	85.30	13.65	74.69	-0.58
3	79.05	67.94	85.07	13.72	74.75	-0.74

Table 2. Ablation on the rank setting of the low-rank task-sharing generic path on PASCAL-Context dataset.

Rank	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Δ_m ↑
16	78.53	66.71	85.27	13.68	73.70	-1.41
128	78.78	67.01	85.08	13.70	73.98	-1.26
vanilla (3x3)	79.26	67.82	85.30	13.65	74.69	-0.58

Table 3. Ablation on the number of experts on PASCAL-Context dataset.

Expert Number	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Δ_m ↑
0	78.30	67.43	85.19	14.03	74.52	-1.58
5	78.73	67.47	85.16	13.76	74.36	-1.11
10	78.95	67.75	85.24	13.63	74.51	-0.72
15	79.26	67.82	85.30	13.65	74.69	-0.58
20	79.19	67.92	85.13	13.57	74.55	-0.53

each scale to 2.

2.2. Low-rank Task-Sharing Generic Path

We further ablate the effectiveness of the low-rank task-sharing generic path. We utilize the low-rank format of a vanilla 3×3 convolution in the task-sharing generic path to explore whether we can design a lighter module with a fully low-rank structure in MLoRE. The results are shown in Tab. 2. With the increase of the rank, the performance improves on most of the tasks. When using the vanilla 3x3 convolution, its performance outperforms the low-rank settings by a large margin. This result shows that it is beneficial to use the vanilla 3x3 convolution rather than its low-rank format to construct the task-sharing generic path.

2.3. Detailed Results for Number of Task-Sharing Low-Rank Experts and Top-k Selection

We present the detailed performance of the number of task-sharing low-rank experts and top-k selection for every task in Tab. 3 and Tab. 4. It can be seen that increasing the number of experts can achieve better performance on most of the tasks, which is also verified in previous work [1]. This also proves the necessity of introducing the linear and low-rank structure into the MoE.

Table 4. Ablation of the top-k with 15 experts on PASCAL-Context dataset.

Top-k	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Δ_m ↑
k=3	79.22	67.84	85.18	13.70	74.25	-0.72
k=6	79.19	67.81	85.25	13.66	74.64	-0.65
k=9	79.26	67.82	85.30	13.65	74.69	-0.58
k=12	79.05	67.75	85.23	13.58	74.43	-0.64
k=15	78.91	67.51	85.18	13.54	74.22	-0.75

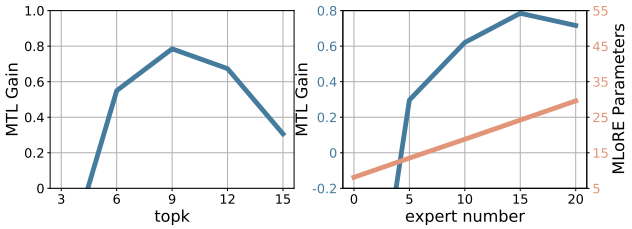

Figure 1. Ablation study on the number of experts N and the number of activated experts K on dataset NYUD-v2. In the right figure, we also present the parameter change of the MLoRE module with the increase in the number of experts.

Table 5. Ablation of the effectiveness of load-balancing loss in different settings on PASCAL-Context dataset. **MoE**: baseline with the standard MoE structure. **LoRE**: baseline with the task-sharing low-rank expert path. **Ours**: our MLoRE is equipped with all the components. **w/o LB loss**: without the load-balancing loss.

Settings	Semseg mIoU ↑	Parsing mIoU ↑	Saliency maxF ↑	Normal mErr ↓	Boundary odsF ↑	MTL Δ_m ↑
MoE	78.56	66.78	85.17	13.58	73.91	-1.20
w/o LB loss	78.32	66.41	85.14	13.58	73.81	-1.40
LoRE	78.38	66.21	85.15	13.71	73.53	-1.71
w/o LB loss	78.04	66.05	85.10	13.69	73.60	-1.80
Ours	79.26	67.82	85.30	13.65	74.69	-0.58
w/o LB loss	79.01	68.03	85.24	13.66	74.38	-0.70

2.4. Effectiveness of Load-Balancing Loss

We conduct extensive experiments to verify the effectiveness of load-balancing loss. To prove its necessity for the MoE structure, we test three settings with different designs. The results are shown in Tab. 5. It can be clearly seen that with the load-balancing loss, it will achieve better performance on most tasks in all three settings. The MTL gain is also improved with the load-balancing loss. The quantitative results demonstrate the effectiveness of the load-balancing loss for the MoE structure and motivates us to propose the task-sharing generic path rather than discard the load-balancing loss to build global relationships.

