Soft Conflict-Resolution Decision Transformer for Offline Multi-Task Reinforcement Learning

Shudong Wang[1 2 3*], **Xinfei Wang**[1 2 3*], **Chenhao Zhang**[1 2 3*†], **Shanchen Pang**[1 2 3], **Haiyuan Gui**[4], **Wenhao Ji**[1 2 3], **Xiaojian Liao**[5]

[1]School of Computer Science and Technology, China University of Petroleum(East China)
[2]State Key Laboratory of Chemical Safety
[3]Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software
[4]School of Information and Control Engineering, Qingdao University of Technology
[5]State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
{wangsd@, z24070040@s., zch@, pangsc@}upc.edu.cn,

Abstract

Multi-task reinforcement learning (MTRL) seeks to learn a unified policy for diverse tasks, but often suffers from gradient conflicts across tasks. Existing masking-based methods attempt to mitigate such conflicts by assigning task-specific parameter masks. However, our empirical study shows that coarse-grained binary masks have the problem of over-suppressing key conflicting parameters, hindering knowledge sharing across tasks. Moreover, different tasks exhibit varying conflict levels, yet existing methods use a one-size-fits-all fixed sparsity strategy to keep training stability and performance, which proves inadequate. These limitations hinder the model's generalization and learning efficiency.

To address these issues, we propose **SoCo-DT**, a *Soft Conflict-resolution* method based by parameter importance. By leveraging Fisher information, mask values are dynamically adjusted to retain important parameters while suppressing conflicting ones. In addition, we introduce a Task-Aware Mask Update with Adaptive Sparsity strategy based on the Interquartile Range (IQR), which constructs task-specific thresholding schemes using the distribution of conflict and harmony scores during training. To enable adaptive sparsity evolution throughout training, we further incorporate an asymmetric cosine annealing schedule to continuously update the threshold. Experimental results on the Meta-World benchmark show that SoCo-DT outperforms the state-of-the-art method by 7.6% on MT50 and by 10.5% on the suboptimal dataset, demonstrating its effectiveness in mitigating gradient conflicts and improving overall multi-task performance.

1 Introduction

Offline Reinforcement Learning (Offline RL) (Levine et al. 2020) enables policy learning from static, pre-collected datasets without interacting with the environment, offering improved safety, lower data costs, and better deployability in domains such as robotics (Kumar et al. 2021), autonomous driving (Shi et al. 2021), and healthcare (Ghasemi et al. 2025). However, offline RL often lacks cross-task generalization and must relearn from scratch when faced with new tasks (Teh et al. 2017), limiting its scalability in complex environments. Multi-task Reinforcement Learning (MTRL) addresses this by jointly training on multiple tasks to improve generalization and efficiency (D'Eramo et al. 2024; Lee et al. 2022; Caruana 1997). Combining the strengths of both, **Offline MTRL** leverages task-mixed datasets to enable knowledge transfer without environment interaction. Despite its promise, offline MTRL faces substantial practical challenges, as task and data heterogeneity often induce gradient conflicts that hinder effective parameter sharing (Tang et al. 2023; Shi et al. 2023).

Existing approaches for mitigating gradient conflict in MTRL can be broadly categorized into the following four directions: ① Model Architecture Methods Based on Parameter Sharing: These methods achieves multi-task learning by sharing parameters across tasks, typically through injecting task identifiers or task embeddings into the model to achieve task differentiation (Xu et al. 2022; He et al. 2023, 2025). ② Gradient Projection and Orthogonalization Methods: This class of methods minimizes conflicts by projecting, discarding, or rescaling conflicting gradient components (Yu et al. 2020a; Chen et al. 2020; Wang et al. 2020; Chai et al. 2022). ③ Optimal Module Routing Methods: These methods construct optimal module compositions for each task via routing or task-specific output heads (He et al. 2024; Yang et al. 2020; Sun et al. 2022). ④ Optimal Parameter Subspace Methods: These methods identify optimal parameter subspaces for each task via masking (Hu et al. 2024; Zhang et al. 2024; Sun et al. 2020). Recent studies, such as HarmoDT (Hu et al. 2024), combine sequence modeling with task-specific masks to identify optimal parameter subspaces while retaining shared parameters, which significantly reduces inter-task gradient conflict. Using HarmoDT(Hu et al. 2024) and Prompt-DT(Xu et al. 2022) as examples, this paper identifies two key issues in current approaches.

First, existing masking-based methods are too coarse-grained to effectively support knowledge sharing, limiting the model's generalization and learning efficiency. For ex-

*These authors contributed equally.
†Corresponding author.

ample, while HarmoDT alleviates gradient conflicts via binary masks, it ignores parameter importance and may inadvertently suppress parameters that are both important and conflicting, which in turn decrease the model performance by approximately 4.67% (Motivation 1, §3.1).

Second, the degree of conflict among different tasks varies, and a one-size-fits-all fixed sparsity will disrupt the optimal subspace of some tasks. For example, HarmoDT applies a uniform sparsity of 20% to conflicting parameters to maintain overall training stability and performance. However, in tasks such as *basketball* and *door-lock*, where the actual conflict ratio reaches 40%–45%, this fixed sparsity fails to provide sufficient coverage. As a result, it may retain too many conflicting parameters or mistakenly suppress important ones, leading to a performance drop of up to 60% on those tasks and an average degradation of about 5.33% overall (Motivation 2, §3.2).

To address these issues, we propose a novel soft masking mechanism tailored for offline MTRL, which enables fine-grained and importance-aware conflict mitigation. Unlike traditional binary masking strategies that indiscriminately suppress all conflicting parameters, our method assigns each conflicting parameter a soft mask value proportional to its task-specific Fisher (Zhang et al. 2024). This allows important yet conflicting parameters to be retained with high weights, while less relevant ones are gradually suppressed—striking a principled balance between conflict alleviation and knowledge preservation.

