7. Appendix

7.1. Non-jointly Labeled Datasets

The non-jointly labeled dataset is widely used in multi-task learning problems. The training datasets consist of T subset of data which includes N_t data points for a certain task t, $\mathcal{D}_t = \{(x_t, y_t)_i | i \in [N_t]\}$. The jointly labeled dataset can be viewed as a special case of the above definition.

The objective function defined on the non-jointly labeled dataset is similar to (6) as

$$\mathcal{L}(\theta) = \sum_{t \in [T]} \sum_{x_t, y_t \in \mathcal{D}} L_t(g_{\theta_t} \circ f_{\theta_s}(x_t), y_t). \tag{6}$$

From a variational inference perspective, the only difference is that each data point has an extra task-specific coefficient which is proportional to the number of samples for that task. We can rewrite (2) as

$$\begin{aligned}
&\log \prod_i p(y_i|x_i) = \sum_i \log \sum_t p(y_{t,i}|x_{i,t})p(t) \\
&\geq \sum_i \sum_t \frac{N_t}{N} \log p(y_{t,i}|t, x_i) \\
&= \sum_t \frac{N_t}{N} \sum_i \log \int_z p(y_{t,i}|z, t, x_i)p(z|t)dz \\
&\geq \sum_t \frac{N_t}{N} \sum_i \int_z q(z|t) \log \frac{p(y_{t,i}|z, t, x_i)p(z|t)}{q(z|t)} dz \\
&= \sum_t \frac{N_t}{N} \sum_i \mathbb{E}_{z \sim q(\cdot|t)}[\log p(y_{t,i}|z, t, x_i)] \\
&\quad - \mathbf{KL}(q(\cdot|t), p(\cdot|t)),
\end{aligned}$$

where $N = \sum_t N_t$.

7.2. Target Distribution Updating

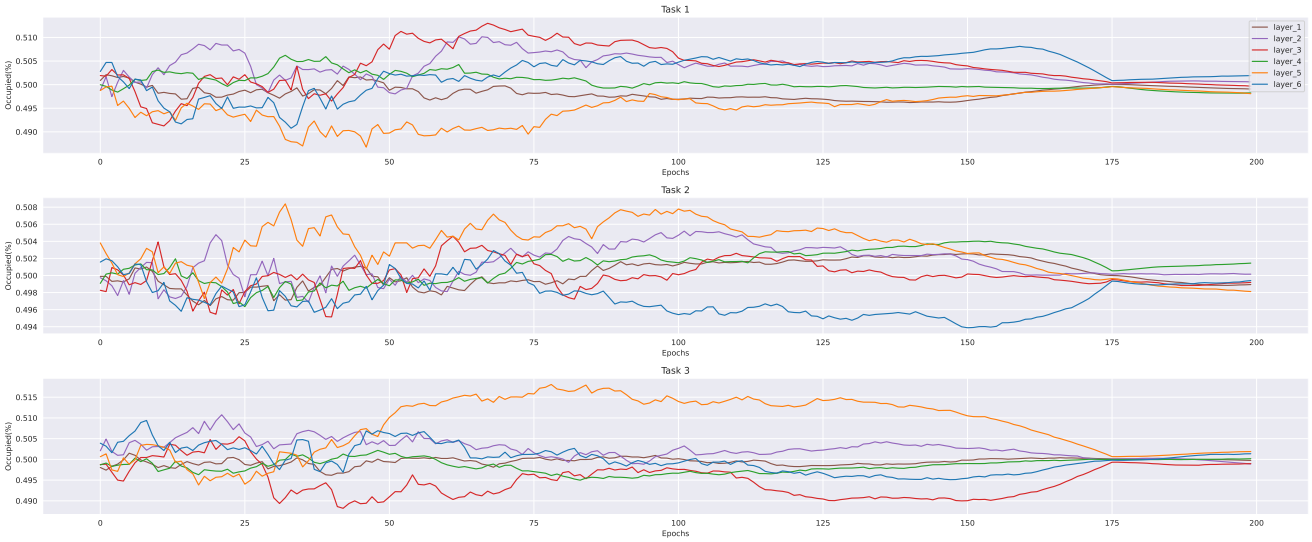

Figure 7. The proportion of tasks SemSeg, Depth, and Normal during the training process.

As mentioned in Section 4.3.4, Figure 7 illustrates the similarities among different tasks by showing the proportion of $\Phi_2/(\Phi_2 + \Phi_3)$, $\Phi_1/(\Phi_1 + \Phi_3)$ and $\Phi_1/(\Phi_1 + \Phi_2)$ respectively. This also reveals the potential structural discrepancy between the learned architecture from the training process and the fully shared backbone.

7.3. Ablation Study on the Multi-CIFAR100

In addition to the main body of the paper, the ablation studies on the Multi-CIFAR100 dataset are shown here as strong complementary evidence. The results shown in the tables below are averaged over 4 experiments with different random seeds.

