Quantifying Task Priority for Multi-Task Optimization

Supplementary Material

A. Theoretical Analysis

A.1. Proof of Theorem 1

Theorem 1. *Updating gradients based on task priority for shared parameters Θ_s (update g_i for each $\theta_{s,i}$) results in a smaller multi-task loss $\sum_{i=1}^{\mathcal{K}} w_i\mathcal{L}_i$ compared to updating the weighted summation of task-specific gradients $\sum_{i=1}^{\mathcal{K}} \nabla w_i\mathcal{L}_i$ without considering task priority.*

Proof. We start from shared parameters Θ_s and we can divide them with task priority.

$$\Theta_s = \{\theta_{s,1}, \theta_{s,2}, ..., \theta_{s,\mathcal{K}}\} \tag{11}$$

Let $\tilde{\Theta}_{s,i}$ represent the parameters in Θ_s, excluding $\theta_{s,i}$. For the sake of simplicity in our proof, we begin by focusing on a subset of shared parameters, specifically $\theta_{s,i}$, to demonstrate that accounting for task priority leads to a reduced multi-task loss compared to neglecting it. Subsequently, we will apply the same process to the remaining shared parameters to complete the proof. Let $\hat{\mathrm{g}}_k^t$ be the gradient of $\theta_{s,i}^t$ for task τ_k as follows:

$$\hat{\mathrm{g}}_k^t = \nabla_{\theta_{s,i}^t}\mathcal{L}_k(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_k^t) \tag{12}$$

Previous optimization methods involving gradient manipulation update the weighted summation of task-specific gradients. Therefore, we can update $\theta_{s,i}^t$ to $\theta_{s,i}^{t+1}$ as follows:

$$\mathrm{g}^t = \sum_{j=1}^{\mathcal{K}} \nabla_{\theta_{s,i}^t} w_j\mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_i^t) = \sum_{j=1}^{\mathcal{K}} w_j\hat{\mathrm{g}}_j^t, \qquad \theta_{s,i}^{t+1} = \theta_{s,i}^t - \eta\mathrm{g}^t \tag{13}$$

where w_i is loss weights of τ_i and $\sum_{i=1}^{\mathcal{K}} w_i = 1$.
From the first order Taylor approximation of $\mathcal{L}_i$ for $\theta_{s,i}$, we have

$$\mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^{t+1}, \Theta_i^t) = \mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_i^t) + (\theta_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_i^t + O(\eta^2) \tag{14}$$

On the other hand, when considering task priority, we can update $\theta_{s,i}^t$ to $\hat{\theta}_{s,i}^{t+1}$ using $\hat{\mathrm{g}}_i$ as follows:

$$\hat{\theta}_{s,i}^{t+1} = \theta_{s,i}^t - \eta\hat{\mathrm{g}}_i^t \tag{15}$$

From the first order Taylor approximation of $\mathcal{L}_i$ from $\theta_{s,i}^t$ to $\hat{\theta}_{s,i}^{t+1}$, we have

$$\mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \hat{\theta}_{s,i}^{t+1}, \Theta_i^t) = \mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_i^t) + (\hat{\theta}_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_i^t + O(\eta^2) \tag{16}$$

The difference between Eq. (14) and Eq. (16) is

$$\begin{aligned}\mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^{t+1}, \Theta_i^t) - \mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \hat{\theta}_{s,i}^{t+1}, \Theta_i^t) &= (\theta_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_i^t - (\hat{\theta}_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_i^t &(17)\\ &= -\eta(\mathrm{g}^t - \hat{\mathrm{g}}_i^t)^\top\hat{\mathrm{g}}_i^t &(18)\end{aligned}$$

Similar to Eq. (14) and Eq. (16), we have the following two inequalities for the last of the losses $\mathcal{L}_j$ where $i \neq j$:

$$\mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^{t+1}, \Theta_i^t) = \mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_i^t) + (\theta_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_j^t + O(\eta^2) \tag{19}$$

$$\mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \hat{\theta}_{s,i}^{t+1}, \Theta_i^t) = \mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}_{s,i}^t, \theta_{s,i}^t, \Theta_i^t) + (\hat{\theta}_{s,i}^{t+1} - \theta_{s,i}^t)^\top\hat{\mathrm{g}}_j^t + O(\eta^2) \tag{20}$$

The result in Eq. (19) corresponds to updating the weighted summation of task-specific gradients, while Eq. (20) reflects the result when updating gradients with consideration for task priority.

The difference between Eq. (19) and Eq. (20) is

$$\mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^{t+1}_{s,i}, \Theta^t_i) - \mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \hat{\theta}^{t+1}_{s,i}, \Theta^t_i) = (\theta^{t+1}_{s,i} - \theta^t_{s,i})^\top \hat{\mathrm{g}}^t_j - (\hat{\theta}^{t+1}_{s,i} - \theta^t_{s,i})^\top \hat{\mathrm{g}}^t_j \tag{21}$$

$$= -\eta(\mathrm{g}^t - \hat{\mathrm{g}}^t_i)^\top \hat{\mathrm{g}}^t_j \tag{22}$$

If we sum Eq. (22) over all task losses $\{\mathcal{L}_k\}_{k=1}^{\mathcal{K}}$ along with their corresponding task-specific weights $\{w_k\}_{k=1}^{\mathcal{K}}$, the following result is obtained:

$$\sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^{t+1}_{s,i}, \Theta^t_i) - \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \hat{\theta}^{t+1}_{s,i}, \Theta^t_i) \tag{23}$$

$$= -\eta \sum_{k=1}^{\mathcal{K}} w_k (\mathrm{g}^t - \hat{\mathrm{g}}^t_i)^\top \hat{\mathrm{g}}^t_k \tag{24}$$

$$= -\eta \sum_{k=1}^{\mathcal{K}} w_k (\sum_{j=1}^{\mathcal{K}} \nabla_{\theta^t_{s,i}} w_j \mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^t_{s,i}, \Theta^t_i) - \nabla_{\theta^t_{s,i}} \mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^t_{s,i}, \Theta^t_i))^\top \hat{\mathrm{g}}^t_k \tag{25}$$

$$= -\eta \sum_{k=1}^{\mathcal{K}} w_k \left(\sum_{j=1}^{\mathcal{K}} w_j \left(\nabla_{\theta^t_{s,i}} \mathcal{L}_j(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^t_{s,i}, \Theta^t_i) - \nabla_{\theta^t_{s,i}} \mathcal{L}_i(\mathcal{X}^t, \tilde{\Theta}^t_{s,i}, \theta^t_{s,i}, \Theta^t_i) \right) \right)^\top \hat{\mathrm{g}}^t_k \tag{26}$$

$$\geq 0 \tag{27}$$

The elements within the brackets of Eq. (26) represent a pairwise comparison of the changes in loss resulting from updating the gradients of each task. Thus, the inequality of Eq. (27) holds from Definition 3 of task priority. The results indicate that taking task priority into account yields a lower multi-task loss compared to neglecting it. Following a similar process for all shared parameters $\Theta_s = \{\theta_{s,1}, \theta_{s,2}, ..., \theta_{s,\mathcal{K}}\}$, we can conclude considering task priority leads to the expansion of the known Pareto frontier.

