Beyond Losses Reweighting: Empowering Multi-Task Learning via the Generalization Perspective

Supplementary Material

Due to space constraints, some details were omitted from the main paper. We therefore include additional theoretical developments (section A) and experimental results (section C) in this appendix.

A. Our Theory Development

This section contains the proofs and derivations of our theory development to support the main submission.

We first start with the following theorem, which is inspired by the general PAC-Bayes in [2].

Theorem 2. *With the assumption that adding Gaussian perturbation will raise the test error:* $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\left[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})\right]$. *Let T be the number of parameter $\boldsymbol{\theta}$, and N be the cardinality of $\mathcal{S}$, then the following inequality is true with the probability $1-\delta$:*

$$\mathcal{L}_{\mathcal{D}}\left(\boldsymbol{\theta}\right) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})\right] + \frac{1}{\sqrt{N}}\Bigg[\frac{1}{2}+ \frac{T}{2}\log\Big(1+\frac{||\boldsymbol{\theta}||^2}{T\sigma^2}\Big) + \log\frac{1}{\delta} + 6\log(N+T) + \frac{L^2}{8}\Bigg],$$

where L is the upper-bound of the loss function.

Proof. We use the PAC-Bayes theory for $P=\mathcal{N}(\mathbf{0},\sigma_P^2\mathbb{I}_T)$ and $Q = \mathcal{N}(\boldsymbol{\theta},\sigma^2\mathbb{I}_T)$ are the prior and posterior distributions, respectively.

By using the bound in [2], with probability at least $1-\delta$ and for all $\beta>0$, we have:

$$\mathbb{E}_{\boldsymbol{\theta}\sim Q}\left[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})\right] \leq \mathbb{E}_{\boldsymbol{\theta}\sim Q}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})\right] + \frac{1}{\beta}\Big[\mathsf{KL}(Q\|P) + \log\frac{1}{\delta} + \Psi(\beta,N)\Big],$$

where we have defined:

$$\Psi(\beta,N) = \log\mathbb{E}_P\mathbb{E}_{\mathcal{S}}\Big[\exp\Big\{\beta\big(\mathcal{L}_{\boldsymbol{D}}(\boldsymbol{\theta})-\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})\big)\Big\}\Big].$$

Note that the loss function is bounded by L, according to Hoeffding's lemma, we have:

$$\Psi(\beta,N) \leq \frac{\beta^2L^2}{8N}.$$

By Cauchy inequality:

$$\frac{1}{\sqrt{N}}\Bigg[\frac{T}{2}\log\Big(1+\frac{||\boldsymbol{\theta}||^2}{T\sigma^2}\Big)+\frac{L^2}{8}\Bigg] \geq \frac{L}{2\sqrt{N}}\sqrt{T\log\Big(1+\frac{||\boldsymbol{\theta}||^2}{T\sigma^2}\Big)} \geq L,$$

which means that the theorem is proved since the loss function is upper bounded by L, following assumptions, if $||\boldsymbol{\theta}||^2 \geq T\sigma^2\Big[\exp\frac{4N}{T}-1\Big]$.

Now, we only need to prove the theorem under the case: $||\boldsymbol{\theta}||^2 \leq T\sigma^2\Big[\exp\frac{4N}{T}-1\Big]$.

We need to specify P in advance since it is a prior distribution. However, we do not know in advance the value of $\boldsymbol{\theta}$ that affects the KL divergence term. Hence, we build a family of distribution P as follows:

$$\mathfrak{P} = \Big\{P_j = \mathcal{N}(\mathbf{0},\sigma_{P_j}^2\mathbb{I}_T) : \sigma_{P_j}^2 = c\exp\big(\frac{1-j}{T}\big), c = \sigma^2\big(1+\exp\frac{4N}{T}\big), j=1,2,\dots\Big\}.$$

Set $\delta_j = \frac{6\delta}{\pi^2j^2}$, the below inequality holds with probability at least $1-\delta_j$:

$$\mathbb{E}_{\boldsymbol{\theta}\sim Q}\left[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})\right] \leq \mathbb{E}_{\boldsymbol{\theta}\sim Q}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})\right] + \frac{1}{\beta}\Big[\mathsf{KL}(Q\|P_j) + \log\frac{1}{\delta_j} + \frac{\beta^2L^2}{8N}\Big].$$

Or it can be written as:

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\left[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})\right] \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}+\boldsymbol{\epsilon})\right] + \frac{1}{\beta}\Big[\mathsf{KL}(Q\|P_j) + \log\frac{1}{\delta_j} + \frac{\beta^2L^2}{8N}\Big].$$

Thus, with probability $1-\delta$ the above inequalities hold for all P_j. We choose:

$$j^* = \left\lfloor 1 + T\log\left(\frac{\sigma^2\big(1+\exp\{4N/T\}\big)}{\sigma^2+\|\boldsymbol{\theta}\|^2/T}\right)\right\rfloor.$$

Since $\frac{\|\boldsymbol{\theta}\|^2}{T} \leq \sigma^2\big[\exp\frac{4N}{T}-1\big]$, we get $\sigma^2+\frac{\|\boldsymbol{\theta}\|^2}{T} \leq \sigma^2\exp\frac{4N}{T}$, thus j^* is well-defined. We also have:

$$\begin{aligned}
& T\log\frac{c}{\sigma^2+\|\boldsymbol{\theta}\|^2/T} \leq j^* \leq 1+T\log\frac{c}{\sigma^2+\|\boldsymbol{\theta}\|^2/T} \\
\Rightarrow\quad & \log\frac{c}{\sigma^2+\|\boldsymbol{\theta}\|^2/T} \leq \frac{j^*}{T} \leq \frac{1}{T}+\log\frac{c}{\sigma^2+\|\boldsymbol{\theta}\|^2/T} \\
\Rightarrow\quad & -\frac{1}{T}+\log\frac{\sigma^2+\|\boldsymbol{\theta}\|^2/T}{c} \leq \frac{-j^*}{T} \leq \log\frac{\sigma^2+\|\boldsymbol{\theta}\|^2/T}{c} \\
\Rightarrow\quad & e^{-1/T}\frac{\sigma^2+\|\boldsymbol{\theta}\|^2/T}{c} \leq e^{-j^*/T} \leq \frac{\sigma^2+\|\boldsymbol{\theta}\|^2/T}{c} \\
\Rightarrow\quad & \sigma^2+\frac{\|\boldsymbol{\theta}\|^2}{T} \leq ce^{\frac{1-j^*}{T}} \leq e^{\frac{1}{T}}\Big(\sigma^2+\frac{\|\boldsymbol{\theta}\|^2}{T}\Big) \\
\Rightarrow\quad & \sigma^2+\frac{\|\boldsymbol{\theta}\|^2}{T} \leq \sigma_{P_{j^*}}^2 \leq e^{\frac{1}{T}}\Big(\sigma^2+\frac{\|\boldsymbol{\theta}\|^2}{T}\Big).
\end{aligned}$$

Hence, we have:

$$\mathsf{KL}(Q\|P_{j^*}) = \frac{1}{2}\Big[\frac{T\sigma^2+\|\boldsymbol{\theta}\|^2}{\sigma^2_{P_{j^*}}} - T + T\log\frac{\sigma^2_{Pj^*}}{\sigma^2}\Big]$$
$$\leq \frac{1}{2}\Big[\frac{T\sigma^2+\|\boldsymbol{\theta}\|^2}{\sigma^2+\|\boldsymbol{\theta}\|^2/T} - T + T\log\frac{e^{1/T}\big(\sigma^2+\|\boldsymbol{\theta}\|^2/T\big)}{\sigma^2}\Big]$$
$$\leq \frac{1}{2}\Big[1 + T\log\big(1+\frac{\|\boldsymbol{\theta}\|^2}{T\sigma^2}\big)\Big].$$

For the term $\log\frac{1}{\delta_{j^*}}$, use the inequality $\log(1+e^t) \leq 1+t$ for $t>0$:

$$\log\frac{1}{\delta_{j^*}} = \log\frac{(j^*)^2\pi^2}{6\delta} = \log\frac{1}{\delta} + \log\Big(\frac{\pi^2}{6}\Big) + 2\log(j^*)$$
$$\leq \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log\Big(1 + T\log\frac{\sigma^2\big(1+\exp(4N/T)\big)}{\sigma^2+\|\boldsymbol{\theta}\|^2/T}\Big)$$
$$\leq \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log\Big(1 + T\log\big(1+\exp(4N/T)\big)\Big)$$
$$\leq \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + 2\log\Big(1 + T\big(1+\frac{4N}{T}\big)\Big)$$
$$\leq \log\frac{1}{\delta} + \log\frac{\pi^2}{6} + \log(1+T+4N).$$

Choosing $\beta = \sqrt{N}$, with probability at least $1-\delta$ we get:

$$\frac{1}{\beta}\Big[\mathsf{KL}(Q\|P_{j^*}) + \log\frac{1}{\delta_{j^*}} + \frac{\beta^2L^2}{8N}\Big]$$
$$\leq \frac{1}{\sqrt{N}}\Big[\frac{1}{2} + \frac{T}{2}\log\Big(1+\frac{\|\boldsymbol{\theta}\|^2}{T\sigma^2}\Big) + \log\frac{1}{\delta} + 6\log(N+T)\Big]$$
$$+ \frac{L^2}{8\sqrt{N}}.$$

Thus the theorem is proved. □

Back to our context of multi-task learning in which we have m tasks with each task model: $\boldsymbol{\theta}^i = [\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}^i_{ns}]$, we can prove the following theorem.

