# Unified Prediction and Planning via Conflict-Aware Disjoint Parameter Training

Taewon Seo[1⋆], Seonae Jeon[1⋆], Giwon Lee[2⋆], Kuk-Jin Yoon[2†], and Daehee Park[1†]

[1] DGIST, Republic of Korea
{taewonseo,seonae,dhpark}@dgist.ac.kr
[2] KAIST, Republic of Korea
{dlrldnjs,kjyoon}@kaist.ac.kr

**Abstract.** Accurate motion prediction of surrounding agents and safe motion planning are two closely coupled key tasks for social robot navigation in crowded environments. Deploying these systems on resource-constrained edge devices necessitates compact, unified models that can perform both tasks simultaneously. However, within these compact shared encoders, recent unified models often overlook severe representational conflicts that arise from the distinct objectives of predicting neighbor behaviors versus ego-centric safety planning. To address this issue, we first identify the *Skill Conflict*—a phenomenon where overlapping parameter assignments cause distinct tasks to compete for the same weights, preventing the model from fully specializing in individual skills. To resolve this, we propose a novel model-merging-based framework, **Disjoint Parameter Training (DPT)**. DPT mitigates performance degradation caused by Skill Conflict through distributed parameter learning, which separates the key parameter regions of each task while preserving their core capabilities prior to merging. In addition, we observe that sparse merging, which selectively integrates only the most influential parameters for each task rather than combining all task-specific parameters, yields optimal performance by preventing interference among adjacent features and concentrating representational capacity. DPT can be applied in parallel with a variety of merging methods. Evaluated on standard crowd navigation benchmarks (JRDB and JTA), our framework demonstrates superior performance, validating its versatility and effectiveness for safe, resource-efficient robot navigation. The project page is available at: https://dpt2026.github.io/

**Keywords:** Motion Prediction · Motion Planning · Model Merging.

## 1 Introduction

Motion planning [6, 47, 76, 78] aims to generate future trajectories of an ego agent based on the surrounding environment and the behaviors of nearby agents.

⋆ Equal contribution. † Corresponding authors.

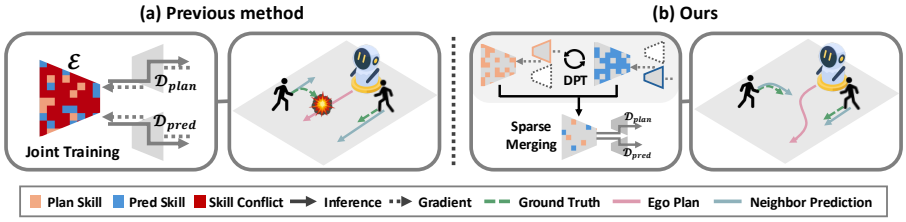


**Fig. 1:** Comparison of learning strategies in unified models. (a) In previous methods, the skills of each task are learned within a single shared encoder, causing conflicts between the two skills in the parameter space. (b) DPT resolves this issue by separately training the major parameter regions allocated to each task and Sparse Merging combines only their core parameter regions, thereby mitigating skill conflicts within the shared encoder.

For safe and socially compliant navigation in highly interactive scenarios, such as mobile robots navigating through dense crowds, the agent must accurately predict how humans will move and plan its own trajectory accordingly [55]. Therefore, achieving both accurate prediction and safe planning simultaneously within a constrained computational budget is essential for reliable and user-friendly mobile service robots.

To realize this efficiently, recent research has focused on unified frameworks [15–17,24] that jointly perform prediction and planning, replacing traditional sequential pipelines [31, 41, 81, 84]. Such unified models [15–17] typically use a shared encoder that serves as the representational backbone for both tasks, as shown in Fig. 1 (a). While these architectures achieve strong joint reasoning performance, they generally overlook representational conflicts that arise in the shared encoder, especially when the model capacity is limited for edge deployment. Prediction requires representations sensitive to the varied behaviors and intentions of surrounding agents [5,11,49,85], whereas planning demands ego-centric features that prioritize safety and feasibility [7, 48, 71, 83]. Optimizing both objectives on the same compact encoder leads to conflicting gradient updates, which can severely degrade the feature quality of both tasks [28, 29, 52, 72].

To address this challenge without simply inflating model size, we propose a model-merging-based unified framework that explicitly mitigates task interference in the shared encoder, as shown in Fig. 1 (b). Model merging [67] is a methodology that combines the parameters of models fine-tuned for individual tasks to create a single model that can effectively perform all tasks simultaneously. Unlike ensemble learning [2, 34] or mixture-of-experts [12, 14, 22, 62], which handle multiple objectives by maintaining multiple models or expert branches, model merging produces a single model, making it effective for autonomous driving where inference latency is critical and both objectives must be achieved within a single deployment. Existing model merging methods, however, have primarily been studied in the context of large-scale models (e.g., LLMs) with massive parameter counts and have mainly focused on language-

related tasks [4, 23, 33, 39, 56, 68, 77]. Therefore, a significant gap exists between these approaches and resource-constrained robot navigation settings, which rely on compact unified models. This discrepancy induces a critical issue during the merging process: in models with limited capacity, the parameter regions crucial for each task tend to densely overlap and conflict with one another. When the overall model size is small, each task must utilize a larger portion of the shared parameters to achieve strong single-task performance, which inevitably intensifies this interference.

We define this phenomenon as the *Skill Conflict*, a condition where overlapping parameter assignments cause distinct tasks to compete for the same weights, preventing the model from fully specializing in individual skills. To resolve this, we propose **Disjoint Parameter Training (DPT)**, which fine-tunes task-specific models by distributing the parameter learning regions across tasks, and then merges them into a single compact unified model. DPT separates the learning regions of each task by exchanging model-sized binary masks, each indicating a unique parameter region assigned to that task. Moreover, instead of allocating all parameter regions for training from the beginning, it gradually expands the learnable parameter space from a local region, allowing each task's core skill to be compactly represented within a small parameter area. Building on this property, we further apply sparse merging, which selectively integrates only the most influential parameters from each task rather than merging all, enabling more efficient and skill-focused unification of the shared encoder while preventing adjacent feature interference. This design enables each task to specialize in distinct subspaces of the shared representation, preserving both prediction accuracy and planning safety within a limited model capacity.

Our main contributions are summarized as follows:

- We define and analyze the *Skill Conflict* problem caused by shared encoder representations in compact, unified prediction-planning models for robot navigation.
- We propose Disjoint Parameter Training (DPT), which separates and localizes each task's core skills within distinct parameter regions, and we introduce sparse merging to prevent representational interference among neighboring features.
- Our method achieves superior performance, maintaining accurate prediction and safe planning across standard crowd navigation benchmarks (JRDB and JTA), validating its effectiveness for resource-efficient mobile robots.