□

A.2. Convergence Analysis

This section provides theoretical analyses of the proposed optimization method, including a convergence analysis. The overview is as follows:

1. We present the concept of Pareto-stationarity. Previous methods [26, 36, 37, 45] have shown their convergence to Pareto stationary points in multi-task optimization. (See Appendix A.2.1).
2. We offer a convergence analysis for Phase 1 of connection strength-based optimization. The analysis is conducted separately for shared and task-specific parameters. For task-specific parameters, it converges to the Pareto optimal point, similar to simple gradient descent. However, for shared parameters, Phase 1 doesn't ensure convergence to the Pareto optimal point; instead, it enhances the correlation between the gradients of tasks. (See Appendix A.2.2)
3. We provide the convergence rate of Phase 1, with a focus on task-specific parameters. (See Appendix A.2.3)
4. We present a convergence analysis for Phase 2 of connection strength-based optimization, specifically focusing on the shared parameters of the network. Our analysis shows that Phase 2 converges to the Pareto optimal point, distinguishing it from previous works that converge to Pareto stationary points. (See Appendix A.2.4)
5. We provide the convergence rate of Phase 2. (See Appendix A.2.5)

A.2.1 Pareto-stationarity

Initially, we establish the concept of a Pareto stationary point. Previous methods [26, 36, 37, 45] have shown their convergence to Pareto stationary points in multi-task optimization.

Definition 5 (Pareto stationarity)**.** *The network parameter Θ is defined with task-specific losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$. If the sum of weighted gradients $\sum_{i=1}^{\mathcal{K}} w_i \nabla_\Theta \mathcal{L}_i = 0$, then the point is termed Pareto stationary, indicating the absence of a descent direction from that point.*

Previous research [26, 36, 37, 45] has demonstrated their convergence to Pareto stationary points, which carries the risk of leading to sub-optimal solutions. This is due to the fact that Pareto-stationarity is a necessary condition for Pareto-optimality. In contrast, our work establishes convergence to the Pareto optimal point during Phase 2 of connection strength-based optimization. Phase 1 doesn't assure attainment of the Pareto optimal solution. Instead, it enhances the correlation between task gradients, amplifying the significance of task-specific parameters to learn task priorities.

A.2.2 Convergence of Phase 1

In the subsequent convergence analysis, we omit the input $\mathcal{X}^t$ for clarity.

Theorem 2 (Convergence of Phase 1)**.** *Assume losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ are convex and differentiable and the gradient of $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ is Lipschitz continuous with constant $H > 0$, i.e. $||\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}_i(y)|| \leq H||x-y||$ for $i = 1, 2, ..., \mathcal{K}$. Phase 1 of connection strength optimization, with a step size $\eta \leq \frac{1}{H}$, will converge to the Pareto optimal point for task-specific parameters $\{\Theta_i\}_{i=1}^{\mathcal{K}}$. For shared parameters Θ_s with a step size $\eta \leq \frac{2}{H}$, it does not guarantee convergence to the Pareto optimal point, but it optimizes in the direction to increase the correlation between tasks' gradients.*

Proof. We begin by conducting a quadratic expansion of the task-specific loss $\mathcal{L}_i(\Theta_s^t, \Theta_i^t)$ concerning the parameters Θ_s^t and Θ_i^t at each update step of Phase 1 for sequential tasks.

$$
\begin{aligned}
\mathcal{L}_i(\Theta_s^{t+i/\mathcal{K}}, \Theta_i^{t+1}) \leq & \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t) && (28)\\
& + \nabla_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}}) && (29)\\
& + \frac{1}{2}\nabla^2_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}})^2 && (30)\\
& + \nabla_{\Theta_i^t} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_i^{t+1} - \Theta_i^t) && (31)\\
& + \frac{1}{2}\nabla^2_{\Theta_i^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_i^{t+1} - \Theta_i^t)^2 && (32)\\
\leq & \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t) && (33)\\
& + \nabla_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}}) && (34)\\
& + \frac{1}{2}H(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}})^2 && (35)\\
& + \nabla_{\Theta_i^t} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_i^{t+1} - \Theta_i^t) && (36)\\
& + \frac{1}{2}H(\Theta_i^{t+1} - \Theta_i^t)^2 && (37)
\end{aligned}
$$

for $i = 1, 2, ..., \mathcal{K}$. The inequality in Eq. (33) holds as $\nabla\mathcal{L}$ is Lipschitz continuous with constant H which implies that $\nabla^2\mathcal{L} - HI \leq 0$. We follow the gradient update rule for Phase 1 in connection strength-based optimization:

$$
\begin{aligned}
\Theta^{t+i/\mathcal{K}} = \Theta^{t+(i-1)/\mathcal{K}} - \eta w_i \nabla_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t) && (38)\\
\Theta_i^{t+1} = \Theta_i^t - \eta w_i \nabla_{\Theta_i^t} \mathcal{L}_i(\Theta_i^t) && (39)
\end{aligned}
$$

for $i = 1, 2, ..., \mathcal{K}$. To simplify the proof, we partition the equation into two subsets—one for shared parameters Θ_s and the other for task-specific parameters Θ_i.
(i) For task-specific parameter Θ_i, the following inequality holds:

$$
\mathcal{L}_i(\Theta_s^t, \Theta_i^{t+1}) \leq \mathcal{L}_i(\Theta_s^t, \Theta_i^t) + \nabla_{\Theta_i^t} \mathcal{L}_i(\Theta_s^t, \Theta_i^t)(\Theta_i^{t+1} - \Theta_i^t) + \frac{1}{2}H(\Theta_i^{t+1} - \Theta_i^t)^2 \quad (40)
$$

We denote g_i^t as the gradient of Θ_i^t for task τ_i as follows:

$$
\mathrm{g}_i^t = \nabla_{\Theta_i^t} \mathcal{L}_i(\Theta_s^t, \Theta_i^t) \quad (41)
$$

If we substitute Eq. (39) into Eq. (40), it becomes as follows:

$$\mathcal{L}_i(\Theta_s^t, \Theta_i^{t+1}) \leq \mathcal{L}_i(\Theta_s^t, \Theta_i^t) - \eta w_i ||\mathrm{g}_i^t||^2 + \frac{\eta_i^2 w_i^2}{2} H ||\mathrm{g}_i^t||^2 \tag{42}$$

$$= \mathcal{L}_i(\Theta_s^t, \Theta_i^t) - \eta w_i (1 - \frac{1}{2}\eta w_i H) ||\mathrm{g}_i^t||^2 \tag{43}$$

$$\leq \mathcal{L}_i(\Theta_s^t, \Theta_i^t) - \frac{1}{2}\eta w_i ||\mathrm{g}_i^t||^2 \tag{44}$$

Eq. (44) is valid when the step size η is sufficiently small, specifically, when $\eta \leq \frac{1}{H w_i}$. When we sum Eq. (44) over all task losses $\{\mathcal{L}_k\}_{k=1}^{\mathcal{K}}$ along with their corresponding task-specific weights $\{w_k\}_{k=1}^{\mathcal{K}}$, the following result is obtained:

$$\sum_{i=1}^{\mathcal{K}} w_i \mathcal{L}_i(\Theta_s^t, \Theta_i^{t+1}) \leq \sum_{i=1}^{\mathcal{K}} w_i \mathcal{L}_i(\Theta_s^t, \Theta_i^t) - \frac{1}{2}\sum_{i=1}^{\mathcal{K}} \eta w_i^2 ||\mathrm{g}_i^t||^2 \tag{45}$$

According to Eq. (45), we can infer that the application of Phase 1 in connection strength-based optimization can result in $\mathrm{g}_i = 0$ for $i = 1, 2, ..., \mathcal{K}$. The condition $g_i^t = 0$ indicates that the proposed updating rule converges to the Pareto-optimal point for task-specific parameters Θ_i for $i = \{1, 2, ..., \mathcal{K}\}$.