Moreover, we introduce TAMU (Task-Aware Mask Update with Adaptive Sparsity), a unified masking strategy designed to improve parameter sharing efficiency and training stability. At its core, TAMU computes fine-grained conflict and harmony scores for each parameter, capturing both gradient consistency and magnitude imbalance across tasks. Built upon these scores, TAMU integrates two key mechanisms: (1) *a magnitude-sensitive harmony scoring mechanism*, which suppresses pseudo-consistent gradients with extreme magnitude deviations using a ReLU-based gating function with a tunable tolerance factor. (2) *a task-adaptive sparsity control module*, which determines masking thresholds based on the Interquartile Range (IQR) of conflict scores and dynamically adjusts them using an asymmetric cosine annealing scheduler, enabling task-sensitive sparsity evolution throughout training. Together, these mechanisms enable TAMU to adaptively evolve the mask during training, resulting in improved multi-task generalization and stable learning dynamics across diverse offline tasks.

Experimental results on Meta-World benchmark demonstrate that our method consistently outperforms existing baselines, achieving up to 10.50% improvement in average success rate while preserving parameter sharing efficiency.

In summary, our main contributions are as follows:

- We identify key limitations in conventional mask-based MTRL methods in handling gradient conflicts.
- We propose a soft mask mechanism and task-aware mask update with adaptive sparsity strategy for more reliable subspace construction.
- We validate our method on multiple task combinations within the Meta-World benchmark, demonstrating consistent outperformance over state-of-the-art methods.

2 Preliminary

2.1 Decision Transformer and Prompt-DT

Decision Transformer (DT) (Chen et al. 2021) formulates reinforcement learning as a conditional sequence modeling problem. Rather than estimating value functions or policy gradients, it directly predicts actions using a Transformer architecture (Vaswani et al. 2017). At each timestep, the model takes as input a trajectory sequence consisting of return-to-go values $R_t = \sum_{t'=t}^{T} r_{t'}$, states s_t, and actions a_t, forming the input sequence $\tau = (\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T)$, and outputs the next action via autoregressive prediction through a causally masked Transformer. The model is trained with a standard behavior cloning loss: $\mathcal{L}_{DT} = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=1}^{T} \|\pi_\theta(\tau_{1:t}) - a_t\|_2^2\right]$.

To enable generalization across tasks, Prompt-DT (Xu et al. 2022) extends DT to the multi-task setting via a prompting mechanism. For each task T_i, a task-specific prompt is constructed from a short K-step demonstration subtrajectory sampled from the target task which is prepended to the online trajectory and used as additional input to encode task identity:

$$\tau_{i,t}^{\text{input}} = (\widehat{R}_{i,1}^*, s_{i,1}^*, a_{i,1}^*, \ldots, \widehat{R}_{i,K}^*, s_{i,K}^*, a_{i,K}^*, \widehat{R}_{i,t-K+1}, s_{i,t-K+1}, a_{i,t-K+1}, \ldots, \widehat{R}_{i,t}, s_{i,t}, a_{i,t}), \quad (1)$$

this design enables Prompt-DT to adapt to new tasks in a few-shot fashion, without requiring fine-tuning.

2.2 HarmoDT

HarmoDT (Hu et al. 2024) extends Prompt-DT by introducing a task-specific mask mechanism that enables efficient and conflict-aware parameter sharing. For each task T_i, HarmoDT learns a binary mask vector $M^{T_i} \in \{0,1\}^d$, which selects a task-specific subspace from the full parameter set $\theta \in \mathbb{R}^d$. The masked parameter is defined as $\theta^{T_i} = \theta \odot M^{T_i}$, where $\odot$ denotes element-wise multiplication.

HarmoDT jointly optimizes the task-specific masks $M = \{M^{T_1}, \ldots, M^{T_N}\}$ and shared model parameters θ,aiming to maximize the expected cumulative return under fixed masks while minimizing the multi-task training loss:

$$\max_M \mathbb{E}_{T_i\sim p(T)}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^{T_i}(s_t,\, \pi(\tau_{i,t}^{\text{input}} \mid \theta^{*T_i}))\right], \quad (2)$$

$$\text{s.t. } \theta^* = \arg\min_\theta \mathbb{E}_{T_i\sim p(T)} \mathcal{L}_{DT}(\theta, M), \quad (3)$$

$$\text{where } \theta^{*T_i} = \theta^* \odot M^{T_i}, \quad M = \{M^{T_i}\}_{T_i\sim p(T)}. \quad (4)$$

To identify the optimal harmony subspace for each task, HarmoDT computes a harmony score for each parameter by combining the agreement score $A(T_i) = \bar{g}_i \odot \frac{1}{N}\sum_{i=1}^{N} \bar{g}_i$, which reflects gradient alignment, and the importance score $F(T_i) = \left(\nabla \log \mathcal{L}_{T_i}\left(\theta^{T_i}\right) \odot M^{T_i}\right)^2$, derived from the Fisher information. The combined harmony score is defined

as $H(T_i) = A(T_i) + \lambda F(T_i)$, where λ balances agreement and importance. HarmoDT maintains an equal sparsity level for each task by removing and recovering the same number of parameters during training, thereby enabling the discovery of an optimal subspace with consistent sparsity. It achieves state-of-the-art performance on the Meta-World benchmark (Yu et al. 2020b) across environments with 5, 30, and 50 tasks (denoted as MT5, MT30, and MT50).