## 2 Related Works

### 2.1 Unified Motion Prediction and Planning

Motion prediction and motion planning are the key tasks for achieving safe navigation in mobile robots and autonomous systems. Motion prediction forecasts the future trajectories of surrounding agents based on current and past information [3,9,38,43,45,54,69,75], while motion planning generates the ego agent's trajectory using these predictions and scene context [26, 57, 64, 66, 80]. Conventional

sequential pipelines predict first and plan later [25, 27, 40, 41, 79], training modules independently [1, 31, 44, 51, 59], which increases real-time inference latency and prevents learning interdependent task relationships [20, 21, 30, 32, 50, 55, 60]. To address these bottlenecks, recent studies have developed unified models that jointly reason over planning and prediction within a single framework [15–17]. However, these prior studies often rely on scaling up model parameters, overlooking the representational task conflict that arises in the shared encoder due to the distinct objectives of prediction and planning. This conflict becomes particularly fatal in compact models deployed on resource-constrained edge devices. In this work, we verify the existence of such task conflicts in unified models and propose a framework that mitigates them through task-wise distributed learning, ensuring high performance for both tasks.

### 2.2 Model Merging

Model merging [67] combines the parameters of distinct fine-tuned models into a single architecture to perform multiple tasks simultaneously. While early approaches relied on simple or weighted parameter averaging [19,37,61], direct combination often increases task interference and degrades performance. To mitigate this, Task Arithmetic [18] uses task vectors to preserve directional characteristics, Ties Merging [65] resolves sign conflicts via majority voting, and Localize-and-Stitch [13] learns local masks to isolate critical parameter regions. Despite these advancements in reducing task conflicts [8, 35, 63, 73, 74], most existing methods are tailored exclusively for Large Language Models (LLMs) equipped with massive parameter capacities [53, 58, 70, 82]. These highly parameterized settings differ fundamentally from our target domain, which requires deploying compact unified models on resource-constrained edge devices for robot prediction and planning. In such compact architectures, parameter overlap inevitably leads to severe representational interference. Therefore, we identify and analyze this *Skill Conflict* and propose the Disjoint Parameter Training (DPT) framework to actively prevent parameter collision, successfully maximizing multi-task performance within a strictly limited parameter budget.

## 3 Methods

We aim to train a unified model that jointly handles motion prediction and motion planning. Among the following sections, Sec. 3.1 provides the problem definition, describing the objective we aim to achieve. Sec. 3.2 introduces preliminaries on model merging and task vectors, and clarifies why naive fine-tuning and merging, which work reasonably well for large foundation models, are problematic for compact unified models. Sec. 3.3 provides empirical evidence of *Skill Conflict* in a shared-encoder prediction–planning setting. Sec. 3.4 introduces *Disjoint Parameter Training (DPT)*. This fine-tuning strategy prepares task-specialized *material* models, which act as individual fine-tuned ingredients for the merging process, with strictly non-overlapping core regions. Finally, Sec. 3.5

describes our *Sparse Merging* strategy that combines these material models by using only salient task-vector components.

### 3.1 Problem Definition

We consider a unified architecture with a shared encoder and two task heads: a prediction decoder and a planning decoder. Let the input space be $\mathbb{X}$ and the output space be $\mathbb{Y}$, where $X \in \mathbb{X}$ and $Y \in \mathbb{Y}$. In a scene with $N$ agents, agent 0 denotes the ego, and agents $1 : N-1$ are surrounding agents.

To capture rich human behaviors, the multi-modal input incorporates human trajectories, poses, and bounding boxes, denoted as $X_{\text{traj}}$, $X_{\text{pose}}$, and $X_{\text{box}}$, respectively. The unified model input is defined as:

$$X = \left\{ X_{\text{traj},\,0:N-1}^{-T_{\text{obs}}:0},\ X_{\text{pose},\,0:N-1}^{-T_{\text{obs}}:0},\ X_{\text{box},\,0:N-1}^{-T_{\text{obs}}:0} \right\}. \tag{1}$$

The output is defined as:

$$Y = \{ Y_{\text{plan}},\ Y_{\text{pred}} \}, \tag{2}$$

where the planned future ego trajectory is

$$Y_{\text{plan}} = \left\{ Y_0^{1:T_{\text{fut}}} \right\}, \tag{3}$$

and the predicted future trajectories of neighbors are

$$Y_{\text{pred}} = \left\{ Y_{1:N-1}^{1:T_{\text{fut}}} \right\}. \tag{4}$$

Here, $T_{\text{obs}}$ is the number of observed steps and $T_{\text{fut}}$ is the prediction/planning horizon. Our goal is to learn the shared parameters so that the unified model can produce safe $Y_{\text{plan}}$ and accurate $Y_{\text{pred}}$. We specifically focus on mitigating representational interference within the shared encoder, a critical bottleneck when determining parameter assignments in compact models.

### 3.2 Preliminaries: Model Merging and Task Vectors

Model merging begins by preparing task-specialized "material" models, and then combines them into a unified model. Let $\Theta_0$ be a pretrained backbone and $t \in T = \{\text{pred}, \text{plan}\}$ denote the target tasks. In the *naive material preparation*, each task is fine-tuned *independently* from the same initialization $\Theta_0$, with no parameter or gradient sharing across tasks:

$$\Theta_t = \arg\min_{\Theta} \mathbb{E}_{(X,Y)\sim\mathcal{D}_t} \left[ \mathcal{L}_t(f_\Theta(X), Y) \right]. \tag{5}$$

The change induced by independent fine-tuning defines the *task vector* [18]:

$$\tau_t := \Theta_t - \Theta_0. \tag{6}$$

After preparing $\Theta_t$ (or equivalently $\tau_t$), a merged model is produced by an operator that combines task contributions:

$$\Theta_{\text{merge}} = \mathcal{M}\left(\Theta_0, \tau_t, \phi\right), \tag{7}$$

where $\mathcal{M}$ denotes a chosen merging family (e.g., scalar-weighted linear combination, coordinate-wise masking, or learned gating) and $\phi$ denotes the merging coefficient associated with the operator $\mathcal{M}$.

For massive foundation models, naive per-task fine-tuning followed by merging often works reasonably well, as their massive capacity naturally allocates distinct skills into separate, sparse parameter subspaces. This was observed in the prior work [13], which found that utilizing only the top 10% (or even 1%) of a task vector can fully preserve single-task performance in large models. However, in compact unified models designed for edge robots, parameter capacity is strictly limited. Consequently, independent fine-tuning forces different tasks to update densely overlapping regions. This overlap causes interference when models are merged, particularly within the shared encoder that both tasks rely on. We find that the other merging methods in Tab. 4 do not yield effective performance on the compact unified model, which supports our claim.

### 3.3 Empirical Evidence of Skill Conflict in a Shared Encoder

We begin by defining a binary mask of the same size as the model parameters to specify how much of a task vector is utilized. For task $t \in T = \{\text{pred}, \text{plan}\}$ and utilization ratio $K$, the *activation mask* $M_t^{(K)}$ selects the top-$K$ coordinates of the task vector by magnitude:

$$M_t^{(K)} = \mathbb{I}\big(|\tau_t| \geq \lambda_{K\%}(|\tau_t|)\big), \tag{8}$$

where $\lambda_{K\%}(\cdot)$ is the $K$-th percentile of $|\tau_t|$. We refer to $K$ as the *mask ratio*, representing the task-vector utilization percentage. By adjusting this *mask ratio*, we can control the capacity allocated to each task and systematically observe how differing levels of parameter usage impact both individual and joint performance.