7.3.1 The Parameterized Distributions

We first test MPPS with different parameterized distributions in Table 9. Note that MPPS with Gaussian distribution shows a worse performance than the baseline. From our observation of the training loss experimental logs and the mask distribution logs, the mask distribution is early converged to a fixed high EC distribution even though the target distribution has a low EC value. The reason we guess is that there may be some numerical optimization problems in calculating KL divergence between Gaussian distributions. The optimization process may easily enter into local optimum when the learning rate is not carefully fine-tuned.

Layers	**Average Acc** ↑	**Lowest Acc** ↑	**Median Acc** ↑	Δ↑
multi	59.23%	37.22%	60.52%	0.00%
MPPS +B	67.62%	41.38%	71.06%	14.36%
MPPS +G	43.66%	27.07%	42.77%	-26.32%

Table 9. Applying MPPS with different parameterized distributions on the Multi-CIFAR100 dataset

7.3.2 Applying on Different Numbers of Network Layers

Table 10 shows the learning performance with a different number of layers on the Multi-CIFAR100 dataset. We recommend our audience to search this hyperparameter for different datasets. Finding the best parameterized scheme is important for MPPS and we guess it depends both on the architecture of the shared backbone and downstream tasks. At this point, we leave this problem as future work.

Layers	**Average Acc** ↑	**Lowest Acc** ↑	**Median Acc** ↑	Δ↑
6	67.62%	41.38%	71.06%	14.36%
12	59.14%	32.43%	61.17%	-0.32%
All	48.86%	29.50%	50.09%	-17.51%

Table 10. Applying MPPS to different numbers of selected layers on the Multi-CIFAR100 dataset

7.3.3 Exclusive Capacity Scheduler

Table 11 shows the comparison results on the Multi-CIFAR100 dataset. Every scheduler we test performs better than the baseline. Separ scheduler has the lowest performance compared with others for the same reason mentioned in Section 4.3.3 at most time of the training process the network is encouraged to learn a separated pattern but use a full share scheme when testing.

Method	**Average Acc** ↑	**Lowest Acc** ↑	**Median Acc** ↑	Δ↑
multi	59.23%	37.22%	60.52%	0.00%
Linear	67.62%	41.38%	71.06%	14.36%
Separ	66.95%	39.43%	69.40%	13.12%
FullShare	67.08%	40.93%	70.21%	13.42%
Step5	67.38%	40.80%	70.23%	13.94%
Step10	67.52%	40.80%	70.87%	14.16%
Step20	67.52%	39.78%	70.72%	14.07%
Quad	67.70%	40.78%	70.45%	14.47%

Table 11. MPPS with different schedulers on the Multi-CIFAR100

7.3.4 Target Distribution Updates

In Table 12, we show the benefits of the target distribution updating mechanism where following the configuration mentioned in Section 4.3.4.

Method	Average Acc ↑	Lowest Acc ↑	Median Acc ↑	Δ↑
multi	59.23%	37.22%	60.52%	0.00%
w/ updating	67.62%	41.38%	71.06%	14.36%
w/o updating	66.51%	46.82%	63.28%	12.16%

Table 12. MPPS with and without target distribution updating

7.4. Limitations

In this section, we analyze the limitations of MPPS.

1. Extra training time and memory cost as shown in Section 4.4.

2. There is a theoretical gap between the final learned stochastic network in training and the fully shared backbone in testing. This may potentially make our approach less effective in some datasets or task combinations.

3. MPPS can hardly be used to fine-tune the backbone models pre-trained on other datasets because MPPS is trained from separated network structures, while a pre-trained model has an entirely shared backbone.

Under the Gaussian distribution parameterized situation, we can give a simple analysis of the magnitude of the theoretical gap described above. Assuming that we have θ', Φ' as the minimization of (2) in the last course where the target distribution is given as $\mathcal{N}(\mathbf{1}, (\epsilon\mathbf{1})^2)$ which approximates a fully shared target distribution when ϵ is a very small value, i.e.

$$\theta', \Phi' = \arg\min_{\theta,\Phi} \sum_t \mathbb{E}_{m\sim p_\Phi(t)}[L(\theta \odot m)] + \mathbf{KL}(p_\Phi, \mathcal{N}(\mathbf{1}, (\epsilon\mathbf{1})^2)).$$

The expected error with the stochastic network on the test dataset is given by

$$\sum_t \mathbb{E}_{m\sim p'_\Phi(t)}[L(\theta' \odot m)], \tag{7}$$

and when we use a fully shared backbone instead of stochastic networks the error is given by