(ii) For shared parameter Θ_s, the following inequality holds:

$$\mathcal{L}_i(\Theta_s^{t+i/\mathcal{K}}, \Theta_i^t) \leq \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t) + \nabla_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t)(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}}) \tag{46}$$

$$+ \frac{1}{2} H (\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}})^2 \tag{47}$$

In case (ii), we denote g_i^t as the gradient of Θ_s^t for task τ_i, and g^t as the weighted sum of $\{\mathrm{g}_i^t\}_{i=1}^{\mathcal{K}}$ with $\{w_i\}_{i=1}^{\mathcal{K}}$ as follows:

$$\mathrm{g}_i^t = \nabla_{\Theta_s^t} \mathcal{L}_i(\Theta_s^t, \Theta_i^t), \qquad \mathrm{g}^t = \sum_{i=1}^{\mathcal{K}} w_i \nabla \mathcal{L}_i(\Theta_s^t, \Theta_i^t) \tag{48}$$

If we substitute Eq. (38) into Eq. (46) and Eq. (47), it becomes as follows:

$$\mathcal{L}_i(\Theta_s^{t+i/\mathcal{K}}, \Theta_i^t) \leq \mathcal{L}_i(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_i^t) - \eta w_i ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 + \frac{\eta^2 w_i^2}{2} H ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 \tag{49}$$

Similarly, the quadratic expansion of $\mathcal{L}_j$ for $\Theta_s^{t+i/\mathcal{K}}$ when $i \neq j$ is as follows:

$$\mathcal{L}_j(\Theta_s^{t+i/\mathcal{K}}, \Theta_j^t) \leq \mathcal{L}_j(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_j^t) + \nabla_{\Theta_s^{t+(i-1)/\mathcal{K}}} \mathcal{L}_j(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_j^t)(\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}}) \tag{50}$$

$$+ \frac{1}{2} H (\Theta_s^{t+i/\mathcal{K}} - \Theta_s^{t+(i-1)/\mathcal{K}})^2 \tag{51}$$

$$\leq \mathcal{L}_j(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_j^t) - \eta w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}} \cdot \mathrm{g}_j^{t+(i-1)/\mathcal{K}} + \frac{\eta^2 w_i^2}{2} H ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 \tag{52}$$

When we sum Eq. (49) and Eq. (52) over all task losses $\{\mathcal{L}_k\}_{k=1}^{\mathcal{K}}$ along with their corresponding task-specific weights $\{w_k\}_{k=1}^{\mathcal{K}}$, the following result is obtained:

$$\sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta_s^{t+i/\mathcal{K}}, \Theta_k^t) \leq \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_k^t) - \eta w_i \sum_{k=1}^{\mathcal{K}} w_k \mathrm{g}_i^{t+(i-1)/\mathcal{K}} \cdot \mathrm{g}_k^{t+(i-1)/\mathcal{K}} + \frac{\eta^2 w_i^2}{2} H ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 \tag{53}$$

$$= \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_k^t) - \eta w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}} \cdot \mathrm{g}^{t+(i-1)/\mathcal{K}} + \frac{\eta^2 w_i^2}{2} H ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 \tag{54}$$

$$\leq \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_k^t) - \eta w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}} \cdot \mathrm{g}^{t+(i-1)/\mathcal{K}} + \eta w_i^2 ||\mathrm{g}_i^{t+(i-1)/\mathcal{K}}||^2 \tag{55}$$

$$= \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta_s^{t+(i-1)/\mathcal{K}}, \Theta_k^t) - \eta w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}} \cdot (\mathrm{g}^{t+(i-1)/\mathcal{K}} - w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}}) \tag{56}$$

Eq. (55) is valid when the step size η is sufficiently small, specifically, when $\eta \leq \frac{2}{H}$. As shown in Eq. (56), Phase 1 of connection strength-based optimization does not strictly ensure convergence. This is attributed to its sequential updating of task-specific connections, leading to fluctuations in their losses during training. Nevertheless, as illustrated in Eq. (56), we can note that the optimization moves in the direction of minimizing the dot product between the gradient of the currently updated task $\mathrm{g}_i^{t+(i-1)/\mathcal{K}}$ and the weighted sum of gradients from the remaining losses $(\mathrm{g}^{t+(i-1)/\mathcal{K}} - w_i \mathrm{g}_i^{t+(i-1)/\mathcal{K}})$. This observation aligns with the experimental results presented in Fig. 3. Phase 1 effectively increases the correlation between tasks in shared parameters Θ_s, which exaggerates the role of task-specific parameters, allowing it to sufficiently grasp and establish task priorities.

□

A.2.3 Convergence rate of Phase 1

Theorem 3 (Convergence rate of Phase 1)**.** *Assume losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ are convex and differentiable and the gradient of $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ is Lipschitz continuous with constant $H > 0$, i.e. $||\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}_i(y)|| \leq H||x-y||$ for $i = 1, 2, ..., \mathcal{K}$. Then, in phase 1 of connection strength optimization with a step size $\eta \leq \frac{1}{H}$, the system will reach the Pareto optimal point for task-specific parameters $\{\Theta_i\}_{i=1}^{\mathcal{K}}$ at a rate of $O(1/T)$, where T is the total number of iterations. This is guaranteed by the following inequality:*

$$\min_{0\leq t\leq T} \sum_{k=1}^{\mathcal{K}} w_k^2 ||\mathrm{g}_k^t||^2 \leq \frac{2}{\eta T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^*)) \tag{57}$$

where Θ^* represents the converged parameters, and T is the total number of iterations.

Proof. We begin with the result from Eq. (45). To simplify, let $\mathcal{L}$ represent the total loss, defined as $\mathcal{L}(\Theta^t) = \sum_{i=1}^{\mathcal{K}} w_i \mathcal{L}_i(\Theta^t)$. We only consider task-specific parameters $\{\Theta_i\}_{i=1}^{\mathcal{K}}$ for analysis.

$$\mathcal{L}(\Theta^{t+1}) \leq \mathcal{L}(\Theta^t) - \frac{1}{2}\sum_{i=1}^{\mathcal{K}} \eta w_i^2 ||\mathrm{g}_i^t||^2 \tag{58}$$

By rearranging the term in Eq. (58):

$$\sum_{i=1}^{\mathcal{K}} w_i^2 ||\mathrm{g}_i^t||^2 \leq \frac{2}{\eta}(\mathcal{L}(\Theta^t) - \mathcal{L}(\Theta^{t+1})) \tag{59}$$

If we consider iterations for $t \in [0, T]$, then we have:

$$\min_{0\leq t\leq T} \sum_{k=1}^{\mathcal{K}} w_k^2 ||\mathrm{g}_k^t||^2 \leq \frac{2}{\eta T} \sum_{t=0}^{T-1} (\mathcal{L}(\Theta^t) - \mathcal{L}(\Theta^{t+1})) \tag{60}$$

$$= \frac{2}{\eta T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^T)) \tag{61}$$

$$\leq \frac{2}{\eta T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^*)) \tag{62}$$

where Θ^* represents the converged parameters. Our approach maintains a convergence rate of $O(1/T)$ for task-specific parameters $\{\Theta_i\}_{i=1}^{\mathcal{K}}$.

□

A.2.4 Convergence of Phase 2

In the subsequent convergence analysis, we omit the input $\mathcal{X}^t$ for clarity.