Theorem 3. *With the assumption that adding Gaussian perturbation will rise the test error: $\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^i) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\big[\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}^i+\boldsymbol{\epsilon})\big]$. Let T_i be the number of parameter $\boldsymbol{\theta}^i$ and N be the cardinality of $\mathcal{S}$. We have the following inequality holds with probability $1-\delta$ (over the choice of training set $\mathcal{S}\sim\mathcal{D}$):*

$$\big[\mathcal{L}^i_{\mathcal{D}}\left(\boldsymbol{\theta}^i\right)\big]^m_{i=1} \leq \big[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2\mathbb{I})}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^i+\boldsymbol{\epsilon})\right] + f^i\left(\|\boldsymbol{\theta}^i\|^2_2\right)\big]^m_{i=1}, \tag{17}$$

where

$$f^i\left(\|\boldsymbol{\theta}^i\|^2_2\right) = \frac{1}{\sqrt{N}}\Bigg[\frac{1}{2} + \frac{T_i}{2}\log\Big(1+\frac{||\boldsymbol{\theta}||^2}{T_i\sigma^2}\Big) + \log\frac{1}{\delta} + 6\log(N+T_i) + \frac{L^2}{8}\Bigg].$$

Proof. The result for the base case $m=1$ can be achieved by using Theorem 2 where $\xi=\delta$ and f^1 is defined accordingly. We proceed by induction, suppose that Theorem 3 is true for all $i\in[n]$ with probability $1-\delta/2$, which also means:

$$\big[\mathcal{L}^i_{\mathcal{D}}\left(\boldsymbol{\theta}^i\right)\big]^n_{i=1} \leq \big[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma\mathbb{I})}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^i+\boldsymbol{\epsilon})\right] + f^i\left(\|\boldsymbol{\theta}^i\|^2_2\right)\big]^n_{i=1}.$$

Using Theorem 2 for $\boldsymbol{\theta}^{n+1}$ and $\xi=\delta/2$, with probability $1-\delta/2$, we have:

$$\mathcal{L}^{n+1}_{\mathcal{D}}\left(\boldsymbol{\theta}^{n+1}\right) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma\mathbb{I})}\left[\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}^{n+1}+\boldsymbol{\epsilon})\right] + f^{n+1}\left(\|\boldsymbol{\theta}^{n+1}\|^2_2\right).$$

Using the inclusion–exclusion principle, with probability at least $1-\delta$, we reach the conclusion for $m=n+1$.

We next prove the result in the main paper. Let us begin by formally restating the main theorem as follows:

Theorem 4. *For any perturbation radius $\rho_{sh}, \rho_{ns} > 0$, with probability $1-\delta$ (over the choice of training set $\mathcal{S}\sim\mathcal{D}$) we obtain:*

$$\big[\mathcal{L}^i_{\mathcal{D}}\left(\boldsymbol{\theta}^i\right)\big]^m_{i=1} \leq \tag{18}$$
$$\max_{\|\boldsymbol{\epsilon}_{sh}\|_2\leq\rho_{sh}}\Bigg[\max_{\|\boldsymbol{\epsilon}^i_{ns}\|_2\leq\rho_{ns}}\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^i_{ns}\right) \tag{19}$$
$$+ f^i\left(\|\boldsymbol{\theta}^i\|^2_2\right)\Bigg]^m_{i=1}, \tag{20}$$

where $f^i\left(\|\boldsymbol{\theta}^i\|^2_2\right)$ is defined the same as in Theorem 3.

Proof. Theorem 3 gives us

$$\big[\mathcal{L}^i_{\mathcal{D}}\left(\boldsymbol{\theta}^i\right)\big]^m_{i=1}$$
$$\leq \big[\mathbb{E}_{\boldsymbol{\epsilon}\sim N(0,\sigma^2\mathbb{I})}\left[\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}^i+\boldsymbol{\epsilon}\right)\right] + f^i\left(\|\boldsymbol{\theta}^i\|_2\right)\big]^m_{i=1}$$
$$= \Bigg[\int \mathbb{E}_{\boldsymbol{\epsilon}^i_{ns}}\left[\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^i_{ns}\right)\right] p\left(\boldsymbol{\epsilon}_{sh}\right) d\boldsymbol{\epsilon}_{sh} + f^i\left(\|\boldsymbol{\theta}^i\|_2\right)\Bigg]^m_{i=1}$$
$$= \mathbb{E}_{\boldsymbol{\epsilon}_{sh}}\left[\mathbb{E}_{\boldsymbol{\epsilon}^i_{ns}}\left[\mathcal{L}^i_{\mathcal{S}}\left(\boldsymbol{\theta}_{sh}+\boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}^i_{ns}+\boldsymbol{\epsilon}^i_{ns}\right)\right] + f^i\left(\|\boldsymbol{\theta}^i\|_2\right)\right]^m_{i=1},$$

where $p(\boldsymbol{\epsilon}_{sh})$ is the density function of Gaussian distribution; $\boldsymbol{\epsilon}_{sh}$ and $\boldsymbol{\epsilon}^i_{ns}$ are drawn from their corresponding Gaussian distributions.

We have $\boldsymbol{\epsilon}^i_{ns} \sim N(0, \sigma^2\mathbb{I}_{ns})$ with the dimension $T_{i,ns}$, therefore $\|\boldsymbol{\epsilon}^i_{ns}\|$ follows the Chi-square distribution. As proven in [37], we have for all i:

$$P\left(\|\boldsymbol{\epsilon}^i_{ns}\|^2_2 \geq T_{i,ns}\sigma^2 + 2\sigma^2\sqrt{T_{i,ns}t} + 2t\sigma^2\right) \leq e^{-t}, \forall t>0$$

$$P\left(\|\boldsymbol{\epsilon}^i_{ns}\|^2_2 < T_{i,ns}\sigma^2 + 2\sigma^2\sqrt{T_{i,ns}t} + 2t\sigma^2\right) > 1-e^{-t}$$

for all $t > 0$.

Select $t = \ln(\sqrt{N})$, we derive the following bound for the noise magnitude in terms of the perturbation radius ρ_{ns} for all i:

$$P\left(\|\boldsymbol{\epsilon}_{ns}^i\|_2^2 \leq \sigma^2(2\ln(\sqrt{N}) + T_{i,ns} + 2\sqrt{T_{i,ns}\ln(\sqrt{N})})\right) > 1 - \frac{1}{\sqrt{N}}. \tag{21}$$

Moreover, we have $\boldsymbol{\epsilon}_{sh} \sim N(0, \sigma^2\mathbb{I}_{sh})$ with the dimension T_{sh}, therefore $\|\boldsymbol{\epsilon}_{sh}\|$ follows the Chi-square distribution. As proven in [37], we have:

$$P\left(\|\boldsymbol{\epsilon}_{sh}\|_2^2 \geq T_{sh}\sigma^2 + 2\sigma^2\sqrt{T_{sh}t} + 2t\sigma^2\right) \leq e^{-t}, \forall t > 0$$
$$P\left(\|\boldsymbol{\epsilon}_{sh}\|_2^2 < T_{sh}\sigma^2 + 2\sigma^2\sqrt{T_{sh}t} + 2t\sigma^2\right) > 1 - e^{-t}$$

for all $t > 0$.

Select $t = \ln(\sqrt{N})$, we derive the following bound for the noise magnitude in terms of the perturbation radius ρ_{sh}:

$$P\left(\|\boldsymbol{\epsilon}_{sh}\|_2^2 \leq \sigma^2(2\ln(\sqrt{N}) + T_{sh} + 2\sqrt{T_{sh}\ln(\sqrt{N})})\right) > 1 - \frac{1}{\sqrt{N}}. \tag{22}$$

By choosing σ less than $\frac{\rho_{sh}}{\sqrt{2\ln N^{1/2} + T_{sh} + 2\sqrt{T_{sh}\ln N^{1/2}}}}$ and $\min_i \frac{\rho_{ns}}{\sqrt{2\ln N^{1/2} + T_{i,ns} + 2\sqrt{T_{i,ns}\ln N^{1/2}}}}$, and referring to (21,22), we achieve both:

$$P\left(\|\boldsymbol{\epsilon}_{ns}^i\| < \rho_{ns}\right) > 1 - \frac{1}{N^{1/2}}, \forall i,$$
$$P\left(\|\boldsymbol{\epsilon}_{sh}\| < \rho_{sh}\right) > 1 - \frac{1}{N^{1/2}}.$$

Finally, we finish the proof as:

$$\begin{aligned}
&\left[\mathcal{L}_{\mathcal{D}}^i\left(\boldsymbol{\theta}^i\right)\right]_{i=1}^m \\
&\leq \mathbb{E}_{\boldsymbol{\epsilon}_{sh}}\left[\mathbb{E}_{\boldsymbol{\epsilon}_{ns}^i}\left[\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)\right] + f^i\left(\|\boldsymbol{\theta}^i\|_2\right)\right]_{i=1}^m \\
&\leq \max_{||\boldsymbol{\epsilon}_{sh}||<\rho_{sh}}\left[\max_{||\boldsymbol{\epsilon}_{ns}^i||<\rho_{ns}}\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)\right. \\
&\left. + \left(1 - \frac{1}{\sqrt{N}}\right)\frac{L}{\sqrt{N}} + \frac{1}{\sqrt{N}} + f^i\left(\|\boldsymbol{\theta}^i\|_2\right)\right]_{i=1}^m.
\end{aligned}$$