To understand how capacity limitations affect multi-task merging, we first investigate the relationship between parameter utilization and task performance using Fig. 2. This figure sweeps the *mask ratio* $K \in [10\%, 90\%]$ and reports the performance (1/ADE; higher is better). For each $K$, we evaluate single-task merges (*Plan Merged*: $\Theta_0 + M_{\text{plan}}^{(K)} \odot \tau_{\text{plan}}$, *Pred Merged*: $\Theta_0 + M_{\text{pred}}^{(K)} \odot \tau_{\text{pred}}$) and a two-task merge (*Both Merged*: $\Theta^{(K,40\%)} = \Theta_0 + M_{\text{target}}^{(K)} \odot \tau_{\text{target}} + M_{\text{other}}^{(40\%)} \odot \tau_{\text{other}}$), where the non-target task is held at a constant 40% utilization. In the single-task cases, performance steadily improves as $K$ increases and plateaus at a high level, confirming that utilizing a larger portion of the task vector is beneficial when merged alone. Conversely, in the two-task setting, attempting to increase $K$ yields only marginal gains before rapidly saturating at a substantially lower ceiling. This early saturation demonstrates that greedy parameter utilization by one task severely degrades performance of the other task.

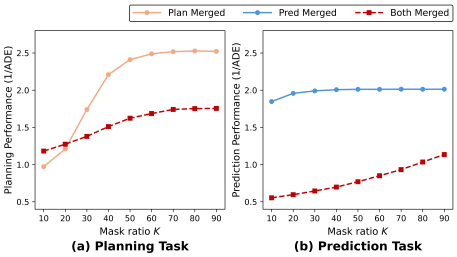


**Fig. 2:** Planning and Prediction performance according to the *mask ratio* $K$. *Plan Merged* and *Pred Merged* use the single task vectors corresponding to the mask ratio. In *Both Merged*, the parameter utilization of the non-target task is fixed at 40%, while that of the target task is used according to the given parameter ratio.

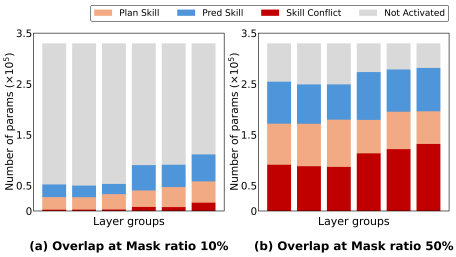


**Fig. 3:** Parameter distribution by layer group and the degree of *Skill Conflict* between *Plan* and *Pred Skill* at different mask ratios. Both tasks use an equal proportion of task vectors for the given mask ratio. At (a) 10% mask ratio, little *Skill Conflict* occurs, whereas at (b) 50% ratio, significant conflict is observed.

To diagnose the underlying cause of this early saturation, Fig. 3 visualizes how the prediction and planning tasks allocate parameters within the shared encoder. We categorize the parameters into four groups based on their activation masks $M_{\text{plan}}^{(K)}$ and $M_{\text{pred}}^{(K)}$: *Plan Skill*, *Pred Skill*, *Skill Conflict* (activated by both), and *Not Activated*. We then quantify the exact coordinate-wise collision by defining the overall overlap as:

$$\text{Overlap}(K) \;=\; \frac{1}{D}\sum_{j=1}^{D}\left(M_{\text{pred}}^{(K)}[j] \wedge M_{\text{plan}}^{(K)}[j]\right) \times 100\%, \tag{9}$$

where $D$ is the total number of parameters activated by at least one task. The analysis reveals that at a conservative mask ratio ($K$=10%), the tasks activate small, largely distinct regions with minimal conflict (Overlap 8.35%). However, as both tasks expand their utilization to $K$=50%, their activated regions heavily overlap within the same layer groups, causing the overlap metric to surge to 39.25%. Thus, the more we try to make the model stronger for each individual task by utilizing larger parameter regions, the more entangled the shared encoder becomes, which explains the early saturation of *Both Merged* in Fig. 2.

All empirical studies are conducted on the Social-Transmotion [46] backbone, with detailed architectural specifications provided in Supplementary Section 2. We also conduct a similar analysis on DTPP [15] (Supplementary Section 6), showing that our findings are common across different backbones. This motivates preparing material models that minimize overlap (Sec. 3.4) and merging them sparsely (Sec. 3.5).

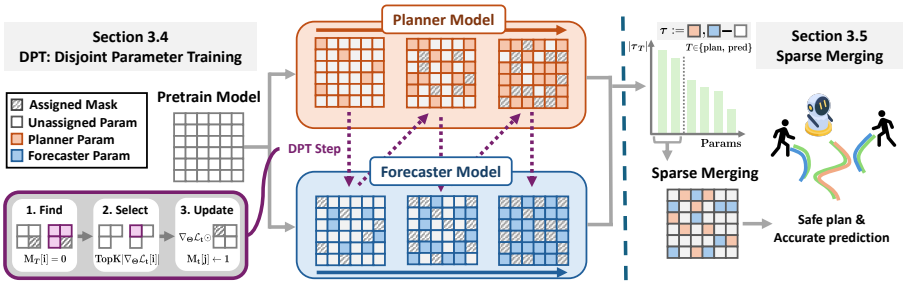


**Fig. 4:** Overview of our proposed framework. In the DPT step, task-specific models for prediction and planning are generated by separating training regions with binary masks to minimize skill conflict. In the Sparse Merging, the two fine-tuned models are combined into a single unified model by utilizing only the most significant task vectors, effectively preserving each task's skill.

### 3.4 DPT: Preparing Task-Specialized Material Models

To resolve the observed conflict, we focus on preparing highly disjoint, task-specialized *material* models prior to merging. We propose *Disjoint Parameter Training* (DPT), a strategy that progressively expands task-critical regions for each task while strictly preventing spatial overlap, as illustrated in Fig. 4. DPT systematically alternates training between prediction and planning tasks. At each alternating step, it computes the task-specific loss and gradients across the entire parameter space. Instead of updating all weights, it selects the top-$K$ unassigned parameters with the largest absolute gradients, permanently allocates them to the current task via a binary mask, and updates only these activated regions. During the first half of the training process, these task-specific masks incrementally grow. In the second half, the masks are frozen, allowing the model to consolidate each task's core region without further spatial expansion. This explicit separation produces fine-tuned models ($f_{\text{plan}}$ and $f_{\text{pred}}$) whose core parameter spaces in the shared encoder are mutually exclusive by construction, drastically reducing interference at merge time. The complete procedure is summarized in Algorithm 1.