Theorem 4 (Convergence of Phase 2)**.** *Assume losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ are convex and differentiable and the gradient of $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ is Lipschitz continuous with constant $H > 0$, i.e. $||\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}_i(y)|| \leq H||x-y||$ for $i = 1, 2, ..., \mathcal{K}$ Then, phase 2 of connection strength optimization with step size $\eta \leq \frac{1}{Hw_i}$ for all $i = 1, 2, ..., \mathcal{K}$ will converge to the Pareto-optimal point.*

Proof. We start from quadratic expansion of task-specific loss of task τ_i for $\theta_{s,j}$.

$$\mathcal{L}_i(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_i^t) \leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) + \nabla\mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t)(\theta_{s,i}^{t+1} - \theta_{s,i}^t) + \frac{1}{2}\nabla^2\mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t)(\theta_{s,i}^{t+1} - \theta_{s,i}^t)^2 \tag{63}$$

$$\leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) + \nabla\mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t)(\theta_{s,i}^{t+1} - \theta_{s,i}^t) + \frac{1}{2}H(\theta_{s,i}^{t+1} - \theta_{s,i}^t)^2 \tag{64}$$

The inequality in Eq. (64) holds as $\nabla\mathcal{L}$ is Lipschitz continuous with constant H. It implies that $\nabla^2\mathcal{L} - HI \leq 0$.
Let g_k^t be the gradient of $\theta_{s,j}^t$ for task τ_k as follows:

$$\mathrm{g}_k^t = \nabla_{\theta_{s,j}^t}\mathcal{L}_k(\tilde{\Theta}_{s,j}^t, \theta_{s,i}^t, \Theta_k^t) \tag{65}$$

The gradient update rule for Phase 1 in connection strength-based optimization is as follows:

$$\theta_{s,i}^{t+1} = \begin{cases} \theta_{s,i}^t - \eta w_i(\mathrm{g}_i^t), & \text{if } i = j. \\ \theta_{s,i}^t - \eta w_j(\mathrm{g}_j^t - \frac{\mathrm{g}_i^t \cdot \mathrm{g}_j^t}{||\mathrm{g}_i^t||^2}\mathrm{g}_i^t), & \text{otherwise.} \end{cases} \tag{66}$$

(i) When $i = j$, if we substitute Eq. (66) into Eq. (64), it becomes as follows.

$$\mathcal{L}_i(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_i^t) \leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \eta w_i||\mathrm{g}_i^t||^2 + \frac{\eta^2 w_i^2}{2}H||\mathrm{g}_i^t||^2 \tag{67}$$

$$= \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \eta w_i||\mathrm{g}_i^t||^2(1 - \frac{1}{2}\eta w_i H) \tag{68}$$

Assuming that the step size η is sufficiently small, such that $\eta \leq \frac{1}{Hw_i}$. Thus the following inequality holds:

$$\mathcal{L}_i(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_i^t) \leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \frac{1}{2}\eta w_i||\mathrm{g}_i^t||^2 \tag{69}$$

(ii) When $i \neq j$, if we substitute Eq. (66) into Eq. (64) similarly, it becomes as follows.

$$\mathcal{L}_i(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_i^t) \leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \eta w_j \mathrm{g}_j^t(\mathrm{g}_j^t - \frac{\mathrm{g}_i^t \cdot \mathrm{g}_j^t}{||\mathrm{g}_i^t||^2}\mathrm{g}_i^t) + \frac{\eta^2 w_j^2}{2}H||(\mathrm{g}_j^t - \frac{\mathrm{g}_i^t \cdot \mathrm{g}_j^t}{||\mathrm{g}_i^t||^2}\mathrm{g}_i^t)||^2 \tag{70}$$

$$= \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \eta w_j(||\mathrm{g}_j^t||^2 - \frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2})) + \frac{\eta^2 w_j^2}{2}H(||\mathrm{g}_j^t||^2 - 2\frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2} + \frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2}) \tag{71}$$

$$= \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \eta w_j(1 - \frac{1}{2}\eta w_j H)(||\mathrm{g}_j^t||^2 - \frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2})) \tag{72}$$

Given that the step size η satisfies $\eta \leq \frac{1}{Hw_j}$, the following inequality holds.

$$\mathcal{L}_i(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_i^t) \leq \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \frac{1}{2}\eta w_j(||\mathrm{g}_j||^2 - \frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2})) \tag{73}$$

$$= \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \frac{1}{2}\eta w_j||\mathrm{g}_j^t||^2(1 - \frac{(\mathrm{g}_i^t \cdot \mathrm{g}_j^t)^2}{||\mathrm{g}_i^t||^2||\mathrm{g}_j^t||^2})) \tag{74}$$

$$= \mathcal{L}_i(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_i^t) - \frac{1}{2}\eta w_j||\mathrm{g}_j^t||^2(1 - cos^2\phi_{ij}^t) \tag{75}$$

where ϕ_{ij}^t is the angle between g_i^t and g_j^t. When we sum Eq. (69) and Eq. (75) over all task losses $\{\mathcal{L}_k\}_{k=1}^{\mathcal{K}}$ along with their corresponding task-specific weights $\{w_k\}_{k=1}^{\mathcal{K}}$, the following result is obtained:

$$\sum_{k=1}^{\mathcal{K}} w_k\mathcal{L}_k(\theta_{s,j}^{t+1}, \tilde{\Theta}_{s,j}^t, \Theta_k^t) \leq \sum_{k=1}^{\mathcal{K}} w_k\mathcal{L}_k(\theta_{s,j}^t, \tilde{\Theta}_{s,j}^t, \Theta_k^t) - \frac{1}{2}\eta(w_j^2||\mathrm{g}_j^t||^2 + \sum_{\substack{k=1 \\ k\neq j}}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2(1 - cos^2\phi_{jk}^t)) \tag{76}$$

We can follow a similar process for all shared parameters $\Theta_s = \{\theta_{s,1}, \theta_{s,2}, ..., \theta_{s,\mathcal{K}}\}$. The second term on the right side of Eq. (76) is not smaller than zero, proving their convergence. This term can be zero only when $\mathrm{g}_k^t = 0$ for all $k = 1, 2, ..., \mathcal{K}$. Thus, we can conclude that the application of Phase 2 in connection strength-based optimization can lead to a Pareto-optimal state, as all task-specific gradients converge to zero in the optimization process. Understanding the task priority of each parameter enables the expansion of the known Pareto frontier which is consistent with the results of Theorem 1. Repeatedly applying Phase 2 of connection strength-based optimization ultimately leads to Pareto optimality. □

A.2.5 Convergence rate of Phase 2

Theorem 5 (Convergence rate of Phase 2)**.** *Assume losses $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ are differentiable and the gradient of $\{\mathcal{L}_i\}_{i=1}^{\mathcal{K}}$ is Lipschitz continuous with constant $H > 0$, i.e. $||\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}_i(y)|| \leq H||x - y||$ for $i = 1, 2, ..., \mathcal{K}$ Then, phase 2 of connection strength optimization with step size $\eta \leq \frac{1}{H}$, the system will reach the Pareto optimal point at a rate of $O(1/T)$, where T is the total number of iterations. This is guaranteed by the following inequality:*

$$\min_{0\leq t\leq T} \sum_{k=1}^{\mathcal{K}} w_k^2 ||\mathrm{g}_k^t||^2 \leq \frac{2}{\eta(1-\alpha^2)T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^*)) \tag{77}$$

where Θ^* represents the converged parameters, α is a constant satisfying $\alpha > -1$, and T is the total number of iterations.

Proof. We start with the outcome (Eq. (76)) derived in Theorem 4. For simplicity, consider the following notation.

$$\mathcal{L}(\Theta^t) = \sum_{k=1}^{\mathcal{K}} w_k \mathcal{L}_k(\Theta^t), \qquad \mathrm{g}^t = \sum_{j=1}^{\mathcal{K}} w_j \nabla\mathcal{L}_j(\Theta^t) \tag{78}$$

And each update iteration t is indicated as a superscript for the gradients. Therefore, Eq. (76) can be expressed as follows:

$$\mathcal{L}(\Theta^{t+1}) \leq \mathcal{L}(\Theta^t) - \frac{1}{2}\eta(w_j^2||\mathrm{g}_j^t||^2 + \sum_{\substack{k=1\\k\neq j}}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2(1 - cos^2\phi_{jk}^t)) \tag{79}$$

The term $(1 - \cos^2\phi_{jk}^t) \leq 1$ holds for all $k = 1, 2, ..., \mathcal{K}$.
Let c represent the task number that minimizes the term $1 - \cos^2\phi_{jk}^t$ excluding j.