To reach the final conclusion, we redefine:

$$f^i\left(\|\boldsymbol{\theta}^i\|_2\right) = \left(1 - \frac{1}{\sqrt{N}}\right)\frac{L}{\sqrt{N}} + \frac{1}{\sqrt{N}} + f^i\left(\|\boldsymbol{\theta}^i\|_2\right).$$

Here we note that we reach the final inequality due to the following derivations:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\epsilon}_{sh}}\left[\mathbb{E}_{\boldsymbol{\epsilon}_{ns}^i}\left[\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)\right]\right]_{i=1}^m \\
&\leq \int_{B_{sh}}\left[\int_{B_{ns}^i}\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)d\boldsymbol{\epsilon}_{ns}^i + \frac{1}{\sqrt{N}}\right]_{i=1}^m d\boldsymbol{\epsilon}_{sh} \\
&+ \int_{B_{sh}^c}\left[\int_{B_{ns}^i}\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)d\boldsymbol{\epsilon}_{ns}^i + \frac{1}{\sqrt{N}}\right]_{i=1}^m d\boldsymbol{\epsilon}_{sh} \\
&\leq \int_{B_{sh}}\left[\int_{B_{ns}^i}\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)d\boldsymbol{\epsilon}_{ns}^i + \left(1 - \frac{1}{\sqrt{N}}\right)\frac{L}{\sqrt{N}} + \frac{1}{\sqrt{N}}\right]_{i=1}^m d\boldsymbol{\epsilon}_{sh} \\
&\leq \max_{||\boldsymbol{\epsilon}_{sh}||<\rho_{sh}}\left[\max_{||\boldsymbol{\epsilon}_{ns}^i||<\rho_{ns}}\left[\mathcal{L}_{\mathcal{S}}^i\left(\boldsymbol{\theta}_{sh} + \boldsymbol{\epsilon}_{sh}, \boldsymbol{\theta}_{ns}^i + \boldsymbol{\epsilon}_{ns}^i\right)\right.\right. \\
&\left.\left. + \left(1 - \frac{1}{\sqrt{N}}\right)\frac{L}{\sqrt{N}} + \frac{1}{\sqrt{N}}\right]\right]_{i=1}^m,
\end{aligned}$$

where $B_{sh} = \{\boldsymbol{\epsilon}_{sh} : ||\boldsymbol{\epsilon}_{sh}|| \leq \rho_{sh}\}$, B_{sh}^c is the complement set, and $B_{ns}^i = \left\{\boldsymbol{\epsilon}_{ns}^i : ||\boldsymbol{\epsilon}_{ns}^i|| \leq \rho_{ns}\right\}$.

We also provide a theoretical justification for our gradient decomposition.

Theorem 5. *If* $\left\langle \boldsymbol{g}_{sh}^{flat}, \boldsymbol{g}_{sh}^{i,loss}\right\rangle \leq 0, \forall i$ *and* $\left\langle \boldsymbol{g}_{sh}^{loss}, \boldsymbol{g}_{sh}^{i,flat}\right\rangle \leq 0, \forall i$*, the gradient decomposition strategy yields a smaller sum of the losses and the gradient norms. More formally, we have*

$$\begin{aligned}
&\left[l^i\left(\boldsymbol{\theta}_{sh} - \eta\boldsymbol{g}_{sh}^{SAM,dec}\right) + \rho_{sh}s^i\left(\boldsymbol{\theta}_{sh} - \eta\boldsymbol{g}_{sh}^{SAM,dec}\right)\right]_i \\
&\leq \left[l^i\left(\boldsymbol{\theta}_{sh} - \eta\boldsymbol{g}_{sh}^{SAM,dir}\right) + \rho_{sh}s^i\left(\boldsymbol{\theta}_{sh} - \eta\boldsymbol{g}_{sh}^{SAM,dir}\right)\right]_i.
\end{aligned}$$

This theorem shows that if the aggregation vector $\boldsymbol{g}_{sh}^{flat}$ *of the flatness component is non-congruent to* $\boldsymbol{g}_{sh}^{i,loss}$ *of the loss component (i.e., they form an obtuse angle) and the aggregation* $\boldsymbol{g}_{sh}^{loss}$ *of the loss component is also non-congruent to* $\boldsymbol{g}_{sh}^{i,flat}$ *of the flatness component, which possibly happens in the early stage of training, the gradient decomposition strategy optimizes the loss and the gradient norm better. The empirical evidence is given in Figure 2 when the gradient decomposition strategy gains lower loss values and gradient norms than the direct strategy.*

Proof. Given $\boldsymbol{\theta}_{ns}^i$, we denote $l^i(\boldsymbol{\theta}_{sh}) = \mathcal{L}_S(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}_{ns}^i)$ and $s^i(\boldsymbol{\theta}_{sh}) = \|\nabla_{\boldsymbol{\theta}_{sh}}\mathcal{L}_S(\boldsymbol{\theta}_{sh}, \boldsymbol{\theta}_{ns}^i)\|_2$. We have the $\boldsymbol{g}_{sh}^{i,\text{SAM}}$ minimizes $h^i(\boldsymbol{\theta}_{sh}) = l^i(\boldsymbol{\theta}_{sh}) + \rho_{sh}s^i(\boldsymbol{\theta}_{sh})$. Therefore,

their aggregation $\boldsymbol{g}_{sh}^{\text{SAM,dir}}$ minimizes $[h^i(\boldsymbol{\theta}_{sh})]_i$. We have

$$
\begin{aligned}
& h^i\left(\boldsymbol{\theta}_{sh}\right)-h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right) \\
\approx & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \nabla_{\boldsymbol{\theta}_{sh}} h^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \nabla_{\boldsymbol{\theta}_{sh}} l^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \nabla_{\boldsymbol{\theta}_{sh}} s^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle .
\end{aligned}
$$

$$
\begin{aligned}
& h^i\left(\boldsymbol{\theta}_{sh}\right)-h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right) \\
\approx & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dec}}, \nabla_{\boldsymbol{\theta}_{sh}} h^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dec}}, \nabla_{\boldsymbol{\theta}_{sh}} l^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dec}}, \nabla_{\boldsymbol{\theta}_{sh}} s^i\left(\boldsymbol{\theta}_{sh}\right)\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dec}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{SAM,dec}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{loss}}+\boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{loss}}+\boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle \\
= & \eta\left\langle \boldsymbol{g}_{sh}^{\text{loss}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle \\
& +\eta\left\langle \boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{loss}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle \\
\leq & \eta\left\langle \boldsymbol{g}_{sh}^{\text{loss}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle+\eta \rho_{sh}\left\langle \boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle .
\end{aligned}
$$

Due to the definition of $\boldsymbol{g}_{sh}^{\text{loss}}$ and $\boldsymbol{g}_{sh}^{\text{flat}}$, we have $\left\langle \boldsymbol{g}_{sh}^{\text{loss}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle \geq \left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \boldsymbol{g}_{sh}^{i,\text{loss}}\right\rangle$ and $\left\langle \boldsymbol{g}_{sh}^{\text{flat}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle \geq \left\langle \boldsymbol{g}_{sh}^{\text{SAM,dir}}, \boldsymbol{g}_{sh}^{i,\text{flat}}\right\rangle$. This follows that

$$
\begin{aligned}
& h^i\left(\boldsymbol{\theta}_{sh}\right)-h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right) \\
& \leq h^i\left(\boldsymbol{\theta}_{sh}\right)-h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right) \\
\Rightarrow & h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right) \leq h^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right) \\
\Rightarrow & l^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right)+\rho_{sh} s^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right) \\
& \leq l^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right)+\rho_{sh} s^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right), \forall i \\
\Rightarrow & \left[l^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right)+\rho_{sh} s^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dec}}\right)\right]_i \\
& \leq\left[l^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right)+\rho_{sh} s^i\left(\boldsymbol{\theta}_{sh}-\eta \boldsymbol{g}_{sh}^{\text{SAM,dir}}\right)\right]_i .
\end{aligned}
$$

□

B. Gradient aggregation strategies overview

This section details how the gradient_aggregate operation is defined according to recent gradient-based multi-task learning methods that we employed as baselines in the main paper, including MGDA [70], PCGrad [77], CAGrad [44] and IMTL [46]. Assume that we are given m vectors $\boldsymbol{g}^1, \boldsymbol{g}^2, \ldots, \boldsymbol{g}^m$ represent task gradients. Typically, we aim to find a combined gradient vector as:

$$
\boldsymbol{g}=\text{gradient\_aggregate}(\boldsymbol{g}^1, \boldsymbol{g}^2, \ldots, \boldsymbol{g}^m).
$$

B.1. Multiple-gradient descent algorithm - MGDA

[70] applies MGDA [16] to find the minimum-norm gradient vector that lies in the convex hull composed by task gradients $\boldsymbol{g}^1, \boldsymbol{g}^2, \ldots, \boldsymbol{g}^m$:

$$
\boldsymbol{g}=\text{argmin}||\sum_{i=1}^m w_i \boldsymbol{g}^i||^2, s.t. \sum_{i=1}^m w_i=1 \quad \text{and} \quad w_i \geq 0 \forall i.
$$

This approach can guarantee that the obtained solutions lie on the Pareto front of task objective functions.