### 3.5 Sparse Merging of Prepared Models

Although DPT effectively eliminates direct parameter overlap between tasks, we observe that combining the full masked coordinates of the fine-tuned task vectors still degrades joint performance in compact models. This indicates that even though the fetched parameters do not overlap spatially due to the disjoint masks, local functional interactions still persist: parameter updates in adjacent coordinates or nearby layers can easily couple features. Consequently, indiscriminately combining full vectors increases cross-task interference, even when their primary supports are largely disjoint. To address this, we restrict each skill to a narrow, high-signal subset, which yields significantly better composability. Operationally,

**Algorithm 1:** Disjoint Parameter Training (DPT)

**Data:** Tasks $T = \{\text{plan}, \text{pred}\}$, total epochs $E$, learning rate $\eta$, mask size $K$, pretrained model $\Theta_0$
**Result:** Fine-tuned models $f_{\text{plan}}$, $f_{\text{pred}}$ with disjoint parameter regions

Initialization: $M_{\text{plan}}, M_{\text{pred}} \leftarrow 0$, $f_{\text{plan}}, f_{\text{pred}} \leftarrow \Theta_0$
**for** $e = 1$ ***to*** $E/2$ **do**
  **for** $t \in T$ **do**
    Compute $\mathcal{L}_t$ and $\nabla_\Theta \mathcal{L}_t$
    `// Find unassigned parameters`
    $\mathcal{U} = \{\, i \mid M_{\text{plan}}[i] = 0 \ \wedge \ M_{\text{pred}}[i] = 0 \,\}$
    `// Select Top-`$K$` gradient indices`
    $\mathcal{I}_{\text{top-}K} \leftarrow \text{TopK}_{i \in \mathcal{U}} |\nabla_\Theta \mathcal{L}_t[i]|$
    `// Update mask and model`
    $M_t[j] \leftarrow 1$ for $j \in \mathcal{I}_{\text{top-}K}$ $\quad f_t \leftarrow f_t - \eta \nabla_\Theta \mathcal{L}_t \odot M_t$
**for** $e = E/2$ ***to*** $E$ **do**
  **for** $t \in T$ **do**
    `// Update model with fixed mask`
    $f_t \leftarrow f_t - \eta \nabla_\Theta \mathcal{L}_t \odot M_t$
Return $f_{\text{plan}}$ and $f_{\text{pred}}$.

we introduce **Sparse Merging**, which integrates only a highly restrictive fraction of the largest-magnitude coordinates from each task vector utilizing the previously defined mask $M_t^{(K\%)}$. We define our robust merging operation as:

$$\Theta_{\text{merged}} = \Theta_0 + M_{\text{pred}}^{(\text{K}\%)} \odot \tau_{\text{pred}} + M_{\text{plan}}^{(\text{K}\%)} \odot \tau_{\text{plan}}. \tag{10}$$

By utilizing only the most salient coordinates, Sparse Merging strictly preserves each task's core skill while actively suppressing detrimental local interactions within the compact shared encoder. Empirically, this sparse approach consistently outperforms dense merging ($K$=100%) and moderately sparse configurations (e.g., $K$=10%) across all joint metrics, validating that isolating skills into narrow parameter footprints is essential for high-performance multi-task composability on edge devices.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets & Backbone.** We conduct experiments on two datasets that provide visual cues: the JTA [10] and JRDB [36] datasets. JRDB is a real-world dataset collected in diverse indoor and outdoor environments and provides pedestrian trajectories with annotated bounding boxes. JTA is a large-scale synthetic dataset that provides rich annotations for pedestrian motion. For both datasets, we follow the standard protocol of predicting the future 12 time steps given the

past 9 time steps at 2.5 frames per second (fps). Additional dataset details are provided in Supplementary Section 1.

The unified backbone is based on social-transmotion [46]. The original architecture, which was designed solely for the prediction task, is extended by adding a planning decoder, enabling the model to perform both prediction and planning jointly. During pretraining, we employ a game-theoretic loss [24], one of the popular joint learning paradigms. Further architectural and learning details of the backbone model are provided in Supplementary Sections 1 and 2.

**Baselines.** Our method aims to achieve both prediction and planning effectively within a single unified model. Therefore, we compare it against other unified architectures that share the same task objectives (1, 2), as well as multi-task learning approaches (3, 4, 5, 6) that jointly perform related tasks. (1) **DIPP** [17] is a jointly unified framework of neural network based on a Transformer that uses a differentiable nonlinear optimizer as a motion planner, enabling all operations differentiable. (2) **DTPP** [15] is also a joint differentiable unified framework that integrates ego conditioned prediction and learnable cost models in a tree structured policy planner. (3) **Ensemble** [2] combines the prediction outputs of a planner-finetuned model and a forecaster-finetuned model by averaging their outputs, without modifying any model parameters. (4) **Task Arithmetic** [18] merges finetuned models by extracting task vectors from the pretrained model and linearly combines these vectors by weighted averaging. (5) **Ties Merging** [65] selects only the important parameters from each task vector, resolves sign conflicts, and merges the remaining parameters in the direction of sign agreement by averaging them. (6) **Localize-and-Stitch** [13] identifies task-critical parameter regions within fine-tuned models by learning a binary mask and merges only 1% of a sparse subset into the pretrained model's weights. (7) **T-Switch** [42] is a dynamic model merging method that represents task vectors in a binary form and selectively activates task-specific parameter directions through a switching mechanism.

**Metrics.** For planning evaluation, we use core metrics corresponding to the categories of *Effectiveness*, *Safety*, and *Goal Success*, which are essential for safe driving. ADE (Average Displacement Error) measures average displacement error between the predicted trajectory and the ground-truth trajectory over the full horizon, reflecting *Effectiveness* of trajectory. Collision Rate quantifies *Safety* by computing the fraction of predicted ego positions that are within a collision threshold of surrounding ground-truth agents; we use a distance threshold of 0.6m. FDE (Final Displacement Error) measures the displacement error at the final timestep, serving as a proxy for *Goal Success*. Miss Rate also evaluates *Goal Success* by checking whether the final ego position deviates from the ground truth by more than 0.5m. For prediction evaluation, we estimate ADE and FDE over neighbor agents by using the same definitions as above. Detailed definitions of the evaluation metrics are provided in Supplementary Section 3.

**Table 1:** Applicability of the DPT to different model merging strategies on the JRDB and JTA datasets. The upper rows show task-specific fine-tuned models, while the lower rows show the effect of applying different model merging methods with and without DPT. Sparse Merging is performed with a mask ratio of $K$=1%.

| Method | JRDB | | | | | | JTA | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Planning Metric | | | | Prediction Metric | | Planning Metric | | | | Prediction Metric | |
| | Effectiveness (ADE↓) | Safety (Col. rate↓) | Goal Success (FDE↓) | Goal Success (Miss rate↓) | ADE↓ | FDE↓ | Effectiveness (ADE↓) | Safety (Col. rate↓) | Goal Success (FDE↓) | Goal Success (Miss rate↓) | ADE↓ | FDE↓ |
| Plan Finetune | 0.3965 | 0.0078 | 0.7371 | 0.3700 | 1.9825 | 2.4087 | 1.3273 | 0.0056 | 2.6118 | 0.9144 | 8.5329 | 9.8826 |
| Plan Finetune + DPT | 0.3699 | 0.0073 | 0.7244 | 0.3600 | 0.7368 | 1.1702 | 1.0008 | 0.0055 | 2.0300 | 0.8877 | 1.2079 | 2.3390 |
| Pred Finetune | 0.8889 | 0.0195 | 1.2140 | 0.9615 | 0.4967 | 0.9144 | 5.1324 | 0.0052 | 8.9957 | 0.9960 | 1.4394 | 2.7745 |
| Pred Finetune + DPT | 0.8469 | 0.0200 | 1.1244 | 0.9189 | 0.5250 | 0.9682 | 1.1139 | 0.0058 | 2.2594 | 0.9612 | 1.1834 | 2.3184 |
| Task Arithmetic [18] | 0.9387 | 0.0189 | 1.1798 | 0.8887 | 0.9904 | 1.3870 | 3.4726 | 0.0058 | 6.1710 | 0.9973 | 4.1518 | 5.7700 |
| Task Arithmetic + DPT | **0.5865** | **0.0145** | **0.8859** | **0.4283** | **0.5849** | **1.0308** | **1.0373** | **0.0056** | **2.0936** | **0.9278** | **1.1406** | **2.2367** |
| Ties Merging [65] | 0.8759 | 0.0167 | 1.1503 | 0.9019 | 0.9463 | 1.3412 | 2.3432 | 0.0067 | 3.2703 | 0.9786 | 3.6955 | 4.6017 |
| Ties Merging + DPT | **0.4193** | **0.0100** | **0.7346** | **0.3731** | **0.5917** | **1.0442** | **1.0354** | **0.0055** | **2.1154** | **0.9291** | **1.1371** | **2.2277** |
| Sparse Merging | 0.9246 | 0.0185 | 1.1694 | 0.9610 | 0.6587 | 1.0873 | 1.9675 | 0.0075 | 3.1188 | 0.9519 | 1.7600 | 2.7035 |
| Sparse Merging + DPT | **0.4044** | **0.0091** | **0.7458** | **0.3706** | **0.5952** | **1.0352** | **1.0322** | **0.0055** | **2.1134** | **0.9305** | **1.1372** | **2.2206** |