$$c = \arg\min_{\substack{k\\k\neq j}} (1 - cos^2\phi_{jk}^t)) \tag{80}$$

By employing Eq. (80) in Eq. (79), the following inequality holds:

$$\mathcal{L}(\Theta^{t+1}) \leq \mathcal{L}(\Theta^t) - \frac{1}{2}\eta(w_j^2||\mathrm{g}_j^t||^2 + \sum_{\substack{k=1\\k\neq j}}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2(1 - cos^2\phi_{jc}^t)) \tag{81}$$

$$\leq \mathcal{L}(\Theta^t) - \frac{1}{2}\eta(w_j^2||\mathrm{g}_j^t||^2(1 - cos^2\phi_{jc}^t) + \sum_{\substack{k=1\\k\neq j}}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2(1 - cos^2\phi_{jc}^t)) \tag{82}$$

$$= \mathcal{L}(\Theta^t) - \frac{1}{2}\eta(1 - cos^2\phi_{jc}^t) \sum_{k=1}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2 \tag{83}$$

By rearranging the term in Eq. (83):

$$\sum_{k=1}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2 \leq \frac{2}{\eta(1 - cos^2\phi_{jc}^t)}(\mathcal{L}(\Theta^t) - \mathcal{L}(\Theta^{t+1})) \tag{84}$$

If we consider iterations for $t \in [0, T]$ and let α satisfy $cos\phi_{jc}^t \geq \alpha > -1$, then we have:

$$\min_{0\leq t\leq T} \sum_{k=1}^{\mathcal{K}} w_k^2||\mathrm{g}_k^t||^2 \leq \frac{2}{\eta(1-\alpha^2)T} \sum_{t=0}^{T-1}(\mathcal{L}(\Theta^t) - \mathcal{L}(\Theta^{t+1})) \tag{85}$$

$$= \frac{2}{\eta(1-\alpha^2)T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^T)) \tag{86}$$

$$\leq \frac{2}{\eta(1-\alpha^2)T}(\mathcal{L}(\Theta^0) - \mathcal{L}(\Theta^*)) \tag{87}$$

where Θ^* represents the converged parameters. Our approach maintains a convergence rate of $O(1/T)$.

□

B. Loss scaling methods

In this paper, we used 4 different loss scaling methods to weigh multiple tasks' losses.

1. All tasks' losses are weighted equally.
2. The weights of tasks are tuned manually following the previous works [40, 43]. For NYUD-v2, the weight of losses is as follows:

   Depth : SemSeg : Surface Normal : Edge = 1.0 : 1.0 : 10.0 : 50.0

   For PASCAL-Context, the weight of losses is as follows:

   Semseg : PartSeg : Saliency : Surface Normal : Edge = 1.0 : 2.0 : 5.0 : 10.0 : 50.0
3. The losses are dynamically weighted by homoscedastic uncertainty [22].
   An uncertainty that cannot be reduced with increasing data is called Aleatoric uncertainty. Homoscedastic uncertainty is a kind of Aleatoric uncertainty that stays constant for all input data and varies between different tasks. So it is also called task-dependent uncertainty. Homoscedastic uncertainty is formulated differently depending on whether the task is a regression task or a classification task as each of them uses different output functions: A regression task uses Gaussian Likelihood, in contrast, a classification task uses softmax function. The objectives of uncertainty weighting are as follows:

$$\mathcal{L}_{Total} = \sum_{i=1}^{\mathcal{K}} \hat{\mathcal{L}}_i \qquad where \qquad \hat{\mathcal{L}}_i = \left\{ \begin{array}{ll} \frac{1}{2\sigma_1^2}\mathcal{L}_i + \log \sigma_i & \text{for regression task} \\ \frac{1}{\sigma_2^2}\mathcal{L}_i + \log \sigma_i & \text{for classification task} \end{array} \right\} \tag{88}$$

4. The losses are dynamically weighted by descending rate of loss [29] which is called Dynamic Weight Average (DWA). The weight of task w_i is defined as follows with DWA:

$$w_i(t) = \frac{\mathcal{K}\exp(w_i(t-1)/T)}{\sum_{i=1}^{\mathcal{K}} exp(w_i(t-1)/T)} \qquad where \quad w_i(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)} \tag{89}$$

   where t is an iteration index and $\mathcal{K}$ is the number of tasks. T represents the temperature parameter governing the softness of task weighting. As T increases, the tasks become likely to be weighted equally. We used $T = 2$ for our experiments following the works in [29].

C. Experimental Details

Implementation details. To train MTI-Net [40] on both NYUD-v2 and PASCAL-Context, we adopted the loss schema and augmentation strategy from PAD-Net[43] and MTI-Net[40]. For depth estimation, we utilized L1 loss, while the cross-entropy loss was used for semantic segmentation. To train for saliency estimation and edge detection, we employed the well-known balanced cross-entropy loss. Surface normal prediction used L1 loss. We augmented input images by randomly scaling them with a ratio from 1, 1.2, 1.5 and horizontally flipping them with a 50% probability. The network was trained for 200 epochs for NYUD-v2 and 50 epochs for PASCAL-Context using the Adam optimizer. We employed a learning rate of 10^{-4} with a poly learning rate decay policy. We used a weight decay of 10^{-4} and batch size of 8.
In contrast, for Cityscapes with SegNet [1], we followed the experimental setting in [13, 26]. We used L1 loss and cross-entropy loss for depth estimation and semantic segmentation, respectively. The network was trained for 200 epochs using the Adam optimizer. We employed a learning rate of 5×10^{-5} with multi-step learning rate scheduling. We used a batch size of 8.
Evaluation metric. To evaluate the performance of tasks, we employed widely used metrics. For semantic segmentation, we utilized mean Intersection over Union (mIoU), Pixel Accuracy (PAcc), and mean Accuracy (mAcc). Surface normal prediction's performance was measured by calculating the mean and median angle distances between the predicted output and ground truth. We also used the proportion of pixels within the angles of $11.25°$, $22.5°$, $30°$ to the ground truth, as suggested by [10]. To evaluate the depth estimation task, we followed the methods proposed in [11, 27, 42]. We used Root Mean Squared Error (RMSE), and Mean Relative Error (abs_rel). For saliency estimation and human part segmentation, we employed mean Intersection over Union (mIoU).

D. Additional Experimental Results

We compare GD, MGDA [36], PCGrad [45], CAGrad [26], Aligned-MTL [37], and connection strength-based optimization on 4 different multi-task loss scaling methods mentioned in Appendix B. We have summarized the experimental overview as follows.

1. NYUD-v2 with HRNet-18 on various loss scaling is evaluated in Tabs. 5 to 7.
2. NYUD-v2 with ResNet-18 on various loss scaling is evaluated in Tabs. 8 to 11.
3. PASCAL-Context with HRNet-18 on various loss scaling is evaluated in Tabs. 12 to 14.

D.1. NYUD-v2 with HRNet-18

Table 5. The experimental results of different multi-task optimization methods on NYUD-v2 with HRNet-18. The losses of all tasks are evenly weighted. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.667	0.186	33.18	65.04	45.07	20.75	14.04	41.32	68.26	78.04	+ 0.00
GD	0.595	0.150	40.67	70.11	53.41	21.45	15.02	39.06	66.42	76.87	+ 10.00
MGDA [36]	0.587	0.148	40.69	70.40	53.15	21.30	14.73	39.59	66.85	77.12	+ 10.66
PCGrad [45]	0.581	0.155	40.33	70.44	52.83	21.23	14.59	40.01	67.17	77.31	+ 10.71
CAGrad [26]	**0.576**	0.149	40.00	70.45	51.75	21.09	14.50	40.18	67.40	77.47	+ 10.85
Aligned-MTL [37]	0.588	0.152	40.58	70.37	52.71	21.17	14.55	40.07	67.23	77.39	+ 10.71
Ours	**0.576**	**0.143**	**41.20**	**71.03**	**53.76**	**20.42**	**13.75**	**42.20**	**69.22**	**78.88**	**+ 13.13**