B.2. Projecting conflicting gradients - PCGrad

PCgrad resolves the disagreement between tasks by projecting gradients that conflict with each other, i.e. $\langle \boldsymbol{g}^i, \boldsymbol{g}^j \rangle < 0$, to the orthogonal direction of each other. Specifically, $\boldsymbol{g}^i$ is replaced by its projection on the normal plane of $\boldsymbol{g}^j$:

$$
\boldsymbol{g}_{\text{PC}}^i=\boldsymbol{g}^i-\frac{\boldsymbol{g}^i \cdot \boldsymbol{g}^j}{||\boldsymbol{g}^j||^2} \boldsymbol{g}^j .
$$

Then compute the aggregated gradient based on these deconflict vectors $\mathbf{g}=\sum_i^m \mathbf{g}_{\text{PC}}^i$.

B.3. Conflict Averse Gradient Descent - CAGrad

CAGrad [44] seeks a worst-case direction in a local ball around the average gradient of all tasks, $\boldsymbol{g}_0$, that minimizes conflict with all of the gradients. The updated vector is obtained by optimizing the following problem:

$$
\max_{\mathbf{g} \in R} \min_{i \in[m]}\langle \mathbf{g}^i, \mathbf{g}\rangle \quad s.t. \quad ||\mathbf{g}-\mathbf{g}^0|| \leq c||\mathbf{g}^0||,
$$

where $\mathbf{g}^0=\frac{1}{m} \sum_i^m \mathbf{g}^i$ is the averaged gradient and c is a hyper-parameter.

B.4. Impartial multi-task learning - IMTL

IMTL [46] proposes to balance per-task gradients by finding the combined vector $\mathbf{g}$, whose projections onto $\{\mathbf{g}^i\}_{i=1}^m$ are equal. Following this, they obtain the closed-form solution for the simplex vector $\boldsymbol{w}$ for reweighting task gradients:

$$
\boldsymbol{w}=\boldsymbol{g}^1 \boldsymbol{U}^{\top}\left(\boldsymbol{D} \boldsymbol{U}^{\top}\right)^{-1}
$$

where $\boldsymbol{u}^i=\boldsymbol{g}^i / \left\|\boldsymbol{g}^i\right\|, \boldsymbol{U}=\left[\boldsymbol{u}^1-\boldsymbol{u}^2, \cdots, \boldsymbol{u}^1-\boldsymbol{u}^m\right]$, and $\boldsymbol{D}=\left[\boldsymbol{g}^1-\boldsymbol{g}^2, \cdots, \boldsymbol{g}^1-\boldsymbol{g}^m\right]$ The aggregated vector is then calculated as $\mathbf{g}=\sum_i^m w_i \mathbf{g}^i$.

C. Implementation Details

In this part, we provide implementation details regarding the empirical evaluation in the main paper, along with additional comparison experiments. All experiments are run on a single A100 GPU (40 GB VRAM).

C.1. Baselines

In this subsection, we briefly introduce some of the comparative methods that appeared in the main text:

- Linear scalarization (LS) minimizes the unweighted sum of task objectives $\sum_i^m \mathcal{L}^i(\boldsymbol{\theta})$.
- Scale-invariant (SI) aims toward obtaining similar convergent solutions even if losses are scaled with different coefficients via minimizing $\sum_i^m \log \mathcal{L}^i(\boldsymbol{\theta})$.
- Random loss weighting (RLW) [41] is a simple yet effective method for balancing task losses or gradients by random weights.
- Dynamic Weight Average (DWA) [47] simply adjusts the weighting coefficients by taking the rate of change of loss for each task into account.
- GradDrop [13] presents a probabilistic masking process that algorithmically eliminates all gradient values having the opposite sign w.r.t a predefined direction.

C.2. Image classification

Datasets.

- Multi-MNIST[+]. Following the protocol of [70], we set up three Multi-MNIST experiments with ResNet18 [25], namely: MultiFashion, MultiMNIST and MultiFashion+MNIST. In each dataset, two images are sampled uniformly from the MNIST [38] or Fashion-MNIST [75], then one is placed on the top left, and the other is on the bottom right. We thus obtain a two-task learning that requires predicting the categories of the digits or fashion items on the top left (task 1) and the bottom right (task 2), respectively.
- CelebA [52] is a face dataset with 200K images and 40 attributes, forming a 40-class multi-label classification problem.

Network Architectures. For two datasets in this problem, Multi-MNIST and CelebA, we replicate experiments from [42, 70] by respectively using the Resnet18 (11M parameters) and Resnet50 (23M parameters) [25] with the last output layer removed as the shared encoders and constructing linear classifiers as the task-specific heads, i.e. 2 heads for Multi-MNIST and 40 for CelebA, respectively.

Training Details. We train the all the models under our proposed framework and baselines using:

- Multi-MNIST: Adam optimizer [34] with a learning rate of 0.001 for 200 epochs using a batch size of 256. Images from the three datasets are resized to 36×36.
- CelebA: Batch-size of 256 and images are resized to $64 \times 64 \times 3$. Adam [34] is used again with a learning rate of 0.0005, which is decayed by 0.85 for every 10 epochs, our model is trained for 50 epochs in total.

Regarding the hyper-parameter for SAM [20], we use their adaptive version [35] where both ρ_{sh} and ρ_{ns} are set equally and extensively tuned from 0.005 to 5. More details can be found in the public source code.

C.3. Scene understanding

Two datasets used in this problem are NYUv2 and CityScapes:

- NYUv2[+] is an indoor scene dataset that contains 3 tasks: 13-class semantic segmentation, depth estimation, and surface normal prediction.

- CityScapes[+] has 19 classes of street-view images, which are coarsened into 7 categories to create two tasks: semantic segmentation and depth estimation.

Similar to [62], all images in the NYUv2 dataset are resized to 288×384 while all images in the CityScapes dataset are resized to 128×256 to speed up the training process. We follow the exact protocol in [62] for implementation. Specifically, SegNet [3] is adopted as the architecture for the backbone and Multi-Task Attention Network MTAN [47] is applied on top of it. We train each method for 200 epochs using Adam optimizer [34] with an initial learning rate of $1e^{-4}$ and reduce it to $5e^{-5}$ after 100 epochs. We use a batch size of 2 for NYUv2 and 8 for CityScapes. The last 10 epochs are averaged to get the final results, and all experiments are run with three random seeds. More details can be found in the public source code.

D. Additional Results

To further show the improvement of our proposed training framework over the conventional one, this section provides additional comparison results in terms of qualitative results, predictive performance, convergent behavior, loss landscape, model sharpness, and gradient norm. We also complete the ablation study in the main paper by providing results on all three datasets in the Multi-MNIST dataset.

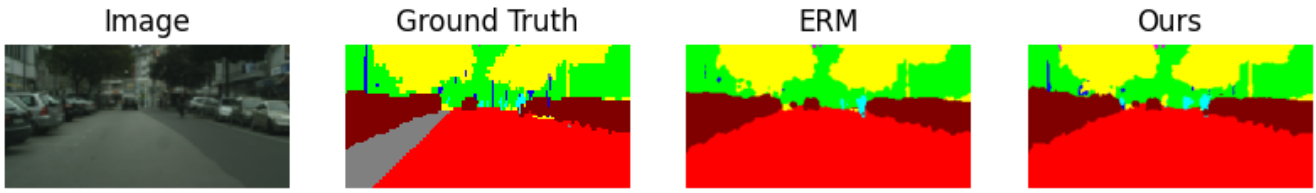
D.1. Image segmentation qualitative result

In this section, we provide qualitative results of our method of the CityScapes experiment. We compare our proposed method against its main baseline by highlighting typical cases where our method excels in generalization performance. Figure 4 shows some visual examples of segmentation outputs on the test set. Note that in the CityScapes dataset, the "void" class is identified as unclear and pixels labeled as void do not contribute to either objective or score [14].

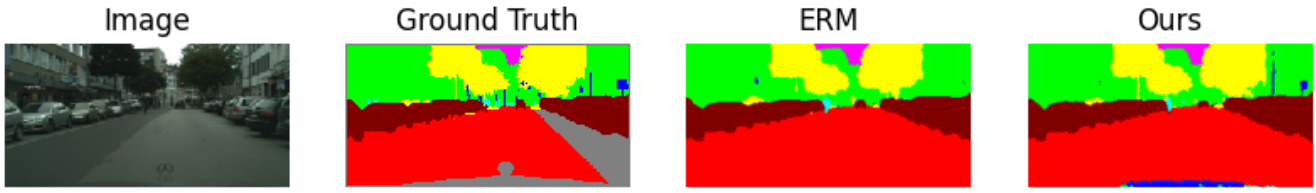
While there is only a small gap between the segmentation performance of ERM and ours, we found that a small area, which is the car hood and located at the bottom of images, is often incorrectly classified. For example, in Figure 4, the third and fourth rows compare the prediction of SegNet [3] with ERM training and with our proposed method. It can be