**Table 2:** Performance comparison according to task-specific parameter allocation in DPT on the JRDB dataset with Sparse Merging. Sparse Merging is performed with a mask ratio of $K$=1%. A higher DPT allocation ratio leads to lower performance in the corresponding task.

| DPT allocation ratio | | Planning Metric | | | | Prediction Metric | |
|---|---|---|---|---|---|---|---|
| Plan (%) | Pred (%) | Effectiveness (ADE ↓) | Safety (Col. rate ↓) | Goal Success (FDE ↓) | Goal Success (Miss rate ↓) | ADE ↓ | FDE ↓ |
| 90 | 10 | 0.4899 | 0.0156 | 0.7748 | 0.3802 | 0.5777 | 1.0139 |
| 75 | 25 | 0.4352 | 0.0121 | 0.7515 | 0.3733 | **0.5724** | **1.0107** |
| 50 | 50 | 0.4044 | 0.0091 | 0.7458 | 0.3706 | 0.5952 | 1.0352 |
| 25 | 75 | 0.3983 | **0.0083** | 0.7449 | 0.3711 | 0.5972 | 1.0375 |
| 10 | 90 | **0.3898** | **0.0083** | **0.7349** | **0.3676** | 0.6046 | 1.0436 |

### 4.2 Experimental Results

**Applicability of DPT to Other Merging Methods.** Existing merging methods mainly focus on combining task vectors that are already fine-tuned, whereas DPT prevents Skill Conflict during task-specific training, i.e., in the process of forming task vectors for each task. Therefore, the proposed DPT framework can be effectively applied not only to our Sparse Merging but also to other model merging methods. As shown in Tab. 1, we conducted experiments comparing the performance before and after applying DPT to existing model merging methods, including Task Arithmetic [18], Ties Merging [65], and our Sparse Merging. Sparse merging is performed using the mask ratio of K=1%. *Plan Finetune* and *Pred Finetune* represent the performance of models fine-tuned on each respective task, and the fact that their performance remains largely unchanged even with DPT indicates that our method does not degrade the inherent capability of the model. Moreover, when merging is performed using DPT-based fine-tuned models, all merging methods exhibit overall performance improvements. This trend is consistently observed on both the JRDB and JTA datasets. These results indicate that DPT is broadly applicable regardless of the merging strategy or dataset domain.

**Table 3:** Performance comparison by *mask ratio K* under Sparse Merging. The experiment uses a 50:50 DPT allocation on JRDB.

| $K$ | Planning Metric | | | | Prediction Metric | |
|---|---|---|---|---|---|---|
| | Effectiveness (ADE ↓) | Safety (Col. rate ↓) | Goal Success (FDE ↓) | Goal Success (Miss rate ↓) | ADE ↓ | FDE ↓ |
| 1 | 0.4044 | **0.0091** | 0.7458 | 0.3706 | 0.5952 | 1.0352 |
| 2 | **0.4041** | 0.0096 | 0.7320 | **0.3697** | **0.5753** | **1.0174** |
| 5 | 0.4156 | 0.0100 | **0.7306** | 0.3722 | 0.5871 | 1.0322 |
| 10 | 0.4193 | 0.0100 | 0.7346 | 0.3730 | 0.5917 | 1.0442 |
| 20 | 0.4213 | 0.0101 | 0.7389 | 0.3750 | 0.5920 | 1.0507 |
| 40 | 0.4229 | 0.0101 | 0.7410 | 0.3747 | 0.5895 | 1.0505 |

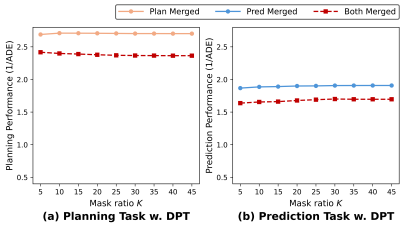


**Fig. 5:** Performance of (a) Planning and (b) Prediction tasks trained with DPT depends on *mask ratio K*.

**Task-Specific Parameter Allocation of DPT.** DPT performance is subject to variation based on the extent of task-specific parameter allocation. As shown in Tab. 2, we conducted experiments under the Sparse Merging setting, where the mask ratio is K=1%. We analyzed how the DPT allocation ratio, which defines the maximum range of learnable parameters during task-specific fine-tuning, affects performance. The results show that as the DPT allocation ratio increases for a particular task, its performance tends to decrease. This indicates that when the learnable parameter space is large, the task's skill becomes distributed across a wider range of parameters, requiring the use of more task vectors beyond the top 1%. This observation supports the finding discussed in Sec. 3.3, which suggests that in unified models, improving single-task performance requires leveraging a larger number of task vectors. In contrast, when the learnable region is smaller, DPT still achieves sufficient task performance through the Sparse Merging strategy, as it learns compactly from a localized parameter region. These results demonstrate that DPT effectively captures each task's skill in a compact manner, enabling it to alleviate *Skill Conflict* within the unified model.

**Effectiveness of Sparse Merging in DPT.** Even when applying DPT to completely separate the trainable parameter regions, interference can still occur during merging if the corresponding features are adjacent. Therefore, we conduct a performance comparison experiment of *Sparse Merging* according to the *mask ratio K*, as shown in Tab. 3. In this experiment, the maximum parameter allocation ratio for DPT is set to 50:50 by default, and evaluations are performed on the JRDB dataset. When the sparsity is high ($K$=2), the model achieves the best overall performance across all metrics, while performance gradually degraded as the task utilization ratio increased. This demonstrates that our Sparse Merging effectively minimizes interference between adjacent parameter regions during merging.