$$\sum_t L(\theta'). \tag{8}$$

The gap(|(7)-(8)|) is

$$
\begin{aligned}
&|\sum_t \mathbb{E}_{m\sim p'_\Phi(t)}[L(\theta'\odot m)] - \sum_t L(\theta')| \\
&\le \sum_t |\mathbb{E}_{m\sim p'_\Phi(t)}[L(\theta'\odot m)] - \mathbb{E}_{m\sim\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)}[L(\theta'\odot m)] + \mathbb{E}_{m\sim\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)}[L(\theta'\odot m)] - L(\theta')| \\
&\le \sum_t |\mathbb{E}_{m\sim p'_\Phi(t)}[L(\theta'\odot m)] - \mathbb{E}_{m\sim\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)}[L(\theta'\odot m)]| + \sum_t |\mathbb{E}_{m\sim\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)}[L(\theta'\odot m)] - L(\theta')| \\
&\le \sum_t |\mathbb{E}_{m\sim p'_\Phi(t)}[L(\theta'\odot m)] - \mathbb{E}_{m\sim\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)}[L(\theta'\odot m)]| + C \qquad (9) \\
&\le \sum_t |\mathbb{E}_{\epsilon_a\sim\mathcal{N}(0,1)}[L(\theta'+\theta'\epsilon_a\sigma_{t,\Phi})] - \mathbb{E}_{\epsilon_b\sim\mathcal{N}(0,1)}[L(\theta'+\theta'\epsilon_b\epsilon)]| + C \qquad (10) \\
&= \sum_t |\mathbb{E}_{\epsilon_a\sim\mathcal{N}(0,1)}[L(\theta'+\theta'\epsilon_a\sigma_{t,\Phi}) - L(\theta') + L(\theta') - L(\theta'+\theta'\epsilon_a\epsilon)]| + C \\
&= \sum_t |\mathbb{E}_{\epsilon_a\sim\mathcal{N}(0,1)}[\nabla L(\theta')\epsilon_a\sigma_{t,\Phi}\theta' + O(||\epsilon\sigma_{t,\Phi}\theta'||_2^2) - (\nabla L(\theta')\epsilon_a\epsilon\theta' + O(||\epsilon_a\epsilon\theta'||_2^2))]| + C \\
&\le \sum_t |\mathbb{E}_{\epsilon_a\sim\mathcal{N}(0,1)}[\nabla L(\theta')\epsilon_a\sigma_{t,\Phi}\theta' + O(||\epsilon\sigma_{t,\Phi}\theta'||_2^2)| + \sum_t |\mathbb{E}_{\epsilon_a\sim\mathcal{N}(0,1)}[\nabla L(\theta')\epsilon_a\epsilon\theta' + O(||\epsilon_a\epsilon\theta'||_2^2)] + C \\
&= \sum_t O(||\sigma_{t,\Phi}\theta'||_2^2) + \sum_t O(||\epsilon\theta'||_2^2) + C \\
&\le ||\theta'||_2^2(\sum_t ||\sigma_{\Phi,t}||_2^2 + C') + C, \qquad (11)
\end{aligned}
$$

where (9) is due to the assumption that the $\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)$ is a good approximation of the fully shared target distribution and causes a bounded error gap C and (10) is due to reparameterization trick, where $m\sim p_\Phi(t)\to m=1+\sigma_{t,\Phi}\epsilon_a, \epsilon_a\sim\mathcal{N}(0,1)$ and $\sigma_{t,\Phi}=\sqrt{\frac{1-\Phi_t}{\Phi_t}}$.

Recall the KL divergence between two Gaussian distributions is $\mathbf{KL}(p,q)=\log\frac{\sigma_2}{\sigma_1}+\frac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2}-\frac{1}{2}$, where $p=\mathcal{N}(\mu_1,\sigma_1), q=\mathcal{N}(\mu_2,\sigma_2)$, so $\arg\min_\Phi \mathbf{KL}(p_\Phi,\mathcal{N}(\mathbf{1},(\epsilon\mathbf{1})^2)) = \arg\min_\Phi \sum_t\sum_i \log\frac{\epsilon}{\sigma_{t,\Phi,i}}+\frac{\sigma_{t,\Phi,i}^2}{2\epsilon^2}-\frac{1}{2}$. Let $\delta_{t,i}=\sigma_{t,\Phi,i}-\epsilon>0$, we can rewrite it as $\sum_t\sum_i\frac{1}{2}(\frac{\delta_{t,i}}{\epsilon})^2+\frac{\delta_{t,i}}{\epsilon}-\log(1+\frac{\delta_{t,i}}{\epsilon})\approx\sum_t\sum_i(\frac{\delta_{t,i}}{\epsilon})^2\le\sum_t\sum_i\frac{1}{\epsilon^2}\sigma_{t,\Phi,i}^2\approx\sum_t||\sigma_{\Phi,t}||_2^2$ when $\frac{\delta_t}{\epsilon}\to 0$.

So the (11) is proportional to two terms, the $\mathbf{KL}$ term and the model weight term. We propose to add a control coefficient $\beta>1$ in (2) before the KL term as

$$
\sum_{x,\mathbf{y}\in\mathcal{D}}\sum_{t\in[T]}\mathbb{E}_{m\sim\tilde{q}_\Phi(\cdot,t)}[L_t(g_{\theta_t}^t\circ f_{\theta_s\odot m}(x),y_t)+\beta(\log\tilde{q}_\Phi(m,t)-\log\tilde{p}_\Pi(m,t))]
$$

to reduce the gap and use a large weight decay coefficient to reduce the weight norm for future works.