[+] https://github.com/Xi-L/ParetoMTL

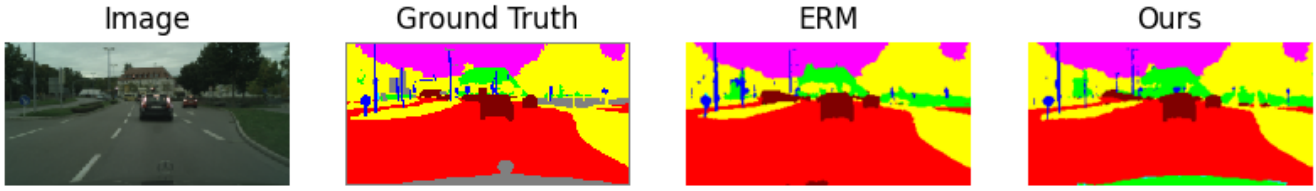
[+] https://github.com/Cranial-XIX/CAGrad



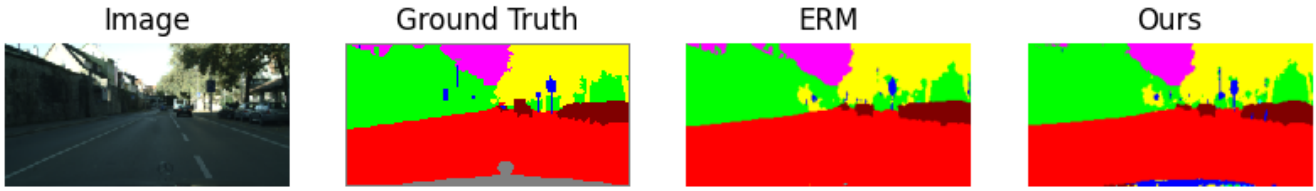
(a) A training sample (after augmentation)



(b) Corresponding original image (before augmentation)



(c) Predictions on an unseen image



(d) Predictions on an unseen image

Figure 4. Semantic segmentation prediction comparison on CityScapes . From left to right are input images, ground truth, and segmentation outputs from SegNet [3] using ERM training and sharpness-aware training. Regions that are represented in gray color are ignored during training. (Best viewed in color).

seen that both of them could not detect this area correctly, this is because this unclear "void" class did not appear during training. Even worse, the currently employed data augmentation technique in the codebase of Nash-MTL and other recent multi-task learning methods [44, 62] consists of RandomCrop, which often unintentionally excludes edge regions. For example, Figure 4a shows an example fed to the neural network for training, which excludes the car hood and its logo, compared to the original image (Figure 4b). Therefore, we can consider this "void" class as a novel class in this experiment, since its appearance is ignored in both training and evaluation. Even though, in Figures 4c and 4d our training method is still able to distinguish between this unknown area and other nearby known classes, which empirically shows the robustness and generalization ability of our method over ERM.

D.2. Predictive performance

In this part, we provide experimental justification for an intriguing insight into the connection between model sharpness and model calibration. Empirically, we found that when a model converges to flatter minima, it tends to be more calibrated. We start by giving the formal definition of a well-calibrated classification model and three metrics to measure the calibration of a model, then we analyze our empirical results.

Consider a C-class classification problem with a test set of N samples given in the form $(x_i, y_i)_{i=1}^N$ where y_i is the true label for the sample x_i. Model outputs the predicted probability for a given sample x_i to fall into C classes, is given by

$$\hat{\boldsymbol{p}}(x_i) = [\hat{p}(y = 1|x_i), \ldots, \hat{p}(y = C|x_i)].$$

$\hat{p}(y = c|x_i)$ is also the confidence of the model when assigning the sample x_i to class c. The predicted label $\hat{y}_i$ is the class with the highest predicted value, $\hat{p}(x_i) := \max_c \hat{p}(y = c|x_i)$. We refer to $\hat{p}(x_i)$ as the confidence score of a sample x_i.

Model calibration is a desideratum of modern deep neural networks, which indicates that the predicted probability of a model should match its true probability. This means that the classification network should be not only accurate but also confident about its prediction, i.e. being aware of when it is likely to be incorrect. Formally stated, the *perfect calibration* [22] is:

$$P(\hat{y} = y|\hat{p} = q) = q, \forall q \in [0, 1]. \tag{23}$$

Metric. The exact computation of Equation 23 is infeasible, thus we need to define some metrics to evaluate how well-calibrated a model is.

- Brier score ↓ (BS) [6] assesses the accuracy of a model's predicted probability by taking into account the absolute difference between its confidence for a sample to fall into a class and the true label of that sample. Formally,

$$BS = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \left(\hat{p}(y = c|x_i) - \mathbf{1}[y_i = c]\right)^2.$$

- Expected calibration error ↓ (ECE) compares the predicted probability (or confidence) of a model to its accuracy [22, 60]. To compute this error, we first bin the confidence interval $[0, 1]$ into M equal bins, then categorize data samples into these bins according to their confidence scores. We finally compute the absolute value of the difference between the average confidence and the average accuracy within each bin, and report the average value over all bins as the ECE. Specifically, let B_m denote the set of indices of samples having their confidence scores belonging to the m^{th} bin. The average accuracy and the average confidence within this bin are:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}[\hat{y}_i = y_i],$$

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}(x_i).$$

Then the ECE of the model is defined as:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)|.$$

In short, the lower ECE neural networks obtain, the more calibrated they are.

- Predictive entropy (PE) is a widely-used measure of uncertainty [36, 56, 65] via the predictive probability of the model output. When encountering an unseen sample, a well-calibrated model is expected to yield a high PE, representing its uncertainty in predicting out-of-domain (OOD) data.

$$PE = \frac{1}{C} \sum_{c=1}^{C} -\hat{p}(y = c|x_i) \log \hat{p}(y = c|x_i).$$

Figures 5 and 6 plot the distribution of the model's predicted entropy in the case of in-domain and out-domain testing, respectively. We can see when considering the flatness of minima, the model shows higher predictive entropy on both in-domain and out-of-domain, compared to ERM. This also means that our model outputs high uncertainty prediction when it is exposed to a sample from a different domain.

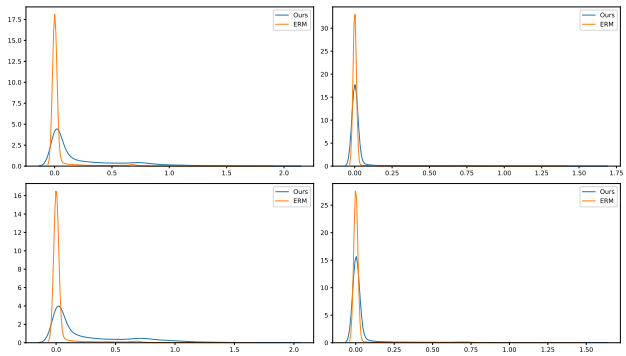


Figure 5. Histograms of predictive entropy of ResNet18 [25] on in-domain dataset, train and test on MultiMNIST (left) and MultiFashion (right). We use the orange lines to denote ERM training while blue lines indicate our proposed method.

Here, we calculate the results for both tasks 1 and 2 as a whole and plot their ECE in Figure 7. When we look

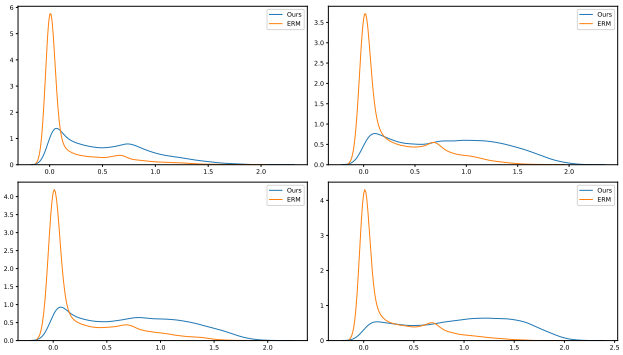


Figure 6. Out of domain: model is trained on MultiMNIST, then tested on MultiFashion (left) and vice versa (right). Models trained with ERM give over-confident predictions as their predictive entropy concentrates around 0.

at the in-domain prediction in more detail, our model still outperforms ERM in terms of expected calibration error. We hypothesize that considering a flat minima optimizer not only lowers errors across tasks but also improves the predictive performance of the model.

Dataset	Task	Multi-Fashion	Multi-Fashion+MNIST	MultiMNIST
ERM	Top left	0.237	0.055	0.082
	Bottom right	0.254	0.217	0.106
	Average	0.246	0.136	0.094
Ours	Top left	**0.172**	**0.037**	**0.059**
	Bottom right	**0.186**	**0.189**	**0.075**
	Average	**0.179**	**0.113**	**0.067**

Table 8. Brier score on Multi-Fashion, Multi-Fashion+MNIST and MultiMNIST datasets. We use the **bold** font to highlight the best results.

We also report the Brier score and ECE for each task in Table 8 and Table 9. As can be observed from these tables, our method shows consistent improvement in the model calibration when both scores decrease over all scenarios.

Dataset	Task	Multi-Fashion	Multi-Fashion+MNIST	MultiMNIST
ERM	Top left	0.113	0.027	0.039
	Bottom right	0.121	0.104	0.050
	Average	0.117	0.066	0.045
Ours	Top left	**0.034**	**0.015**	**0.022**
	Bottom right	**0.032**	**0.083**	**0.028**
	Average	**0.033**	**0.049**	**0.025**

Table 9. Expected calibration error on Multi-Fashion, Multi-Fashion+MNIST and MultiMNIST datasets. Here we set the number of bins equal to 10.

D.3. Effect of choosing perturbation radius ρ.