Table 6. The experimental results of different multi-task optimization methods on NYUD-v2 with HRNet-18. The losses are weighted using Dynamic Weight Average (DWA). Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.667	0.186	33.18	65.04	45.07	20.75	14.04	41.32	68.26	78.04	+ 0.00
GD	0.592	0.146	40.86	70.19	53.01	21.15	14.52	40.20	67.36	77.48	+ 10.82
MGDA [36]	0.593	0.147	40.46	70.10	52.83	21.30	14.68	39.73	66.90	77.16	+ 10.13
PCGrad [45]	0.593	0.147	40.34	70.00	52.37	21.36	14.77	39.57	66.78	77.07	+ 9.91
CAGrad [26]	0.576	0.146	40.52	70.23	52.73	21.09	14.59	40.18	67.40	77.49	+ 11.38
Aligned-MTL [37]	0.590	0.147	40.43	70.09	52.66	21.18	14.61	39.98	67.21	77.39	+ 10.44
Ours	**0.565**	**0.141**	**41.64**	**70.97**	**54.49**	**20.35**	**13.48**	**43.04**	**69.60**	**78.95**	**+ 14.24**

Table 7. The experimental results of different multi-task optimization methods on NYUD-v2 with HRNet-18. The losses are weighted by homoscedastic uncertainty. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.667	0.186	33.18	65.04	45.07	20.75	14.04	41.32	68.26	78.04	+ 0.00
GD	0.589	0.148	39.93	70.15	51.99	21.13	14.46	40.47	67.28	77.38	+ 9.87
MGDA [36]	0.590	0.148	39.78	69.77	51.80	21.24	14.69	39.78	66.94	77.22	+ 9.69
PCGrad [45]	0.587	0.147	40.56	69.97	53.07	21.19	14.40	40.51	67.46	77.41	+ 10.71
CAGrad [26]	0.583	0.147	40.23	70.06	52.74	21.09	14.47	40.23	67.48	77.50	+ 10.73
Aligned-MTL [37]	0.589	0.147	40.08	69.91	52.23	21.15	14.47	10.19	67.45	77.45	+ 10.17
Ours	**0.569**	**0.140**	**41.16**	**70.83**	**53.65**	**20.19**	**13.39**	**43.33**	**70.07**	**79.30**	**+ 13.81**

D.2. NYUD-v2 with ResNet-18

Table 8. The experimental results of different multi-task optimization methods on NYUD-v2 with ResNet-18. The losses of all tasks are evenly weighted. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.659	0.183	34.46	65.51	46.50	23.36	16.67	34.89	62.19	73.33	+ 0.00
GD	0.613	**0.160**	38.54	68.89	51.04	22.09	15.35	38.29	65.12	75.61	+ 8.09
MGDA [36]	0.616	0.165	**39.49**	69.30	**52.30**	22.52	15.61	37.92	64.25	74.77	+ 8.24
PCGrad [45]	0.618	0.164	38.76	69.01	51.12	22.05	15.28	38.55	65.36	75.77	+ 8.10
CAGrad [26]	0.610	**0.160**	39.20	**69.38**	51.58	22.18	15.61	37.65	64.70	75.42	+ 8.75
Aligned-MTL [37]	0.612	0.161	39.35	69.21	51.80	22.34	15.47	38.12	64.83	75.61	+ 8.56
Ours	**0.601**	0.162	38.30	68.78	51.01	**21.09**	**14.31**	**40.95**	**67.57**	**77.50**	**+ 9.89**

Table 9. The experimental results of different multi-task optimization methods on NYUD-v2 with ResNet-18. The weights of tasks are manually tuned. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.659	0.183	34.46	65.51	46.50	23.36	16.67	34.89	62.19	73.33	+ 0.00
GD	0.622	0.163	38.07	68.31	50.84	21.49	14.63	40.04	66.87	76.87	+ 8.03
MGDA [36]	0.635	0.166	38.18	68.22	49.70	22.07	15.01	39.11	65.81	75.90	+ 6.65
PCGrad [45]	0.617	0.165	37.80	67.94	50.00	21.52	14.53	40.27	66.91	76.71	+ 7.98
CAGrad [26]	0.620	0.163	37.02	67.96	49.71	21.67	14.80	39.55	66.46	76.56	+ 6.86
Aligned-MTL [37]	0.625	0.166	38.01	68.12	50.43	21.62	14.75	39.62	66.58	76.68	+ 7.64
Ours	**0.600**	**0.157**	**39.00**	**69.02**	**51.11**	**20.65**	**13.77**	**42.78**	**68.97**	**78.30**	**+ 11.24**

Table 10. The experimental results of different multi-task optimization methods on NYUD-v2 with ResNet-18. The losses are weighted using Dynamic Weight Average (DWA). Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			MTP
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.659	0.183	34.46	65.51	46.50	23.36	16.67	34.89	62.19	73.33	+ 0.00
GD	0.607	0.159	38.65	68.99	51.72	22.17	15.52	38.51	65.11	75.47	+ 8.38
MGDA [36]	0.616	0.165	39.38	69.18	51.78	22.53	15.69	37.68	64.12	74.67	+ 8.12
PCGrad [45]	0.612	0.162	38.56	68.97	51.16	22.11	15.40	38.20	65.07	75.58	+ 8.13
CAGrad [26]	0.609	0.157	**39.40**	**69.30**	**51.84**	22.28	15.68	37.62	64.46	75.24	+ 8.85
Aligned-MTL [37]	0.609	0.161	39.22	69.04	69.01	22.15	15.48	38.30	65.08	75.52	+ 8.86
Ours	**0.592**	**0.148**	38.41	68.82	51.15	**20.96**	**14.25**	**40.97**	**67.59**	**77.10**	**+ 10.63**

Table 11. The experimental results of different multi-task optimization methods on NYUD-v2 with ResNet-18. The losses are weighted by homoscedastic uncertainty. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	Depth		SemSeg			Surface Normal					
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			MTP
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.659	0.183	34.46	65.51	46.50	23.36	16.67	34.89	62.19	73.33	+ 0.00
GD	0.608	0.158	39.02	69.29	51.48	22.06	15.47	37.98	65.01	75.68	+ 8.85
MGDA [36]	0.623	0.162	**39.43**	**69.30**	**51.79**	22.65	15.77	37.39	64.03	74.66	+ 7.64
PCGrad [45]	0.606	0.158	39.40	69.25	51.68	22.25	15.43	38.05	64.81	75.35	+ 9.04
CAGrad [26]	0.600	0.156	38.62	68.74	51.03	22.27	15.43	38.11	64.85	75.32	+ 8.56
Aligned-MTL [37]	0.605	0.158	39.10	69.23	51.56	22.13	15.49	37.77	64.89	75.51	+ 8.97
Ours	**0.595**	**0.153**	38.67	69.01	51.01	**21.05**	**14.11**	**41.43**	**67.91**	**77.59**	**+ 10.61**

D.3. PASCAL-Context with HRNet-18

Table 12. The experimental results of different multi-task optimization methods on PASCAL-Context dataset with HRNet-18. The losses of all tasks are evenly weighted. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	SemSeg		PartSeg	Saliency		Surface Normal					
Method	(Higher Better)		(Higher Better)	(Higher Better)		Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			MTP
	mIoU	PAcc	mIoU	mIoU	maxF	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	60.30	89.88	60.56	67.05	78.98	14.76	11.92	47.61	81.02	90.65	+ 0.00
GD	61.65	90.14	58.35	65.80	78.07	16.71	13.82	39.70	75.18	87.17	- 4.12
MGDA [36]	**63.52**	**90.68**	60.38	64.99	77.57	17.00	14.13	38.58	74.47	86.77	- 3.30
PCGrad [45]	63.21	90.33	60.42	64.77	77.48	16.65	13.71	39.64	75.10	87.07	- 2.90
CAGrad [26]	63.44	90.53	60.11	64.83	77.52	16.92	13.98	39.03	75.01	86.92	- 3.37
Aligned-MTL [37]	62.38	90.31	60.36	65.68	79.92	16.73	13.88	39.68	75.18	87.10	- 3.07
Ours	62.64	90.39	**61.42**	**67.10**	**78.91**	**15.58**	**12.68**	**43.93**	**78.69**	**89.26**	**- 0.05**