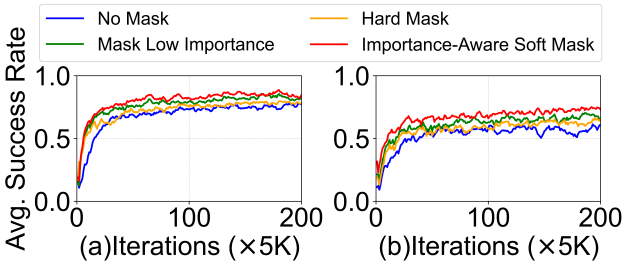


Figure 1: Average success rate of four conflict-handling strategies on MT15: ① PromptDT (no mask), ② HarmoDT (hard mask, SOTA), ③ Ours (soft mask, ideal). (a) Meta-World with near-optimal datasets; (b) Meta-World with sub-optimal datasets. See §5 for details.

3 Rethinking the Mask in MTRL

Through extensive exploration of existing mask-based works, we have two key motivations that demonstrate the vast potential for improvement in offline MTRL analysis, as shown in Fig. 1 and 2.

3.1 Parameter-Sharing

Motivation I. Coarse-grained binary masks make it difficult for tasks to share knowledge, limiting the generalization ability and learning efficiency of the model.

In MTRL, conflicting parameter updates across tasks are a key factor limiting generalization. Prompt-DT(Xu et al. 2022) and HarmoDT(Hu et al. 2024) represent the two extremes of the conflict handling strategy: Prompt-DT does not mask the parameters of conflicts, while HarmoDT employs a binary mask to completely block conflicting parameters to alleviate inter-task interference—achieving a 6.8% performance gain Fig. 1(a). However, our further observation that such hard masking approaches have increasingly evident limitations: they tend to unintentionally suppress parameters that are crucial for individual tasks. As shown in Fig. 2(a), we find that approximately 1,550 top-importance parameters (within the top 30%) are mistakenly masked due to binary conflict handling. When we manually restore these parameters, the average task success rate improves by 4.45%. On sub-optimal datasets, the gain increases to 3.92%, as illustrated by the green curve in Fig. 1. These results suggest that rigid masking sacrifices valuable knowledge and weakens parameter sharing, particularly when tasks diverge significantly or datasets contain many sub-optimal trajectories.

Such substantial improvement highlights the urgent need for the design of more balanced and resilient masking strategies in MTRL. We validate this hypothesis by adopting a soft masking mechanism (§4.1). As shown by the red curve in Fig.1, the soft mask can effectively avoid unnecessary parameter suppression, achieving a notable 6.96% performance gain. Moreover, under more challenging conditions involving sub-optimal datasets, this ideal mechanism achieves 9.33% potential performance improvement.

3.2 Multi-Task Training Stability

Motivation II. A fixed sparsity level is insufficient to accommodate the complexity and variability of diverse tasks.

As the number of tasks increases, inter-task interactions become more complex, resulting in significant variation in conflict intensity. To stabilize training, existing mask-based methods (Hu et al. 2024; Zhang et al. 2024; Sun et al. 2020) typically impose a fixed sparsity level on the mask matrix, aiming to limit the number of active parameters and prevent unstable updates. However, this static strategy proves inadequate in heterogeneous task scenarios, where conflict patterns vary considerably both across tasks and over time. Fig. 2(b) illustrates the distribution of conflicting parameters at convergence under the MT15 setting using Prompt-DT without masking. The five selected tasks differ markedly in their target distributions. The results reveal two key findings: **(1) the proportion of conflicting parameters differs significantly across tasks, and (2) the conflict ratios exhibit dynamic changes over the course of training**. Obviously, a one-size-fits-all fixed sparsity constraint struggles to accommodate task-specific conflict structures, making it difficult to discover an optimal shared parameter subspace. **This highlights the need for a strategy that balances training stability with adaptability to task-specific characteristics.**

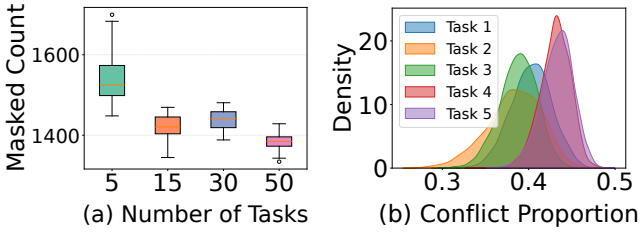


Figure 2: (a) Average number of important parameters wrongly masked during training under Fisher-based detection. (b) Distribution of conflicting parameters among tasks. Task 1-5 represent basketball, coffee-push, disassemble, door-close, and door-lock, respectively.

4 Method

Our SoCo-DT framework consists of three key components: (i) an Importance-Aware Soft Masking Mechanism (§4.1), (ii) a Task-Aware Mask Update with Adaptive Sparsity Strategy (§4.2), The overall framework and training pipeline are shown in Figure 3 and Algorithm 1.

4.1 Importance-Aware Soft Masking Mechanism

To balance conflict suppression and parameter preservation, we assign a soft importance weight to each conflicting pa-