The experimental results analyzing the sensitivity of the model w.r.t ρ are given in Figure 8. We evenly picked ρ from 0 to 3.0 to run experiments on three Multi-MNIST datasets. We find that the average accuracy of each task is rather stable from $\rho = 0.5$, which means the effect of different values of ρ in a reasonably small range is similar. It can also be easy to notice that the improvement tends to saturate when $\rho \geq 1.5$.

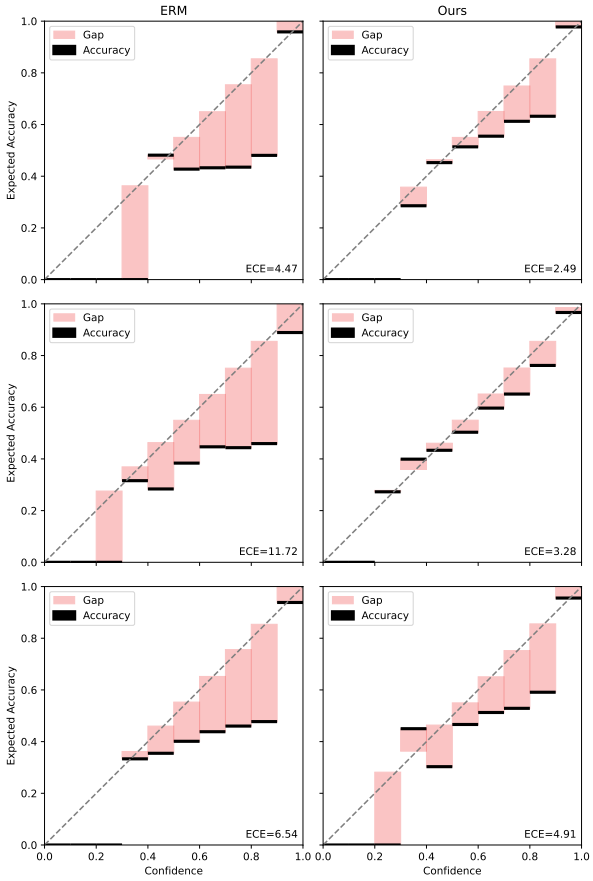


Figure 7. The predictive performance (measured by the expected calibration error) of neural networks has been enhanced by using our proposed training method (right column).

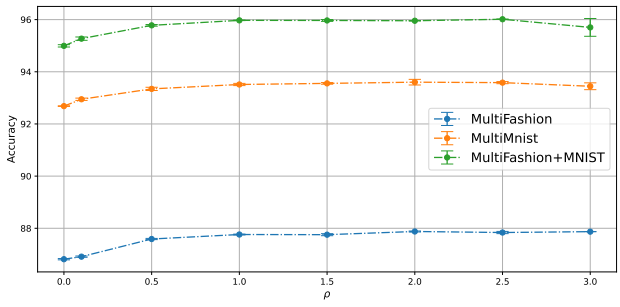
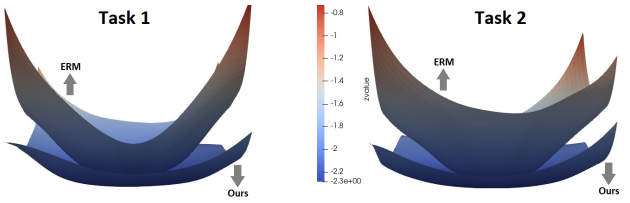


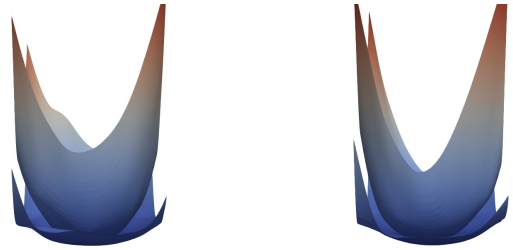
Figure 8. Average accuracy when varying ρ from 0 to 3.0 (with error bar from three independent runs).

D.4. Loss landscape

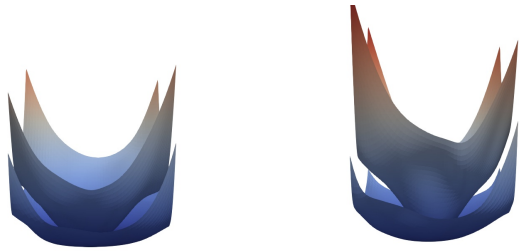
Firstly, following [39], we provide additional visual comparisons of the loss landscapes trained with standard training and with our framework across two tasks of three datasets of Multi-MNIST in Figure 9. These are test loss surfaces of checkpoints that have the highest validation accuracy. The solution found by our proposed method not only mitigates the test loss sharpness for both tasks but also can reduce the test loss value itself, in comparison to traditional ERM. This is a common behavior when using flat minimizers as the gap between train and test performance has been narrowed [27, 30].



(a) MultiMNIST



(b) MultiFashion



(c) MultiFashion+MNIST

Figure 9. Loss landscapes of task 1 and task 2 on MultiMNIST, MultiFashion and MultiFashion+MNIST, respectively.

D.5. Model robustness against weight perturbation

Thirdly, to verify that SAM can orient the model to the common flat and low-loss region of all tasks, we measure the model performance within a r-radius Euclidean ball. To be more specific, we perturb parameters of two converged models by ϵ, which lies in a r-radius ball and plot the accuracy of the perturbed models of each task as we increase r from 0 to 1000. At each value of r, 10 different models around the $r-$radius ball of the converged model are sampled.

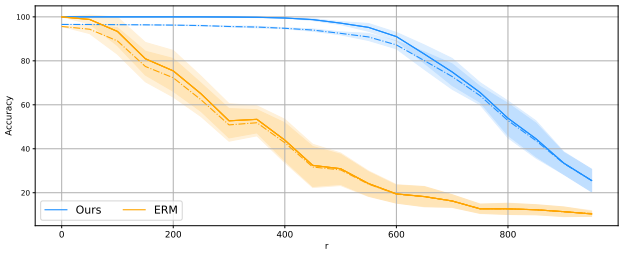


Figure 10. **Accuracy** within r-radius ball. Solid/dashed lines denote performance on train/test sets.

In Figure 10, the accuracy of the model trained using our method remains at a high level when noise keeps increasing until $r = 800$. This also gives evidence that our model found a region that changes slowly in loss. By contrast, the naively trained model loses its predictive capabilities as soon as the noise appears and becomes a dummy classifier that attains 10% accuracy in a 10-way classification.

D.6. Gradient conflict

Secondly, in the main paper, we measure the percentage of gradient conflict on the MultiFashion+MNIST dataset. Here, we provide the full results on three different datasets. As can be seen from Figure 11, there is about half of the mini-batches lead to the conflict between task 1 and task 2 when using traditional training. Conversely, our proposed method significantly reduces such confliction to less than 5% via updating the parameter toward flat regions.

D.7. Training curves

Thirdly, we compare the test accuracy of trained models under the two settings in Fig. 12. It can be seen that from the early epochs (20-th epoch), the *flat-based* method outperforms the *ERM-based* method on all tasks and datasets. Although the ERM training model is overfitted after such a long training, our model retains a high generalizability, as discussed throughout previous sections.

Furthermore, we also plot the training accuracy curves across experiments in Figure 13 to show that training accuracy scores of both ERM and our proposed method are similar and reach $\approx$ 100% from 50-th epoch, which illustrates that the improvement is associated with generalization enhancement, not better training.

D.8. Model sharpness

Fourthly, Figure 14 displays the evolution of ρ-sharpness of models along training epochs under conventional loss function (ERM) and worst-case loss function (ours) on training sets of three datasets from Multi-MNIST, with

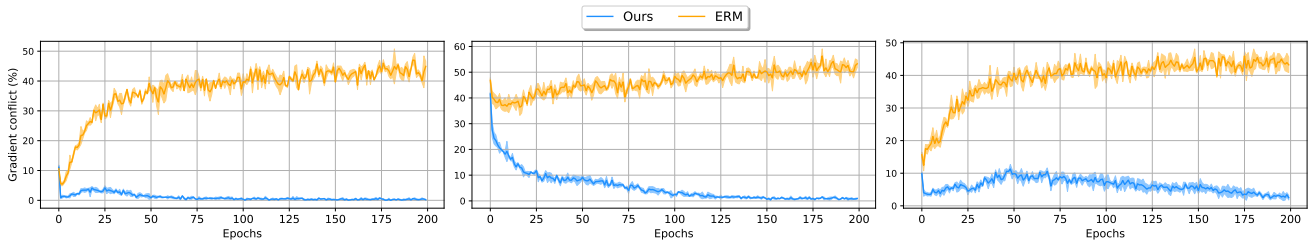


Figure 11. **Task gradient conflict proportion** of models trained with our proposed method and ERM across MultiFashion, MultiFashion+MNIST, and MultiMNIST datasets (columns).