Table 13. The experimental results of different multi-task optimization methods on PASCAL-Context dataset with HRNet-18. The losses are weighted using Dynamic Weight Average (DWA). Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	SemSeg		PartSeg	Saliency		Surface Normal					
Method	(Higher Better)		(Higher Better)	(Higher Better)		Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			MTP
	mIoU	PAcc	mIoU	mIoU	maxF	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	60.30	89.88	60.56	67.05	78.98	14.76	11.92	47.61	81.02	90.65	+ 0.00
GD	**64.70**	**91.18**	60.60	66.54	78.18	15.13	12.23	45.77	79.91	89.96	+ 1.02
MGDA [36]	64.56	90.72	60.69	65.93	77.37	16.87	13.95	39.35	74.69	86.82	- 2.17
PCGrad [45]	64.35	90.98	60.99	66.12	77.65	15.92	13.11	41.98	76.21	88.03	- 0.45
CAGrad [26]	64.03	90.77	60.62	66.01	77.42	16.63	13.86	40.02	75.22	87.41	- 1,98
Aligned-MTL [37]	64.41	91.00	60.77	66.09	77.51	16.22	13.48	42.26	76.92	88.66	- 1.04
Ours	63.89	90.73	**61.89**	**67.39**	**79.08**	**14.94**	**12.10**	**46.27**	**80.57**	**90.41**	**+ 1.86**

Table 14. The experimental results of different multi-task optimization methods on PASCAL-Context dataset with HRNet-18. The losses are weighted by homoscedastic uncertainty. Experiments are repeated over 3 random seeds and average values are presented. $\triangle_m \uparrow(\%)$ is used to indicate the percentage improvement in multi-task performance (MTP). The best results are expressed in **bold** numbers.

Tasks	SemSeg		PartSeg	Saliency		Surface Normal					
Method	(Higher Better)		(Higher Better)	(Higher Better)		Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			MTP
	mIoU	PAcc	mIoU	mIoU	maxF	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	60.30	89.88	60.56	67.05	78.98	14.76	11.92	47.61	81.02	90.65	+ 0.00
GD	64.40	91.05	62.28	68.13	79.64.	14.95	12.14	46.19	80.36	90.34	+ 2.49
MGDA [36]	64.04	90.88	61.18	67.65	79.23	15.02	12.20	45.93	80.02	90.11	+ 1.59
PCGrad [45]	**64.75**	**91.11**	**62.41**	68.16	79.65	14.86	11.93	47.03	80.60	90.31	+ 2.85
CAGrad [26]	64.01	90.77	61.32	67.55	79.01	15.08	12.31	45.87	79.98	90.05	+ 1.50
Aligned-MTL [37]	64.48	91.09	62.23	67.61	79.18	15.01	12.11	46.01	80.17	90.20	+ 2.21
Ours	64.01	90.70	61.78	**68.32**	**81.50**	**14.53**	**11.52**	**48.21**	**81.88**	**90.74**	**+ 2.90**

E. Additional Ablation Studies

The order of updating tasks in Phase 1 has little impact on multi-task performance. To learn task priority in shared parameters, Phase 1 updates each task-specific gradient one by one sequentially. To determine the influence of the order of tasks on optimization, we randomly chose 5 sequences of tasks and showed their performance in Tab. 15. From the results, we can see that the order of updating tasks in Phase 1 does not have a significant impact on multi-task performance.

Table 15. The experimental results for NYUD-v2 with HRNet-18 involved exploring different task sequence orders in Phase 1. We conducted ablation experiments with five randomly selected task sequences. Each task was represented by a single alphabet letter, as follows: S for semantic segmentation, D for depth estimation, E for edge detection, and N for surface normal estimation.

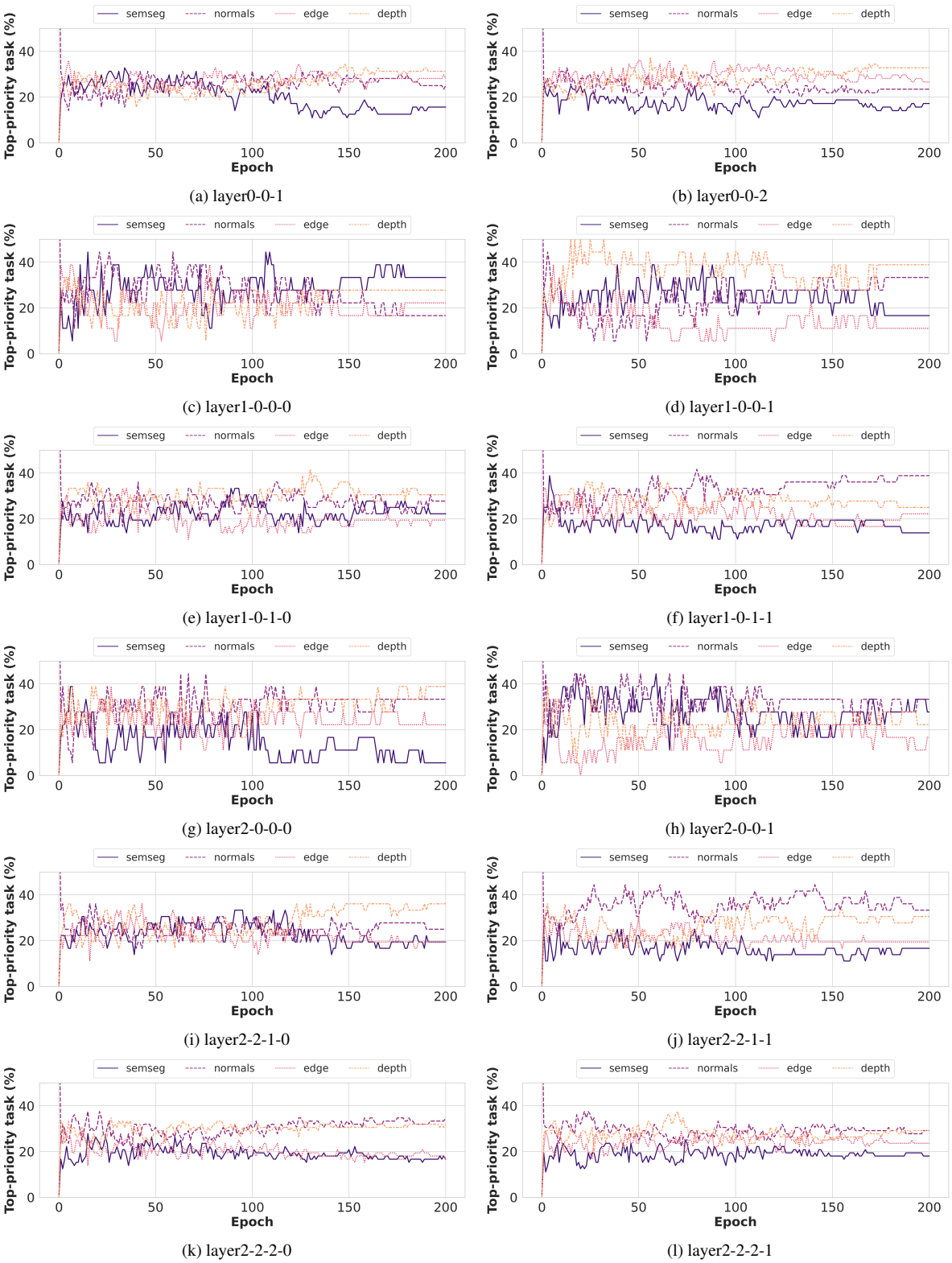
Tasks	Depth		SemSeg			Surface Normal					MTP
Method	Distance (Lower Better)		(%) (Higher Better)			Angle Distance (Lower Better)		Within t degree (%) (Higher Better)			
	rmse	abs_rel	mIoU	PAcc	mAcc	mean	median	11.25	22.5	30	$\triangle_m \uparrow(\%)$
Independent	0.667	0.186	33.18	65.04	45.07	20.75	14.04	41.32	68.26	78.04	+ 0.00
N-D-S-E	0.574	0.157	41.12	70.44	53.77	19.60	12.52	46.01	71.33	80.02	+ 14.47
D-S-N-E	0.568	0.153	40.92	70.23	53.56	19.55	12.47	46.09	71.50	80.12	+ 14.65
E-D-S-N	0.568	0.150	40.97	70.22	53.59	19.58	12.50	46.08	71.44	80.07	+ 14.65
D-N-E-S	0.571	0.153	41.03	70.31	53.68	19.49	12.44	46.17	71.58	80.17	+ 14.71
S-D-E-N	0.565	0.148	41.10	70.37	53.74	19.54	12.45	46.11	71.54	80.12	+ 15.00