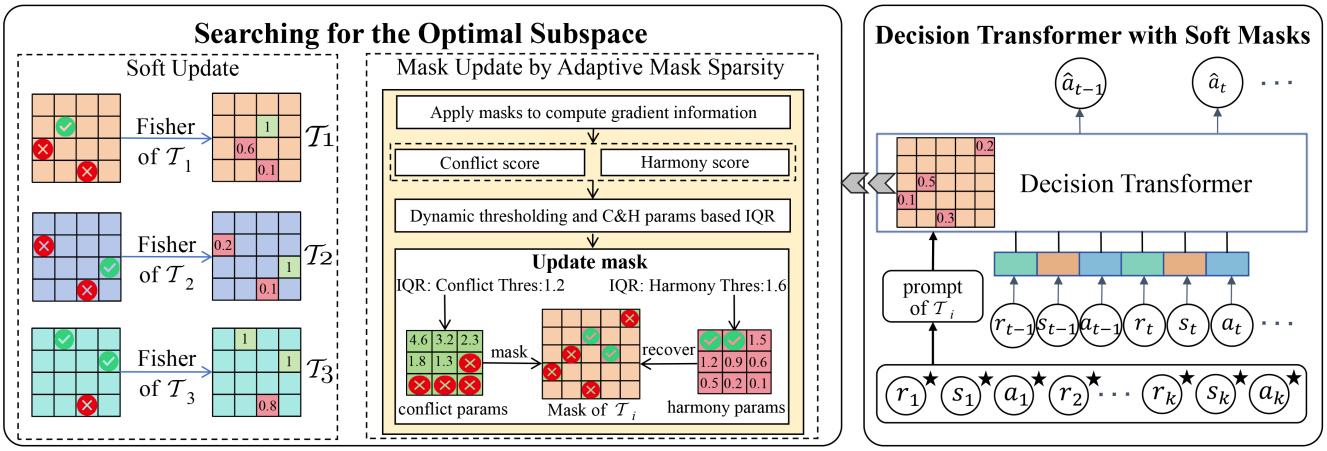


Figure 3: Illustrates the overall framework of SoCo-DT. The left panel shows the process of identifying task-optimal subspaces using the soft masking strategy and TAMU. The right panel presents the workflow of SoCo-DT based on a prompt-enhanced Decision Transformer architecture.

rameter based on its significance to the current task. Specifically, for each task T_i $(i = 1, \ldots, N)$ with loss $\mathcal{L}_{T_i}$, parameter subspace θ^{T_i}, and mask matrix M^{T_i}, the Fisher information (Hu et al. 2024) is estimated as:

$$F(T_i) = \left(\nabla \log \mathcal{L}_{T_i}(\theta^{T_i}) \odot M^{T_i}\right)^2, \tag{5}$$

where $\odot$ represents element-wise multiplication. We normalize Fisher information across all parameters of task T_i. The soft mask for parameter j is computed as:

$$M_j^{T_i} = \begin{cases} 1 & \text{if parameter is in harmony,} \\ \dfrac{F(T_i)_j - F_{\min}}{F_{\max} - F_{\min}} & \text{if parameter is conflicting,} \end{cases} \tag{6}$$

where $F_{\min} = \min_j F(T_i)_j, F_{\max} = \max_j F(T_i)_j$.

During the forward pass, we apply a binary mask to retain task-specific parameters. During the backward pass, we perform discounted gradient correction on conflicting parameters, allowing limited yet informed updates without disrupting prior tasks.

$$\theta_{t+1} = \theta_t - \eta\, \mathbb{E}\left[\nabla \mathcal{L}_{\mathcal{T}_i}(\theta \odot \widetilde{M}^{T_i})\right] \odot M^{T_i}, \tag{7}$$

where $\widetilde{M}^{T_i} = \mathbb{I}(M^{T_i} = 1)$ is the binary mask applied during the forward pass, $M^{T_i} \in [0, 1]$ is the soft mask used to control the magnitude of gradient updates for conflicting parameters, and η denotes the learning rate.

4.2 TAMU: Task-Aware Mask Update with Adaptive Sparsity

During the mask update phase, we propose TAMU, a task-aware strategy with adaptive sparsity that enhances parameter sharing while ensuring training stability. It estimates each parameter's conflict and harmony scores via a magnitude-sensitive metric, then applies task-specific thresholds to guide masking. Each task's mask is updated through a unified strategy.

Conflict Score. For each task T_i and parameter j, we compute the element-wise product between the task gradient $\mathbf{g}_i$ and the average gradient $\bar{\mathbf{g}}_j$, and incorporate the Fisher information $F(T_i)_j$ to reflect parameter importance. The conflict score is defined as:

$$C_{\text{conflict}}(T_i)_j = (\mathbf{g}_i \odot \bar{\mathbf{g}})_j + \lambda F(T_i)_j, \tag{8}$$

where λ is a trade-off coefficient that balances gradient alignment and parameter importance. A lower score indicates stronger conflict, and thus the parameter is more likely to be masked.

Harmony Score. For each task T_i, we refine the dot-product term $(\mathbf{g}_i \odot \bar{\mathbf{g}})$ using a magnitude-sensitive modulation factor $H(T_i)$, designed to downweight parameters with large magnitude deviations. Specifically, we define $H(T_i)$ as a ReLU-based gating function:

$$H(T_i) = \text{ReLU}\left(\frac{\alpha|\bar{\mathbf{g}}| - |\mathbf{g}_i|}{\alpha|\bar{\mathbf{g}}|}\right), \tag{9}$$

α is a tunable hyperparameter that controls the acceptable tolerance range. The Harmony score is then defined as:

$$C_{\text{harmony}}(T_i)_j = \begin{cases} (\mathbf{g}_i \odot \bar{\mathbf{g}})_j \cdot H(T_i)_j & (\mathbf{g}_i \odot \bar{\mathbf{g}})_j > 0, \\ (\mathbf{g}_i \odot \bar{\mathbf{g}})_j, & \text{otherwise.} \end{cases} \tag{10}$$

By suppressing parameters with large inter-task gradient magnitude discrepancies, this method prevents individual tasks from dominating the shared gradient update, even when directions are aligned (For instance, when one task contributes a gradient of 100 while the average across the others is only 0.1).