2.5. Ablation on NYUD-v2

We also conduct some important ablations on the NYUD-v2 dataset. Specifically, we conduct the ablation on the num-

Table 6. Performance comparison between our method and the baseline based on ViT-L on the NYUD-v2 and PASCAL-Context datasets.

	Pascal-Context					NYUDv2			
Method	**Semseg**	**Parsing**	**Sal.**	**Nor.**	**Bound.**	**Semseg**	**Depth**	**Nor.**	**Bound.**
Baseline	78.59	67.78	84.43	13.87	70.26	54.55	0.6043	18.62	75.21
Ours	81.41	70.52	84.90	13.51	75.42	55.96	0.5076	18.33	78.43

Table 7. Ablation of the top-k with 15 experts on PASCAL-Context dataset.

Method (HRNet18)	**Semseg** mIoU ↑	**Parsing** mIoU ↑	**Saliency** maxF ↑	**Normal** mErr ↓	**Boundary** odsF ↑	**MTL** Δ_m ↑
MTI-Net	61.70	60.18	84.78	14.23	70.80	-2.12
ATRC	62.69	59.42	84.70	14.20	70.96	-1.98
Ours	62.43	60.78	84.85	14.05	71.84	-1.13

ber of task-sharing low-rank experts and top-k selection on NYUD-v2. The results are shown in Fig. 1. In addition, we also show the relation visualization on NYUD-v2 in Fig. 2 It can be seen that the results of these ablations still support the conclusions in our main paper.

2.6. More Quantitative Results

We show the performance of the baseline models based on ViT-L in Tab. 6. Furthermore, we conduct experiments to evaluate the performance based on HRNet18 [6]. The results are shown in Tab. 7. Our method outperforms other methods in terms of Δ_m by a large margin.

3. More Visual Results

3.1. More Visual Comparison Results

We present more qualitative results compared with the former SOTA methods, TaskPrompter [8] and InvPT [7]. In Fig. 3 and Fig. 4, we can see that our method generates better visual results than previous SOTA methods on most of the tasks.

3.2. Relation Visualizations from More Layers

We present the relation visualization between tasks and the low-rank experts from every scale in Fig. 5. These visualizations clearly show that experts with different ranks tend to learn different subsets of tasks for all the MLoRE modules. We also show the ratio for different experts to be activated by different numbers of tasks when the task-sharing generic path is not added for all the MLoRE modules in Fig. 6. Most of the experts in MLoRE modules are seldom activated by all the tasks which ties in with our motivation. A few experts are shared by all the tasks more frequently, though, we find that the minimum gating value among the five tasks for these experts is relatively low, which is under half of the average value (0.11) for most of the time. This can also prove that it is hard for one expert to build global relationships and effectively aid the final results without the task-sharing generic path.

Figure 2. (a) The relations between tasks and low-rank experts on dataset NYUD-v2. (b) The ratio of an expert activated by different numbers of tasks in the MLoRE module without the task-sharing generic path on dataset NYUD-v2. We can see that without the task-sharing generic path, there are only a few experts can be activated by all four tasks. Horizontal coordinates represent the ranks of different experts.