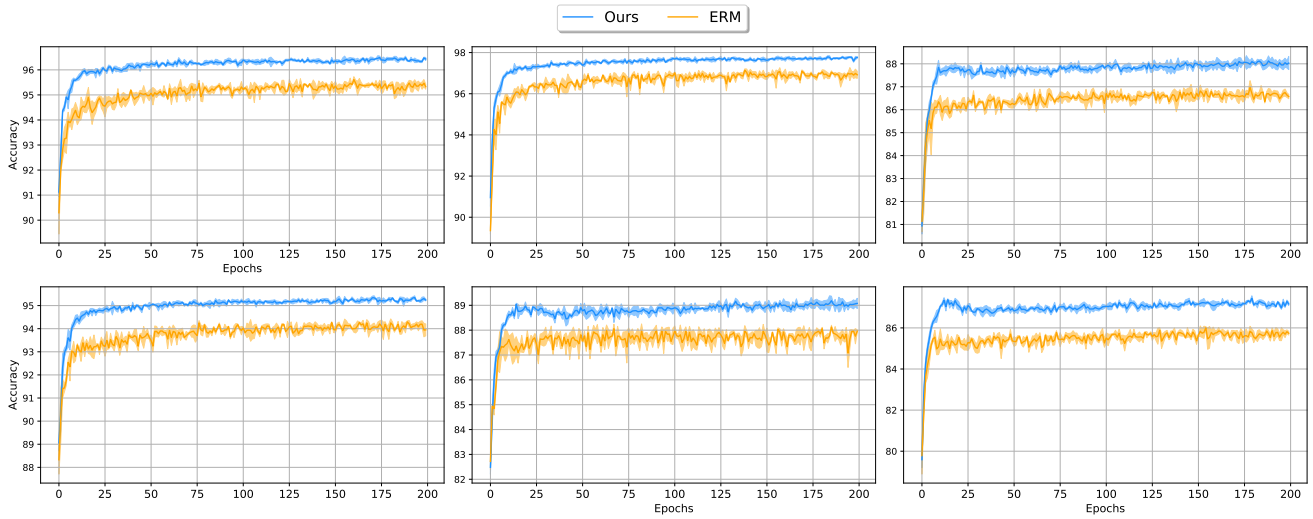


Figure 12. **Test accuracy** of models trained with our proposed method and ERM across 2 tasks (rows) of MultiFashion, MultiFashion+MNIST and MultiMNIST datasets (columns).

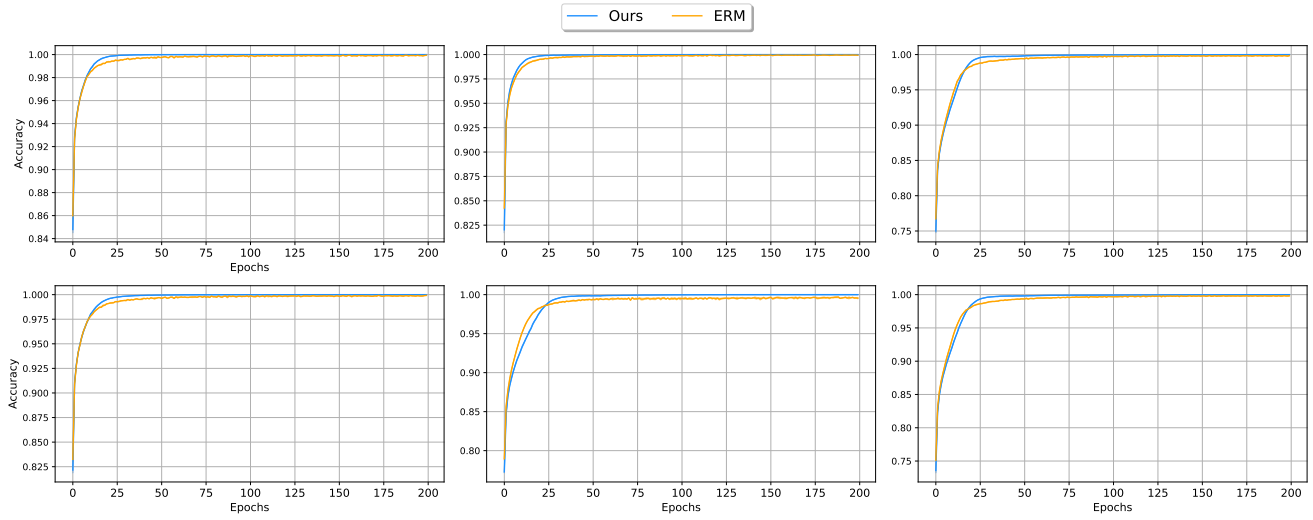


Figure 13. **Train accuracy** of models trained with our proposed method and ERM across 2 tasks (rows) of MultiFashion, MultiFashion+MNIST and MultiMNIST datasets (columns).

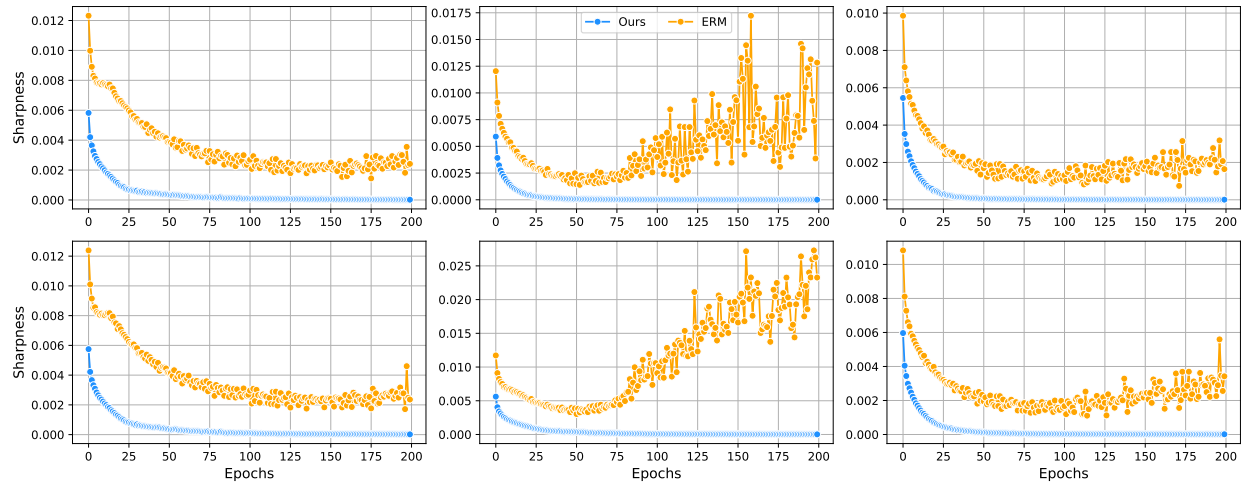
multiple values of ρ. We can clearly see that under our framework, for both tasks, the model can guarantee uniformly low loss value in the ρ-ball neighborhood of parameter across training process. In contrast, ERM suffers from sharp minima from certain epochs when the model witnesses a large gap between the loss of worst-case perturbed model and current model. This is the evidence for the benefit that our framework brings to gradient-based methods, which is all tasks can concurrently find flat minima thus achieving better generalization.

D.9. Gradient norm

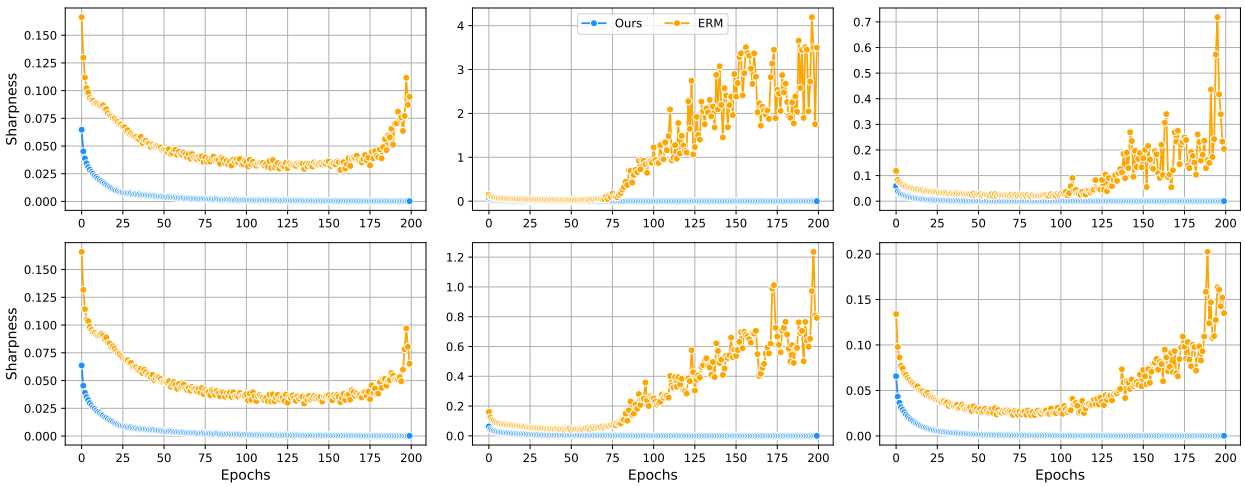
Finally, we demonstrate the gradient norm of the loss function w.r.t the worst-case perturbed parameter of each task. On the implementation side, we calculate the magnitude of the flat gradient $\boldsymbol{g}^{i,\text{flat}}$ for each task at different values of ρ in Figure 15. As analyzed by Equation 6 from the main paper, following the negative direction of $\boldsymbol{g}_{sh}^{i,SAM}$ will lower the $\mathcal{L}_2$ norm of the gradient, which orients the model towards flat regions. This is empirically verified in Figure 15. In contrast, as the number of epochs increases, gradnorm of the model trained with ERM tends to increase or fluctuate around a value higher than that of model trained with SAM.