Task-Adaptive Sparsity Control Module. To adapt to task-specific conflict levels, we propose this module. Taking the selection of conflicting parameters as an example, the procedure is as follows:

First, we flatten the conflict score matrix $C_{\text{conflict}}(T_i)$ for task T_i, and sort it in ascending order to obtain a vector as $\mathbf{X}^{T_i}$. Let n be the total number of parameters. For each task T_i, we compute the quantile $Q_q^{T_i}$ as:

$$Q_q^{T_i} = \begin{cases} \mathbf{X}_{(k)}^{T_i}, & \text{if } k = qn \in \mathbb{Z} \\ (1-\gamma)\cdot \mathbf{X}_{(\lfloor k \rfloor)}^{T_i} + \gamma \cdot \mathbf{X}_{(\lceil k \rceil)}^{T_i}, & \text{if } k = qn \notin \mathbb{Z} \end{cases} \tag{11}$$

where $\gamma = qn - \lfloor qn \rfloor$, $k = qn$ and $\mathbf{X}_{(i)}$ denotes the i-th smallest element in $\mathbf{X}$. Using the quantiles, we define the interquartile range as:

$$\text{IQR}^{T_i} = Q_{q3}^{T_i} - Q_{q1}^{T_i}, \tag{12}$$

Second, a dynamic threshold is computed as:

$$\text{Threshold}_{T_i} = Q_{q1}^{T_i} - \beta_t \times \text{IQR}^{T_i}\,. \tag{13}$$

here, β_t is a dynamic coefficient that evolves during training, and q_1, q_3 are predefined hyperparameters. Parameters with scores below the threshold are considered significantly conflicting:

$$\mathcal{C}_i = \left\{ j \mid C_{\text{conflict}}(T_i)_j < \text{Threshold}_{T_i} \right\}. \tag{14}$$

To improve the stability and efficiency of conflict-aware mask updates, we first introduce a general dynamic modulation function based on *cosine annealing* (Loshchilov and Hutter 2016), defined as:

$$g(\eta_{\max}, \eta_{\min}) = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left[1 + \cos\left(2\pi \frac{t}{T} \right) \right], \tag{15}$$

where $g(\cdot)$ denotes the standard cosine annealing function, t is the current iteration step, T is the total number of iterations, and $\eta_{\max}, \eta_{\min}$ are the upper and lower bounds of the annealing schedule, respectively.

To account for the asymmetric tolerance toward conflicting parameters at different training phases, we propose a *piecewise asymmetric cosine annealing strategy* to dynamically generate the update coefficient β_t. Specifically:

$$\beta_t = \begin{cases} g(\beta_{\text{left\_max}}, \beta_{\min}, t, T) & t \le T/2, \\ g(\beta_{\text{right\_max}}, \beta_{\min}, t, T) & t > T/2, \end{cases} \tag{16}$$

where $\beta_{\text{left\_max}} > \beta_{\text{right\_max}}$ denote the maximum conflict tolerance in the early and late stages, respectively, and $\beta_{\min}$ is the minimum threshold. This design adaptively adjusts the masking sparsity throughout training. In the early stage, a larger β_t (from $\beta_{\text{left\_max}}$) prevents excessive pruning under noisy conflict estimates; as training stabilizes, a smaller β_t promotes fine-grained conflict masking. Toward the end, β_t slightly increases to ensure smooth convergence.

Finally, we apply the same TAMU to the harmony score $C_{\text{harmony}}(T_i)$ of each task T_i. Specifically, we compute $Q_{q1}^{T_i}$, $Q_{q3}^{T_i}$ and IQR^{T_i} of the harmony scores, and define a task-specific harmony threshold as: $\text{Threshold}_{T_i} = Q_{q3}^{T_i} + \beta_t \times \text{IQR}^{T_i}$. Parameters with scores exceeding this threshold are considered recoverable harmonious parameters and are collected into the set $\mathcal{R}_i$.

Mask Update. After identifying the conflicting and harmonious parameters, we proceed to update the mask matrix for each task. For each newly identified conflicting parameter $\mathcal{C}_i$, we adopt a soft masking strategy by setting its mask value to the normalized Fisher information, as follows:

$$M_{\text{mask},j}^{T_i} = \begin{cases} \frac{F(T_i)_j - F(T_i)_{\min}}{F(T_i)_{\max} - F(T_i)_{\min}} & \text{if } j \in \mathcal{C}_i. \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Meanwhile, for parameters in the recovered harmonious set $\mathcal{R}_i$, we assign their mask values in the following form to facilitate consistent and unified mask updates:

$$M_{\text{recover},j}^{T_i} = \begin{cases} 1 - M_j^{T_i} & \text{if } j \in \mathcal{R}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Finally, we perform the mask update via matrix addition and subtraction. Specifically, we replace the original mask values of conflicting parameters $\mathcal{C}_i$ with their newly computed soft values, and restore the recovered harmonious parameters $\mathcal{R}_i$ by setting their mask values to 1. The update rule is formulated as:

$$M^{T_i} = M^{T_i} - (M_{\mathcal{C}_i}^{T_i} - M_{\text{mask}}^{T_i}) + M_{\text{recover}}^{T_i}. \tag{19}$$

Algorithm 1: SoCo-DT Training Framework

Input: Number of tasks N, max training epochs E, mask update interval $interval_{mask}$, adaptive sparsity hyperparameters $\beta_{\text{left\_max}}, \beta_{\text{right\_max}}, \beta_{\min}$

Output: Task-specific optimized masks M^{T_i} and shared parameters θ

```
Initialize model parameters θ and mask matrices M
for each step t = 1 to E do
  if t mod interval_mask == 0 then
    Compute average gradient:
    ḡ = (1/N) Σ_{i=1}^{N} ∇L_{T_i}(θ ⊙ M̃^{T_i})
    for each task T_i do
      Compute task gradient: g_i = ∇L_{T_i}(θ ⊙ M̃^{T_i})
      Compute Fisher importance F(T_i) use Eq. (5)
      Compute conflict score C_mask and harmony score
      C_recover use Eq. (8) and Eq. (10)
      Update β_t by Eq. (16)
      Determine Threshold_{T_i} use Eq. (12)
      Identify conflict indices C_i use Eq. (14)
      Similarly, compute harmony indices R_i for recovery
      Assign M_mask^{T_i} and M_recover^{T_i} use Eq. (17) and Eq. (18)
      Final mask update:
      M^{T_i} = M^{T_i} − (M_{C_i}^{T_i} − M_mask^{T_i}) + M_recover^{T_i}
    end for
  end if
  // Parameter Update with Masking
  θ_{t+1} = θ_t − η E[∇L_{T_i}(θ ⊙ M̃^{T_i})] ⊙ M^{T_i}
end for
```