References

[1] Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11828–11837, 2023. 1, 2

[2] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 1

[3] Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In *Adv. Neural Inform. Process. Syst.*, volume 35, pages 28441–28457, 2022. 1

[4] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 1

[5] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 1

[6] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 3

[7] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *Eur. Conf. Comput. Vis.*, pages 514–530. Springer, 2022. 3, 5, 6

[8] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *Int. Conf. Learn. Represent.*, 2022. 3, 5, 6

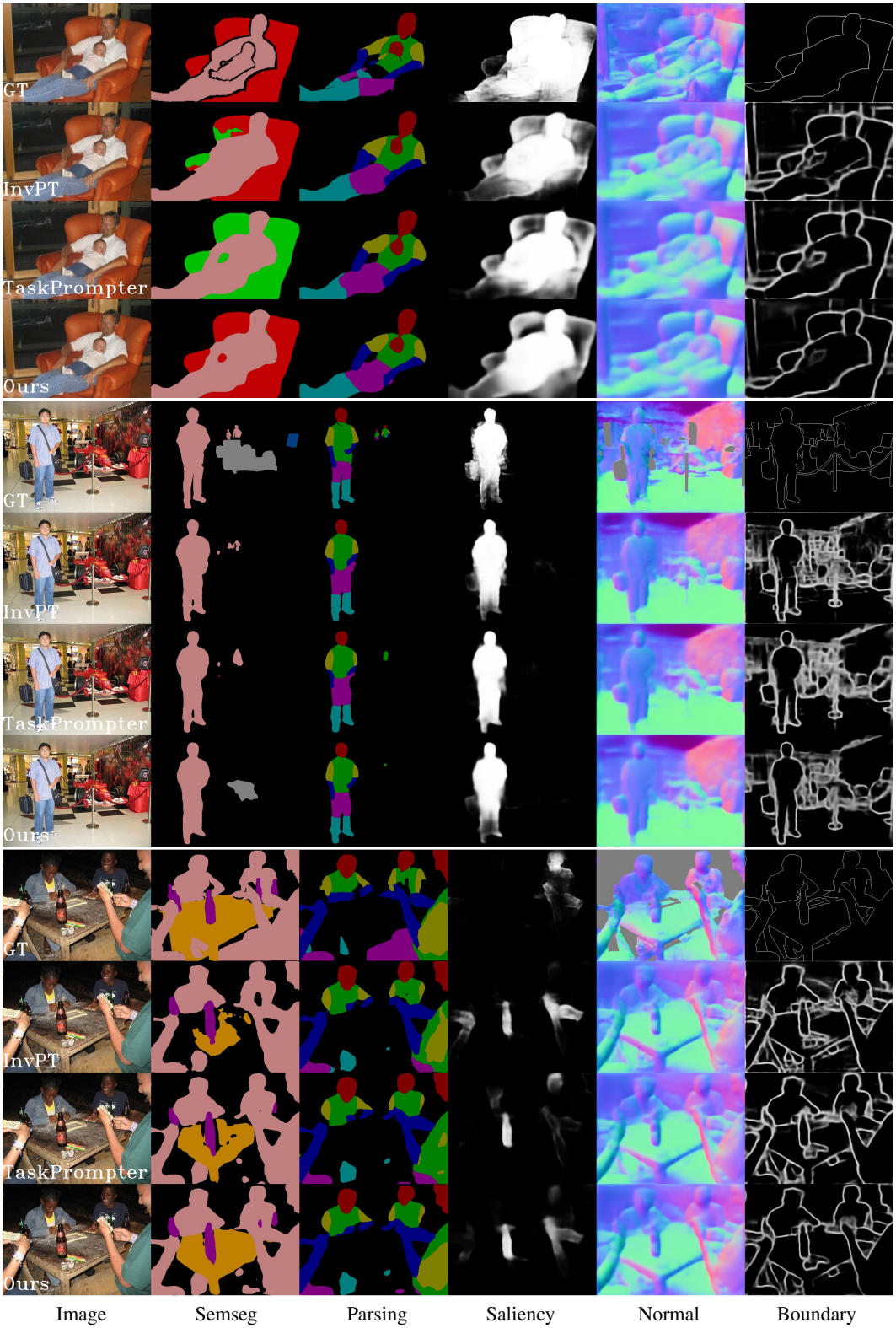

Figure 3. More Qualitative comparison on PASCAL-Context dataset among different methods, including InvPT [7], TaskPrompter [8], and ours. Best viewed with zoom-in. It can be seen that our method achieves better visual results than other methods on all five tasks thanks to the proposed MLoRE module.