**Mitigating Skill Conflict through DPT.** To verify whether DPT effectively resolves the *Skill Conflict* observed in Fig. 2, we conduct the same experiments with DPT applied. As shown in Fig. 5, even when a large number of parameters are allocated to each task, DPT maintains high performance for both tasks, unlike the results in Fig. 2. Moreover, even with a smaller parameter allocation,

**Table 4:** Quantitative comparison between the proposed DPT with Sparse Merging framework and the baselines on the JRDB dataset. Sparse Merging is performed using a mask ratio of $K$=1%. We also compare the results obtained by applying joint reasoning within the existing framework. The comparison baselines are categorized into unified models with different architectures and methods applicable in parallel to our unified model, including ensemble and merging-based approaches. (SM: Sparse Merging; JR: Joint Reasoning)

| Method | Planning Metric | | | | Prediction Metric | |
|---|---|---|---|---|---|---|
| | Effectiveness (ADE ↓) | Safety (Col. rate ↓) | Goal Success (FDE ↓) | Goal Success (Miss rate ↓) | ADE ↓ | FDE ↓ |
| DIPP [17] | 0.8048 | 0.0167 | 1.3542 | 0.6940 | 0.8834 | 1.6406 |
| DTPP [15] | 0.6232 | 0.0145 | 1.1338 | 0.6931 | 0.7686 | 1.4253 |
| Ensemble [2] | 0.5807 | 0.0146 | 0.9000 | 0.4786 | 1.1313 | 1.5258 |
| Task Arithmetic [18] | 0.9387 | 0.0189 | 1.1798 | 0.8887 | 0.9904 | 1.3870 |
| Ties Merging [65] | 0.8759 | 0.0167 | 1.1503 | 0.9019 | 0.9463 | 1.3412 |
| Localize-and-stitch [13] | 0.8838 | 0.0176 | 1.1393 | 0.9487 | 0.8104 | 1.2670 |
| T-Switch [42] | 0.8047 | 0.0199 | 1.1009 | 0.8784 | 0.8091 | 1.2112 |
| DPT + SM | 0.4044 | 0.0091 | 0.7458 | 0.3706 | 0.5952 | 1.0352 |
| DPT + SM + JR | **0.3640** | **0.0074** | **0.7105** | **0.3589** | **0.5097** | **0.9396** |

DPT quickly achieves high single-task performance. This is because DPT begins training within a narrow parameter region and gradually expands the learnable region, allowing the model to encode each task's skill within a compact area. These results show that DPT effectively mitigates *Skill Conflict* and enables each task's capability to be represented efficiently using a small set of parameters.

**Comparison to Baselines.** Tab. 4 compares the proposed DPT and Sparse Merging framework with various baselines on the JRDB dataset. All experiments are conducted with a mask ratio of $K$=1% in Sparse Merging. Compared with other unified model architectures such as DIPP [17] and DTPP [15], our method achieves superior performance across planning and prediction metrics. This suggests that the task conflict between planning and prediction also exists in other unified model architectures, leading to suboptimal performance for both tasks. Among the merging baselines, the Ensemble method, which averages the inference results of each fine-tuned model, performs notably better than other merging strategies. This observation indicates that the conflict between planning and prediction primarily arises at the parameter level rather than only at the output level, supporting the existence of the previously identified *Skill Conflict*. Overall, the proposed DPT combined with Sparse Merging achieves the highest performance across all planning and prediction metrics on the JRDB dataset, demonstrating the effectiveness of our framework.

**Effect of Adding Joint Reasoning.** The DPT and Sparse Merging strategies reduce inter-task conflicts by separating the parameter regions used by each task. However, this separation may weaken the inherent advantage of unified models—namely, joint reasoning across tasks. To preserve the task-specific characteristics while still enabling joint reasoning, we perform an additional fine-tuning stage using a joint task loss on the parameter regions that remain inactive after

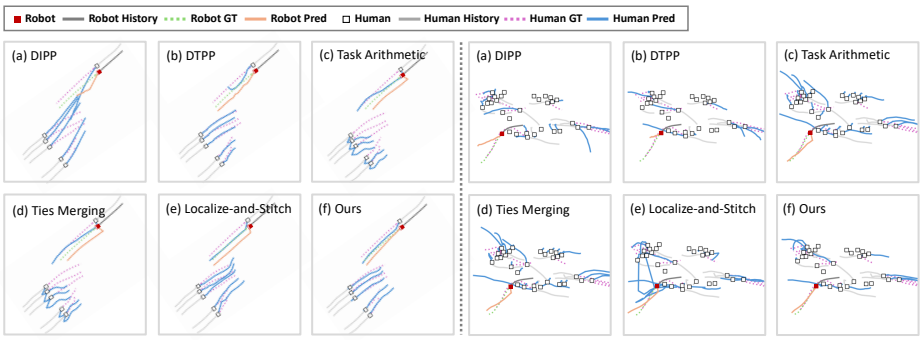


**Fig. 6:** Qualitative Results on the JRDB dataset. (a)–(b) show unified models with different architectures, while (c)–(e) present results of applying various baseline model merging methods on our backbone-based unified model. These are compared with (f) Ours, which employs DPT and Sparse Merging. In both examples, our method accurately predicts surrounding agents and produces safer plans compared to the baselines.

DPT and Sparse Merging. As shown in Tab. 4, models that incorporate this joint reasoning stage achieve better performance than those that only separate task-specific parameters. This highlights the importance of joint reasoning in unified models and demonstrates that, by isolating the core task-specific parameter regions while allowing the remaining parameters to support joint reasoning, our approach is more effective than conventional unified training schemes (DIPP [17], DTPP [15], Pretrained Model).

**Qualitative Results.** Fig. 6 compares qualitative results on the JRDB dataset. (a) and (b) show the results of other unified model architectures, DIPP and DTPP, while (c)–(e) present the results of applying different baseline model-merging methods on our unified-model backbone. These are compared with (f) Ours, which employs the proposed DPT and Sparse Merging approach. In the left sample of Fig. 6, both (a) DIPP and (b) DTPP exhibit generally poor prediction and planning performance for nearby agents, indicating that conventional unified models fail to effectively handle skill conflict between tasks. Meanwhile, in (e) Localize-and-Stitch, only a subset of key parameter regions from the fine-tuned models is added to the pretrained model. As a result, the behavior of the pretrained model—characterized by aggressive prediction and safe planning learned in a game-theoretic manner—remains distinctly reflected. Other model-merging baselines show similar tendencies, suggesting that existing merging methods fail to fully exploit the fine-tuned task skills due to skill conflict. In contrast, our approach—performing task-wise disjoint fine-tuning via DPT and applying Sparse Merging to minimize interference between adjacent features—achieves both accurate prediction and safe planning. This trend consistently appears even in dense scenes, as shown in the right sample of Fig. 6, demonstrating that our method effectively mitigates skill conflict in unified models and achieves strong performance across both tasks.

Additional analyses are provided in the supplementary material, including additional qualitative results in dense scenes (Suppl. Sec. 5), extended empirical studies of Skill Conflict across different architectures, model capacities, DPT settings and random seeds, controlled overlap, sub-task decomposition, mask evolution, and layer-wise conflict analysis (Suppl. Sec. 6), additional baseline comparisons with T-Switch, MTL baselines, and full-JTA results (Suppl. Sec. 7), training cost and inference efficiency analysis (Suppl. Sec. 8), and an extension to end-to-end autonomous driving with closed-loop and open-loop evaluation in the vehicle domain (Suppl. Sec. 9).