5 Experiments

In this section, we conduct comprehensive experiments on the Meta-World (Yu et al. 2020b) benchmark (covering

Method	Meta-World 5 Tasks		Meta-World 30 Tasks		Meta-World 50 Tasks	
	Near-optimal	Sub-optimal	Near-optimal	Sub-optimal	Near-optimal	Sub-optimal
MTDIFF	**100.0** $\pm$ 0.0	66.30 $\pm$ 2.31	67.52 $\pm$ 0.35	54.21 $\pm$ 1.10	61.32 $\pm$ 0.89	48.94 $\pm$ 0.95
MTDT	**100.0** $\pm$ 0.0	64.67 $\pm$ 5.25	71.89 $\pm$ 0.95	49.33 $\pm$ 2.05	65.80 $\pm$ 1.02	42.33 $\pm$ 1.89
Prompt-DT*	**100.0** $\pm$ 0.0	67.00 $\pm$ 2.33	71.30 $\pm$ 0.67	54.33 $\pm$ 0.78	71.60 $\pm$ 1.40	51.40 $\pm$ 0.35
HarmoDT-R*	**100.0** $\pm$ 0.0	68.00 $\pm$ 2.33	80.10 $\pm$ 3.12	59.00 $\pm$ 3.66	74.24 $\pm$ 2.66	52.45 $\pm$ 1.17
HarmoDT-M*	**100.0** $\pm$ 0.0	72.12 $\pm$ 3.48	79.67 $\pm$ 1.46	**62.33** $\pm$ 0.66	**78.80** $\pm$ 0.67	56.20 $\pm$ 0.56
HarmoDT-F*	**100.0** $\pm$ 0.0	**75.04** $\pm$ 2.33	**81.60** $\pm$ 0.78	60.60 $\pm$ 1.12	77.80 $\pm$ 0.66	**57.50** $\pm$ 0.48
SoCo-DT (Ours)	**100.0** $\pm$ 0.0	**80.00** $\pm$ 2.32	**87.38** $\pm$ 1.20	**69.60** $\pm$ 1.12	**85.88** $\pm$ 0.86	**67.19** $\pm$ 0.90

Table 1: Performance comparison on Meta-World with 5, 30, and 50 randomly sampled tasks under near-optimal and sub-optimal datasets. Each result is averaged over three random seeds with 50 evaluations per task. To ensure fair comparison, we adopt the same random seeds as those used in the previous baseline methods.

MT5, MT30, and MT50 configurations) to answer the following key research questions: (1) Can the proposed SoCo-DT method outperform other mainstream multi-task offline RL algorithms on both near-optimal and sub-optimal datasets? (2) How does the diversity of random task combinations affect the stability and generalization capability of our model under MT5 and MT30 settings? (3) How do different design choices and components, such as the soft mask mechanism, task-adaptive sparsity control module and magnitude-sensitive harmony scoring mechanism, influence overall performance?

5.1 Environment and Baselines

We evaluate our method on the Meta-World multi-task robotic manipulation benchmark, which includes 50 tasks with shared dynamics and diverse object interactions. Following recent works (Hu et al. 2024) (He et al. 2023), we adopt random goal settings to assess task generalization. The evaluation metric is the average success rate across tasks.

We consider two types of offline datasets for training: (1) **Near-optimal dataset**: generated by a SAC-Replay (Haarnoja et al. 2018) policy that mixes random and expert trajectories, resulting in high-quality data overall. (2) **Sub-optimal dataset**: composed of early-stage trajectories with only 50% expert demonstrations retained, representing a more challenging and realistic scenario.

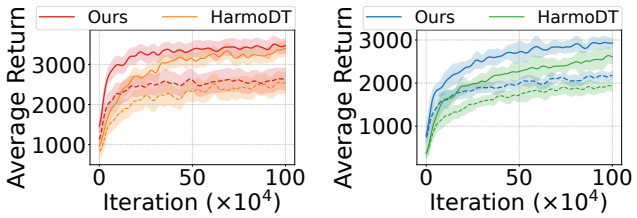
Baselines. We compare our method against several representative multi-task offline RL baselines: (1) **MTBC** (He et al. 2023): Multi-task Behavior Cloning with task ID conditioning. (2) **MTIQL**: Multi-task IQL (Kostrikov, Nair, and Levine 2021) with multi-head critic and task-conditioned policy network. (3) **MTDIFF-P** (He et al. 2023): A diffusion-based policy learning method with prompting and transformer modules. (4) **MTDT** (He et al. 2023): Decision Transformer adapted for multi-task learning with task ID embedding. (5) **Prompt-DT** (Xu et al. 2022): Decision Transformer with trajectory prompt and reward-to-go for unseen tasks. Additionally, we include four representative online RL methods for reference: (6) **CARE** (Sodhani, Zhang, and Pineau 2021): Uses meta-data and encoder mixing for task representation. (7) **PaCo** (Sun et al. 2022): Adopts parameter composition for task-specific parameter reassembly. (8) **Soft-M** (Yang et al. 2020): Uses routing networks for modular soft composition. (9) **D2R** (He et al. 2024): Employs diverse routing paths for different tasks.(10) **HarmoDT** (Hu et al. 2024): as discussed in § 2.2. We use results from (He et al. 2023) for most baselines, * indicate baselines of our own implementation.