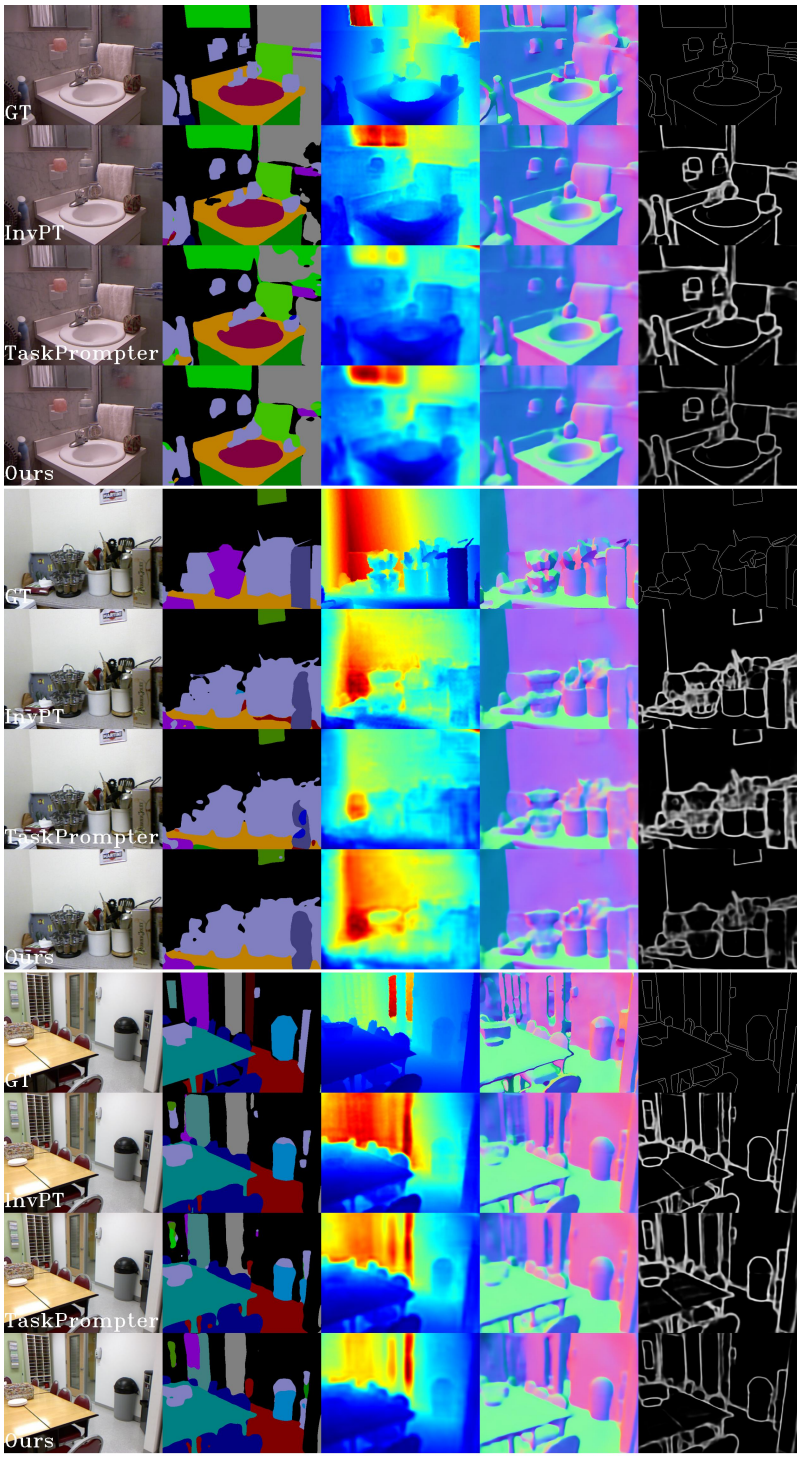

Figure 4. More Qualitative comparison on NYUD-v2 dataset among different methods, including InvPT [7], TaskPrompter [8], and ours. Best viewed with zoom-in. It can be seen that our method achieves better visual results than other methods on all five tasks thanks to the proposed MLoRE module.