E. Discussion and Limitations

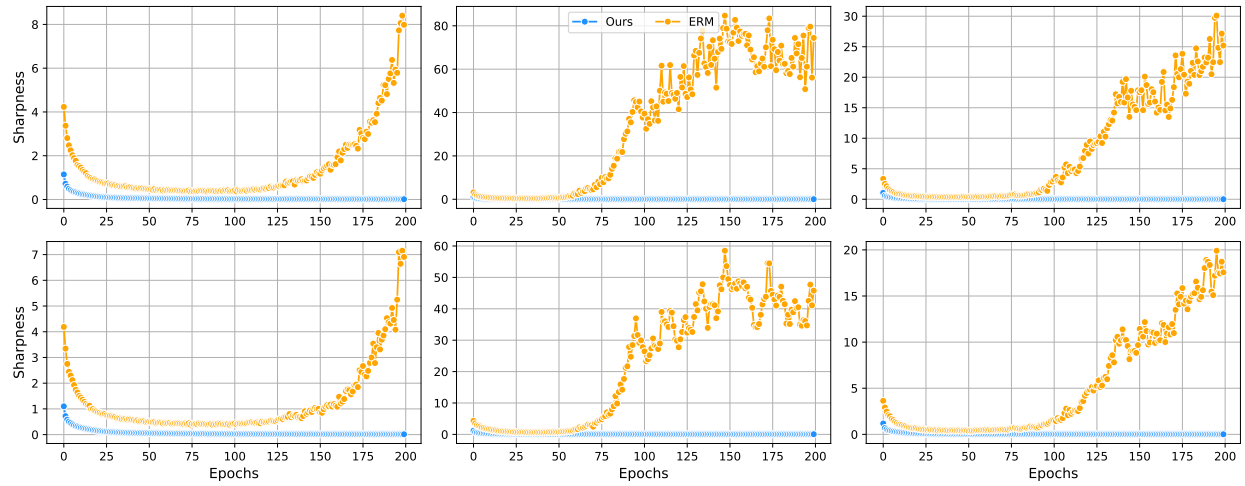
The primary limitations of our work lie in time and space complexity. Specifically, our method demands an additional forward-backward pass to compute the worst-case gradient for each task, resulting in approximately twice the runtime compared to ERM counterparts. This could potentially be mitigated by employing a periodic update strategy as in [50], or by applying weight perturbation on a randomly chosen set of weights and data [18], or even applying proposed training procedure on last few epochs [82]. However, we leave this exploration for future work, as the main focus of our paper is to demonstrate the effectiveness of encouraging flatness in MTL. In terms of space complexity, our approach requires approximately double the memory compared to traditional gradient-based methods. This is due to the need to store both the flat gradient and the loss gradient for each task as part of our gradient decomposition process.



(a) $\rho = 0.005$

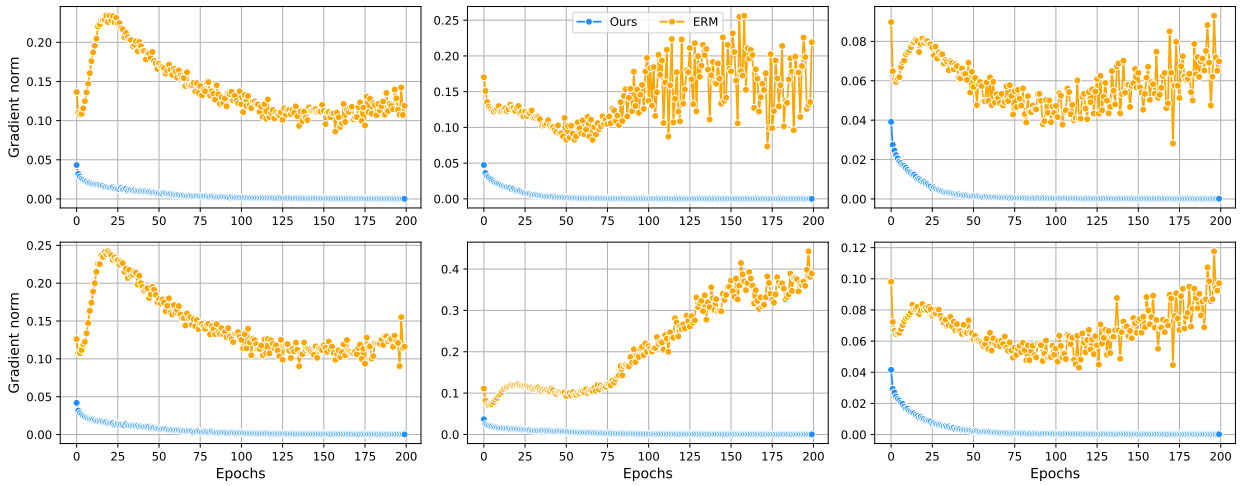


(b) $\rho = 0.05$

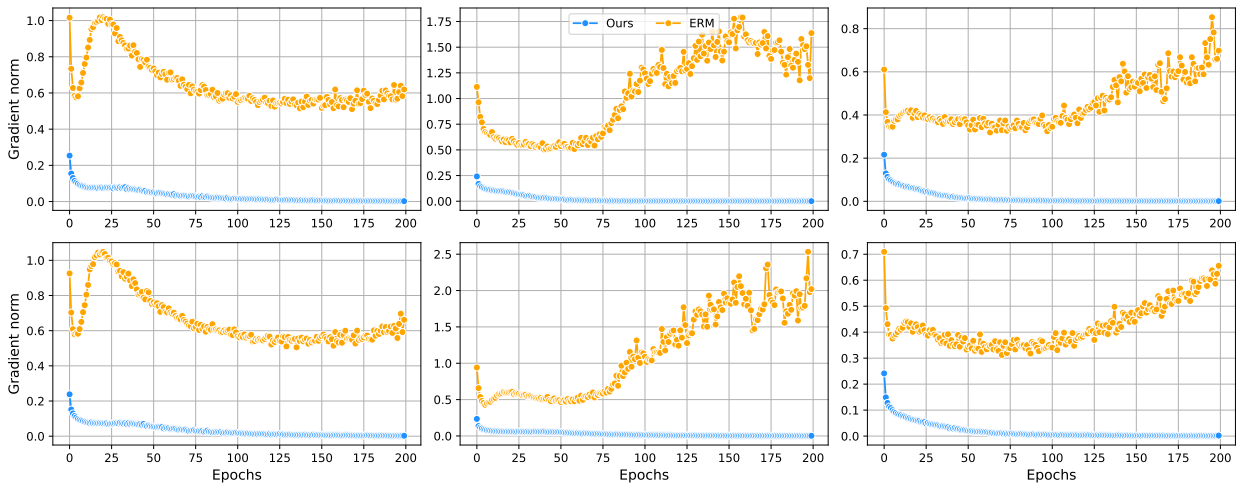


(c) $\rho = 0.5$

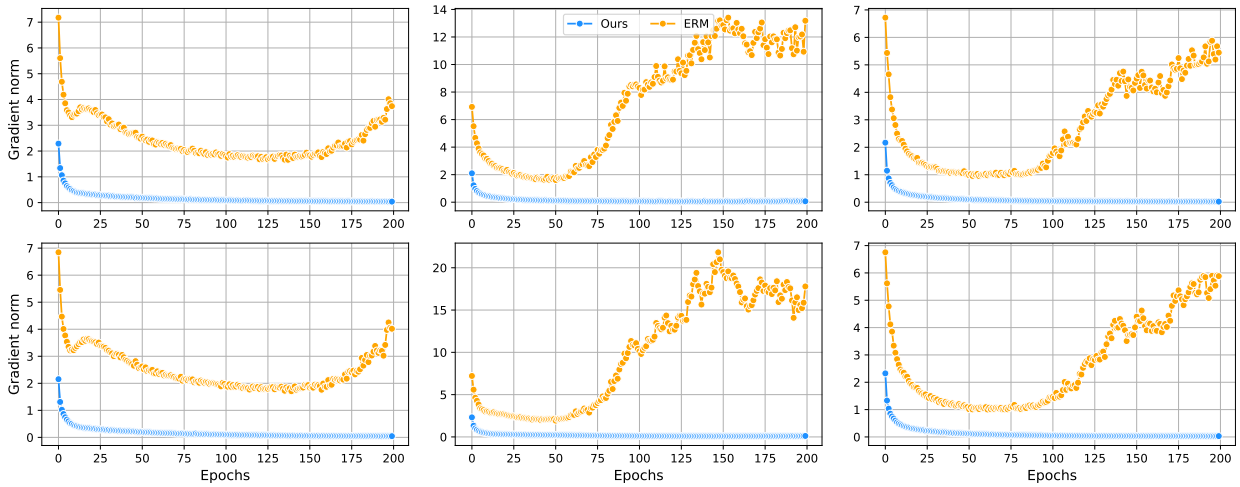
Figure 14. **Sharpness** of models trained with our proposed method and ERM with different values of ρ. For each ρ, the top and bottom row respectively represents the first and second task, and each column respectively represents each dataset in Multi-MNIST: from left to right are MultiFashion, MultiFashion+MNIST, MultiMNIST.



(a) $\rho = 0.005$



(b) $\rho = 0.05$



(c) $\rho = 0.5$

Figure 15. **Gradient magnitude** at the worst-case perturbations of models trained with our proposed method and ERM with different values of ρ. For each ρ, the top and bottom row respectively represents the first and second task, and each column respectively represents each dataset in Multi-MNIST: from left to right are MultiFashion, MultiFashion+MNIST, MultiMNIST.