Our method demands the least computational load when compared to previous optimization methods. In Tab. 16, we show the impact of the proposed optimization on training time. The training time for each method is measured in seconds per epoch. To ensure a fair comparison, all methods were evaluated using the same architecture, guaranteeing an equal number of parameters and memory usage. The majority of the computational burden is concentrated on the forward pass, backpropagation, and gradient manipulation. While all optimization methods follow a similar process in the forward pass and backpropagation, the primary distinction arises from gradient manipulation. In Phase 1, no gradient manipulation is required, resulting in the shortest time consumption. In phase 2, it still exhibits the shortest training time compared to previous optimization methods. Unlike these previous methods that handle all shared components of the network, Phase 2 specifically targets the shared convolutional layer along with the task-specific batch normalization layer. This selective focus significantly reduces the time consumed per epoch.

Table 16. Training time comparison for different multi-task optimization methods on NYUD-v2 with HRNet18.

Method	MGDA[36]	PCGrad[45]	CAGrad[26]	Aligned-MTL [37]	Phase 1	Phase 2
Time (s)	363.98	421.48	378.12	811.57	296.74	331.53

The speed of learning the task priority differs based on the convolutional layer's position. Phase 1 establishes the task priority during the initial stages of the network's optimization. Meanwhile, Phase 2 maintains this learned task priority, ensuring robust learning even when the loss for each task fluctuates. However, The timing at which task priority stabilizes varies based on the position of the convolutional layer within the network, as illustrated in Fig. 5. This may suggest that optimizing by wholly separating each phase could be inefficient.



(a) layer0-0-1

(b) layer0-0-2

(c) layer1-0-0-0

(d) layer1-0-0-1

(e) layer1-0-1-0

(f) layer1-0-1-1

(g) layer2-0-0-0

(h) layer2-0-0-1

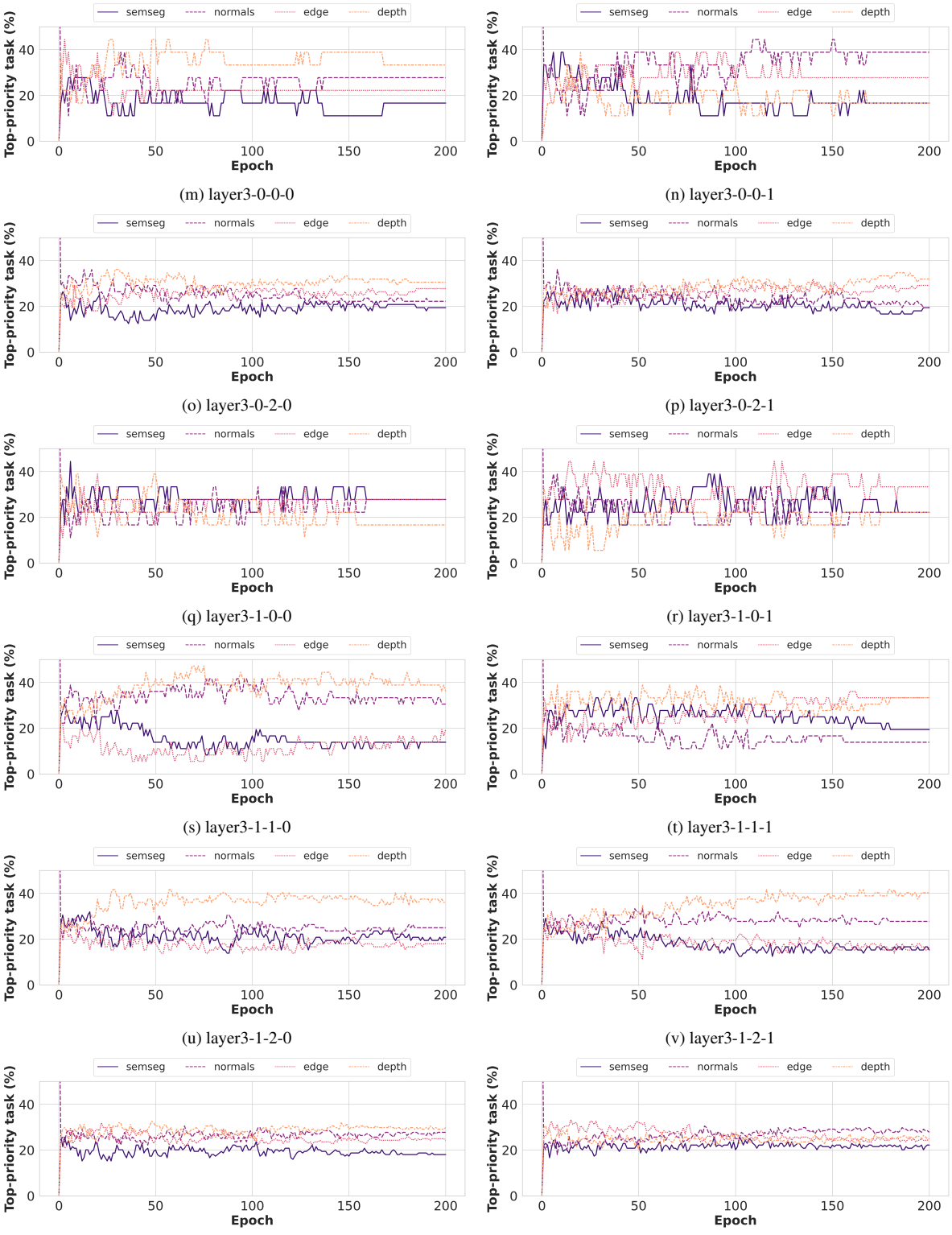
(i) layer2-2-1-0

(j) layer2-2-1-1

(k) layer2-2-2-0

(l) layer2-2-2-1

Figure 5. Visualization of the percentage of top-priority tasks over training epoch depending on the position in the network. We randomly selected several convolutional layers from the Network. The timing at which task priority stabilizes varies depending on the position of the convolutional layer.



(m) layer3-0-0-0 (n) layer3-0-0-1

(o) layer3-0-2-0 (p) layer3-0-2-1

(q) layer3-1-0-0 (r) layer3-1-0-1

(s) layer3-1-1-0 (t) layer3-1-1-1

(u) layer3-1-2-0 (v) layer3-1-2-1

(w) layer3-1-3-0 (x) layer3-1-3-1

Figure 5. Visualization of the percentage of top-priority tasks over training epoch depending on the position in the network. We randomly selected several convolutional layers from the Network. The timing at which task priority stabilizes varies depending on the position of the convolutional layer.