Figure 5. The relations between tasks and low-rank experts in all the MLoRE modules in our model.

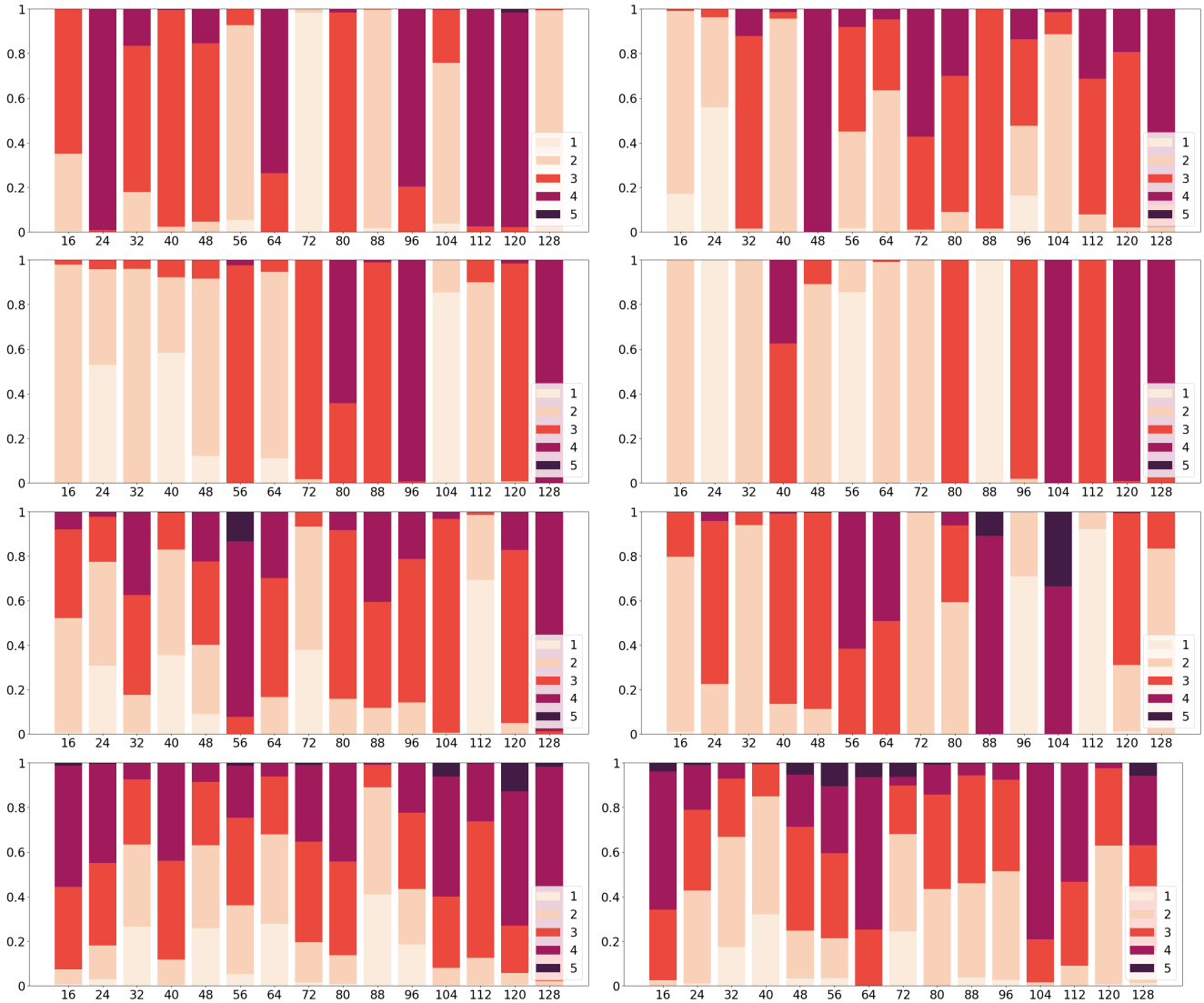

Figure 6. The ratio of an expert activated by different numbers of tasks in all the MLoRE modules in our model when the task-sharing generic path is not equipped. Horizontal coordinates represent the ranks of different experts.