## 5 Conclusion

In this paper, we identify the Skill Conflict that arises within the shared encoder of multiple unified models through an empirical study, and propose two sequential methods—Disjoint Parameter Training (DPT) and Sparse Merging—to address it. DPT mitigates task interference by separating the learnable parameter regions of the prediction and planning tasks when constructing task-specialized material models. Sparse Merging performs merging using only the core task vectors to mitigate residual conflicts that may still occur in adjacent features, even when the main parameter regions are separated. This framework outperforms existing baselines and demonstrates the applicability of DPT to various merging methods. Overall, our results suggest that explicitly controlling parameter-level task interference is important for achieving accurate prediction and safe planning within compact unified models.

## Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. RS-2026-25494733 and No. RS-2025-22802992), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain, and No. RS-2025-02219277, AI Star Fellowship Support (DGIST)), the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2025-25420118), and the InnoCORE program of the Ministry of Science and ICT (No. N10260003 and No. 26-InnoCORE-01).

## References

1. Akbiyik, M.E., Savov, N., Paudel, D.P., Popovic, N., Vater, C., Hilliges, O., Gool, L.V., Wang, X.: Leveraging driver field-of-view for multimodal ego-trajectory prediction. In: The Thirteenth International Conference on Learning Representations (2025)

2. Arpit, D., Wang, H., Zhou, Y., Xiong, C.: Ensemble of averages: Improving model selection and boosting performance in domain generalization. Advances in Neural Information Processing Systems **35**, 8265–8277 (2022)
3. Bahari, M., Saadatnejad, S., Farsangi, A.A., Moosavi-Dezfooli, S.M., Alahi, A.: Certified human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12301–12311 (2025)
4. Bercovich, A., Dabbah, M., Puny, O., Galil, I., Geifman, A., Geifman, Y., Golan, I., Karpas, E., Levy, I., Moshe, Z., et al.: FFN fusion: Rethinking sequential computation in large language models. arXiv preprint arXiv:2503.18908 (2025)
5. Chen, K., Zhao, X., Huang, Y., Fang, G., Song, X., Wang, R., Wang, Z.: Social-MOIF: Multi-order intention fusion for pedestrian trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22465–22475 (2025)
6. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
7. Chen, X., Feng, S., Xiong, Z., An, S., Mao, Y., Yan, L., Tao, G., Guo, W., Zhang, X.: Temporal logic-based multi-vehicle backdoor attacks against offline RL agents in end-to-end autonomous driving. arXiv preprint arXiv:2509.16950 (2025)
8. Deep, P.T., Bhardwaj, R., Poria, S.: DELLA-Merging: Reducing interference in model merging through magnitude-based sampling. arXiv preprint arXiv:2406.11617 (2024)
9. Dong, Z., Ding, R., Li, W., Zhang, P., Tang, G., Guo, J.: Leveraging sd map to augment hd map-based trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17219–17228 (2025)
10. Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., Cucchiara, R.: Learning to detect and track visible and occluded body joints in a virtual world. In: Proceedings of the European conference on computer vision (ECCV). pp. 430–446 (2018)
11. Fang, Z., Hsu, D., Lee, G.H.: Neuralized markov random field for interaction-aware stochastic human trajectory prediction. In: The Thirteenth International Conference on Learning Representations (2025)
12. He, S., Cai, W., Huang, J., Li, A.: Capacity-aware inference: Mitigating the straggler effect in mixture of experts. arXiv preprint arXiv:2503.05066 (2025)
13. He, Y., Hu, Y., Lin, Y., Zhang, T., Zhao, H.: Localize-and-stitch: Efficient model merging via sparse task arithmetic. arXiv preprint arXiv:2408.13656 (2024)
14. Huang, W., Zhang, Y., Zheng, X., Chao, F., Ji, R., Cao, L.: Discovering important experts for mixture-of-experts models pruning through a theoretical perspective. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
15. Huang, Z., Karkus, P., Ivanovic, B., Chen, Y., Pavone, M., Lv, C.: DTPP: Differentiable joint conditional prediction and cost evaluation for tree policy planning in autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 6806–6812. IEEE (2024)
16. Huang, Z., Liu, H., Lv, C.: GameFormer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3903–3913 (2023)
17. Huang, Z., Liu, H., Wu, J., Lv, C.: Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. IEEE Transactions on Neural Networks and Learning Systems **35**(11), 15222–15236 (2024)

18. Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. arXiv preprint arXiv:2212.04089 (2022)
19. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407 (2018)
20. Jia, X., You, J., Zhang, Z., Yan, J.: DriveTransformer: Unified transformer for scalable end-to-end autonomous driving. In: The Thirteenth International Conference on Learning Representations (2025)
21. Jiang, B., Chen, S., Gao, H., Liao, B., Zhang, Q., Liu, W., Wang, X.: VADv2: End-to-end autonomous driving via probabilistic planning. In: The Fourteenth International Conference on Learning Representations (2026)
22. Jiang, H., Huang, X., Lu, Y., Wang, D., Cao, Y., Sha, C., Chen, B., Chen, K., Peng, X.: ExpertAD: Enhancing autonomous driving systems with mixture of experts. arXiv preprint arXiv:2511.11740 (2025)
23. Jiang, S., Liao, Y., Zhang, Y., Wang, Y., Wang, Y.: Fine-tuning with reserved majority for noise reduction. In: The Thirteenth International Conference on Learning Representations (2025)
24. Kedia, K., Dan, P., Choudhury, S.: A game-theoretic framework for joint forecasting and planning. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6773–6778. IEEE (2023)
25. Knoche, M., de Geus, D., Leibe, B.: Donut: A decoder-only model for trajectory prediction. arXiv preprint arXiv:2506.06854 (2025)
26. Lee, G., Jeong, W., Park, D., Jeong, J., Yoon, K.J.: Interaction-merged motion planning: Effectively leveraging diverse motion datasets for robust planning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28610–28621 (2025)
27. Lee, G., Park, D., Jeong, J., Yoon, K.J.: Non-differentiable reward optimization for diffusion-based autonomous motion planning. arXiv preprint arXiv:2507.12977 (2025)
28. Lee, Y., Jung, J., Baik, S.: Mitigating parameter interference in model merging via sharpness-aware fine-tuning. arXiv preprint arXiv:2504.14662 (2025)
29. Li, M., Nie, Z., Zhang, Y., Long, D., Zhang, R., Xie, P.: Improving general text embedding model: Tackling task conflict and data imbalance through model merging. arXiv preprint arXiv:2410.15035 (2024)
30. Li, P., Zheng, Y., Wang, Y., Wang, H., Zhao, H., Liu, J., Zhan, X., Zhan, K., Lang, X.: Discrete diffusion for reflective vision-language-action models in autonomous driving. arXiv preprint arXiv:2509.20109 (2025)
31. Li, R., Li, C., Lv, R., Li, Y., Gao, Y., Zhang, X., Zhou, J.: NATRA: Noise-agnostic framework for trajectory prediction with noisy observations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27872–27884 (2025)
32. Li, Y., Xiong, K., Guo, X., Li, F., Yan, S., Xu, G., Zhou, L., Chen, L., Sun, H., Wang, B., et al.: ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. arXiv preprint arXiv:2506.08052 (2025)
33. Li, Y., Ma, Y., Yan, S., Zhang, C., Liu, J., Lu, J., Xu, Z., Chen, M., Wang, M., Zhan, S., et al.: Model merging in pre-training of large language models. arXiv preprint arXiv:2505.12082 (2025)
34. Li, Z., Lin, Y., Gong, C., Wang, X., Liu, Q., Gong, J., Lu, C.: An ensemble learning framework for vehicle trajectory prediction in interactive scenarios. In: 2022 IEEE Intelligent Vehicles Symposium (IV). pp. 51–57. IEEE (2022)