5.2 Performance Comparison

We report performance comparisons with mainstream methods on MT5, MT30, and MT50. As shown in Table 1 and 2, our method consistently outperforms all baselines across different task scales. Specifically, it achieves 100% success rate on the near-optimal MT5 dataset and gains an improvement of 4.96% on the sub-optimal dataset. For MT30, our method shows gains of 5.78% and 7.27% on the near-optimal and sub-optimal datasets, respectively. For MT50, it achieves 7.08% and 9.69% improvements. Notably, the performance gains are larger on sub-optimal datasets, indicating stronger stability and robustness when reward signals are ambiguous.

Method	Near-optimal	Sub-optimal
CARE (online)	$46.12_{\pm 1.30}$	–
PaCo (online)	$54.31_{\pm 1.32}$	–
Soft-M (online)	$53.41_{\pm 0.72}$	–
D2R (online)	$63.53_{\pm 1.22}$	–
MTBC	$60.39_{\pm 0.86}$	$34.53_{\pm 1.25}$
MTIQL	$56.21_{\pm 1.39}$	$43.28_{\pm 0.90}$
MTDIFF-P	$59.53_{\pm 1.12}$	$48.67_{\pm 1.32}$
MTDIFF-P-ONEHOT	$61.32_{\pm 0.89}$	$48.94_{\pm 0.95}$
MTDT	$65.80_{\pm 1.02}$	$42.33_{\pm 1.89}$
Prompt-DT*	$71.60_{\pm 1.40}$	$51.40_{\pm 0.35}$
HarmoDT-R*	$74.24_{\pm 2.66}$	$52.45_{\pm 1.17}$
HarmoDT-M*	$\mathbf{78.80}_{\pm 0.67}$	$56.20_{\pm 0.56}$
HarmoDT-F*	$77.80_{\pm 0.66}$	$\mathbf{57.50}_{\pm 0.48}$
SoCo-DT(ours)	$\mathbf{85.88}_{\pm 0.86}$	$\mathbf{67.19}_{\pm 0.90}$

Table 2: Average success rate across 3 seeds on MT50 with random goals (MT50-rand) under both near-optimal and sub-optimal cases. Each task is evaluated for 50 episodes.



(a) Return comparison on MT5 (b) Return comparison on MT30

Figure 4: Average return comparison for MT5 and MT30 task settings. (a) shows average return curves for MT5 and its sub-task variant; (b) shows the same for MT30. Both are compared with the state-of-the-art method HarmoDT-F. Dashed lines represent sub-optimal datasets.

Setting	Method	Success Rate
5 Tasks	HarmoDT-F	Near-opt: 95.33 ± 5.25 Sub-opt: 74.67 ± 7.37
	SoCo-DT (Ours)	Near-opt: **96.20** ± 4.42 Sub-opt: **79.43** ± 13.77
30 Tasks	HarmoDT-F	Near-opt: 81.30 ± 3.12 Sub-opt: 63.70 ± 2.35
	SoCo-DT (Ours)	Near-opt: **86.27** ± 2.58 Sub-opt: **67.63** ± 1.10

Table 3: Performance comparison on random task combinations. Results are averaged over three seeds.

To further verify robustness under varying task combinations, we randomly sample 5 and 30 tasks from the Meta-World using three random seeds and compare with HarmoDT-F. Results are shown in Table 3 and Figure 4.

5.3 Ablation Study

To evaluate each component's contribution, we conduct ablation studies on the near-optimal MT30 dataset, covering both architectural modules and key hyperparameters.

Component Ablation. We evaluate the average success rate of model variants by ablating or replacing key components under the same environment: **(1) Without Soft Mask Mechanism**: Replace soft masks with hard masks that directly zero out conflicting parameters without importance weighting. **(2) Fixed Mask Sparsity Mechanism**: Use fixed mask sparsity instead of IQR-based dynamic adjustment. **(3) Without Magnitude-Sensitive Harmony Scoring Mechanism**: Replace magnitude-sensitive harmony scoring with conventional dot product during recovery. Results are shown in Table 4.

Hyperparameter Ablation. We further analyze the impact of key hyperparameters on the model's performance. The hyperparameters include: **(1) Initial Mask Sparsity**: Specifies the initial proportion of parameters to be masked, which controls the starting level of sparsity in the model. **(2) Tolerance Parameter** α: Determines the level of tolerance for gradient magnitude discrepancy in the same direction during parameter recovery, see Eq. (9). **(3) Asymmetric Cosine Annealing for** β_t: Controls the dynamic adjustment range of the conflict and harmony thresholds during the selection of conflicting and harmonious parameters, see Eq. (13). **(4) Initial Quantile Values** q_1 **and** q_3: Define the initial quantile points used in the IQR method for task-adaptive threshold computation, see Eq. (12). The experimental results are illustrated in Figure 5.

Datasets	Method	Avg. Success Rate	Avg. Return
near-optimal	A+M	72.00	2549.8
	S+M	76.33	2635.7
	S+A	76.00	2577.8
	S+A+M	**86.33**	**3072.7**
sub-optimal	A+M	61.33	2094.5
	S+M	59.67	2037.8
	S+A	59.33	1871.1
	S+A+M	**69.60**	**2417.7**

Table 4: Results of the component ablation study. S, A, and M denote the Soft Mask Mechanism, Adaptive Mask Sparsity Mechanism, and Magnitude-Sensitive Harmony Scoring Mechanism, respectively. The table reports average success rate and return for different component combinations on both near-optimal and sub-optimal datasets under the MT30.

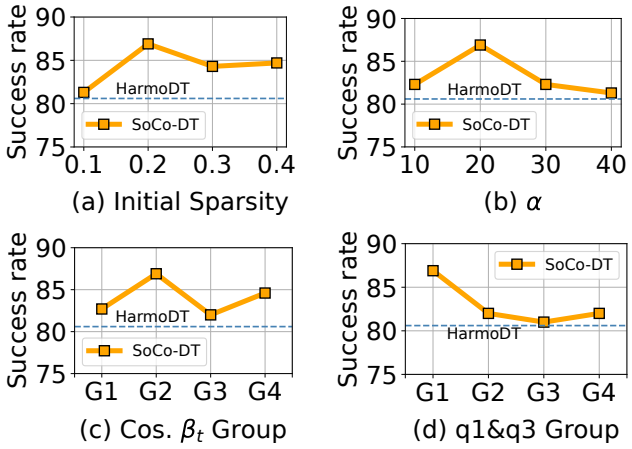


Figure 5: Ablation study on the near-optimal MT30 setting. (c) Cosine annealing schedule β_t, G1–G4: (10-5-20), (20-5-30), (30-5-40), (20-10-30); (d) Quantile thresholds (q_1, q_3), G1–G4: (0.05-0.95), (0.1-0.9), (0.2-0.8), (0.3-0.7).

6 Conclusion

In this study, we identify limitations in existing mask-based methods for multi-task reinforcement learning and propose a soft masking mechanism to reduce the negative impact of traditional hard masks on task-critical parameters, improving model generalization and learning efficiency. We also introduce a task-aware mask update with adaptive sparsity strategy, dynamically adjusting mask sparsity based on task complexity. Extensive experiments demonstrate the superior performance of our proposed method.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant Nos. 2021YFA1000102 and 2021YFA1000103), the Shandong Provincial Natural Science Foundation (Grant Nos. ZR2024MF129 and ZR2025QC1540), and the Fundamental Research Funds for the Central Universities (Grant No. 25CX06034A).

References

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chai, H.; Yin, Z.; Ding, Y.; Liu, L.; Fang, B.; and Liao, Q. 2022. A model-agnostic approach to mitigate gradient interference for multi-task learning. *IEEE Transactions on Cybernetics*, 53(12): 7810–7823.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Chen, Z.; Ngiam, J.; Huang, Y.; Luong, T.; Kretzschmar, H.; Chai, Y.; and Anguelov, D. 2020. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33: 2039–2050.

D'Eramo, C.; Tateo, D.; Bonarini, A.; Restelli, M.; and Peters, J. 2024. Sharing knowledge in multi-task deep reinforcement learning. *arXiv preprint arXiv:2401.09561*.

Ghasemi, P.; Greenberg, M.; Southern, D. A.; Li, B.; White, J. A.; and Lee, J. 2025. Personalized decision making for coronary artery disease treatment using offline reinforcement learning. *npj Digital Medicine*, 8(1): 99.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. Pmlr.

He, H.; Bai, C.; Xu, K.; Yang, Z.; Zhang, W.; Wang, D.; Zhao, B.; and Li, X. 2023. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in neural information processing systems*, 36: 64896–64917.

He, J.; Li, K.; Zang, Y.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2024. Not all tasks are equally difficult: Multi-task deep reinforcement learning with dynamic depth routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12376–12384.

He, J.; Li, K.; Zang, Y.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2025. Goal-Oriented Skill Abstraction for Offline Multi-Task Reinforcement Learning. *arXiv preprint arXiv:2507.06628*.

Hu, S.; Fan, Z.; Shen, L.; Zhang, Y.; Wang, Y.; and Tao, D. 2024. Harmodt: Harmony multi-task decision transformer for offline reinforcement learning. *arXiv preprint arXiv:2405.18080*.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Kumar, A.; Singh, A.; Tian, S.; Finn, C.; and Levine, S. 2021. A workflow for offline model-free robotic reinforcement learning. *arXiv preprint arXiv:2109.10813*.

Lee, K.-H.; Nachum, O.; Yang, M. S.; Lee, L.; Freeman, D.; Guadarrama, S.; Fischer, I.; Xu, W.; Jang, E.; Michalewski, H.; et al. 2022. Multi-game decision transformers. *Advances in neural information processing systems*, 35: 27921–27936.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Shi, G.; Li, Q.; Zhang, W.; Chen, J.; and Wu, X.-M. 2023. Recon: Reducing conflicting gradients from the root for multi-task learning. *arXiv preprint arXiv:2302.11289*.

Shi, T.; Chen, D.; Chen, K.; and Li, Z. 2021. Offline reinforcement learning for autonomous driving with safety and exploration enhancement. *arXiv preprint arXiv:2110.07067*.

Sodhani, S.; Zhang, A.; and Pineau, J. 2021. Multi-task reinforcement learning with context-based representations. In *International conference on machine learning*, 9767–9779. PMLR.

Sun, L.; Zhang, H.; Xu, W.; and Tomizuka, M. 2022. Paco: Parameter-compositional multi-task reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 21495–21507.

Sun, T.; Shao, Y.; Li, X.; Liu, P.; Yan, H.; Qiu, X.; and Huang, X. 2020. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 8936–8943.

Tang, A.; Shen, L.; Luo, Y.; Ding, L.; Hu, H.; Du, B.; and Tao, D. 2023. Concrete subspace learning based interference elimination for multi-task model fusion. *arXiv preprint arXiv:2312.06173*.

Teh, Y.; Bapst, V.; Czarnecki, W. M.; Quan, J.; Kirkpatrick, J.; Hadsell, R.; Heess, N.; and Pascanu, R. 2017. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; Tsvetkov, Y.; Firat, O.; and Cao, Y. 2020. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *arXiv preprint arXiv:2010.05874*.

Xu, M.; Shen, Y.; Zhang, S.; Lu, Y.; Zhao, D.; Tenenbaum, J.; and Gan, C. 2022. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, 24631–24645. PMLR.

Yang, R.; Xu, H.; Wu, Y.; and Wang, X. 2020. Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33: 4767–4777.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020a. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020b. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, 1094–1100. PMLR.

Zhang, Y.; Qian, Y.; Ma, G.; Zheng, K.; Liu, G.; and Zhang, Q. 2024. Learning multi-task sparse representation based on fisher information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16899–16907.