35. Liu, T., Li, J., Zheng, Y., Niu, H., Lan, Y., Xu, X., Zhan, X.: Skill expansion and composition in parameter space. arXiv preprint arXiv:2502.05932 (2025)
36. Martin-Martin, R., Patel, M., Rezatofighi, H., Shenoi, A., Gwak, J., Frankel, E., Sadeghian, A., Savarese, S.: JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments. IEEE transactions on pattern analysis and machine intelligence **45**(6), 6748–6765 (2021)
37. Matena, M.S., Raffel, C.A.: Merging models with fisher-weighted averaging. Advances in Neural Information Processing Systems **35**, 17703–17716 (2022)
38. Messaoud, K., Cord, M., Alahi, A.: Towards generalizable trajectory prediction using dual-level representation learning and adaptive prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27564–27574 (2025)
39. Panariello, A., Marczak, D., Magistri, S., Porrello, A., Twardowski, B., Bagdanov, A.D., Calderara, S., van de Weijer, J.: Accurate and efficient low-rank model merging in core space. arXiv preprint arXiv:2509.17786 (2025)
40. Park, D., Surana, M., Desai, P., Mehta, A., John, R., Yoon, K.J.: Generative active learning for long-tail trajectory prediction via controllable diffusion model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27839–27850 (2025)
41. Pei, M., Shi, S., Chen, X., Liu, X., Shen, S.: Foresight in motion: Reinforcing trajectory prediction with reward heuristics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28303–28312 (2025)
42. Qi, B., Li, F., Wang, Z., Gao, J., Li, D., Ye, P., Zhou, B.: Less is more: Efficient model merging with binary task switch. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15265–15274 (2025)
43. Qiu, R., Gong, J., Zhang, X., Luo, S., Zhang, B., Cen, Y.: Adapting to observation length of trajectory prediction via contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1645–1654 (2025)
44. Rao, B., Liao, H., Guan, Y., Wang, C., Wang, B., Zhang, J., Li, Z.: AMD: Adaptive momentum and decoupled contrastive learning framework for robust long-tail trajectory prediction. arXiv preprint arXiv:2507.01801 (2025)
45. Ren, Z., Wei, P., Deng, S., Tang, H., Li, J., Li, H.: TOTP: Transferable online pedestrian trajectory prediction with temporal-adaptive mamba latent diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26263–26272 (2025)
46. Saadatnejad, S., Gao, Y., Messaoud, K., Alahi, A.: Social-Transmotion: Promptable human trajectory prediction. In: The Twelfth International Conference on Learning Representations (2024)
47. Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R.: Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16. pp. 414–430. Springer (2020)
48. Shang, S., Chen, Y., Wang, Y., Li, Y., Zhang, Z.: DriveDPO: Policy learning via safety dpo for end-to-end autonomous driving. arXiv preprint arXiv:2509.17940 (2025)
49. Shi, S., Jiang, L., Dai, D., Schiele, B.: Motion transformer with global intention localization and local movement refinement. Advances in Neural Information Processing Systems **35**, 6531–6543 (2022)

50. Su, H., Wu, W., Song, F., Zhang, J., Yang, Z., Yan, J.: DriveMamba: Task-centric scalable state space model for efficient end-to-end autonomous driving. arXiv preprint arXiv:2602.13301 (2026)
51. Sun, J., Li, Y., Chai, L., Lu, C.: Interactive adjustment for human trajectory prediction with individual feedback. In: The Thirteenth International Conference on Learning Representations (2025)
52. Sun, W., Li, Q., Wang, W., Geng, Y., Li, B.: Task arithmetic in trust region: A training-free model merging approach to navigate knowledge conflicts. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 5178–5187 (2025)
53. Sun, W., Li, Q., Wang, W., Liu, Y., Geng, Y.a., Li, B.: Towards minimizing feature drift in model merging: Layer-wise task vector fusion for adaptive knowledge integration. arXiv preprint arXiv:2505.23859 (2025)
54. Taketsugu, H., Oba, T., Maeda, T., Nobuhara, S., Ukita, N.: Physical plausibility-aware trajectory prediction via locomotion embodiment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12324–12334 (2025)
55. Tang, Y., Xu, Z., Meng, Z., Cheng, E.: HiP-AD: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 25605–25615 (October 2025)
56. Theus, A., Cabodi, A., Anagnostidis, S., Orvieto, A., Singh, S.P., Boeva, V.: Generalized linear mode connectivity for transformers. arXiv preprint arXiv:2506.22712 (2025)
57. Wang, H., Liang, W., Van Gool, L., Wang, W.: Dreamwalker: Mental planning for continuous vision-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10873–10883 (2023)
58. Wang, K., Dimitriadis, N., Favero, A., Ortiz-Jimenez, G., Fleuret, F., Frossard, P.: LiNeS: Post-training layer scaling prevents forgetting and enhances model merging. arXiv preprint arXiv:2410.17146 (2024)
59. Wen, D., Wang, Y., Wu, Z., He, Z., Wu, Z., Qingfang, Z.: A driving-style-adaptive framework for vehicle trajectory prediction. Advances in Neural Information Processing Systems **38**, 80138–80185 (2026)
60. Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: PARA-Drive: Parallelized architecture for real-time autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15449–15458 (2024)
61. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International conference on machine learning. pp. 23965–23998. PMLR (2022)
62. Xu, X., Kong, L., Shuai, H., Pan, L., Liu, Z., Liu, Q.: LiMoE: Mixture of LiDAR representation learners from automotive scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27368–27379 (2025)
63. Xu, Z., Yuan, K., Wang, H., Wang, Y., Song, M., Song, J.: Training-free pretrained model merging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5915–5925 (2024)
64. Xu, Z.: DAA*: Deep angular a star for image-based path planning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25284–25293 (2025)

65. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: TIES-Merging: Resolving interference when merging models. Advances in Neural Information Processing Systems **36**, 7093–7115 (2023)
66. Yang, B., Su, H., Gkanatsios, N., Ke, T.W., Jain, A., Schneider, J., Fragkiadaki, K.: Diffusion-ES: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15342–15353 (2024)
67. Yang, E., Shen, L., Guo, G., Wang, X., Cao, X., Zhang, J., Tao, D.: Model merging in LLMs, MLLMs, and beyond: Methods, theories, applications and opportunities. arXiv preprint arXiv:2408.07666 (2024)
68. Yang, E., Tang, A., Shen, L., Guo, G., Wang, X., Cao, X., Zhang, J.: Continual model merging without data: Dual projections for balancing stability and plasticity. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
69. Yang, J., Zhu, H., Wang, Y., Wu, G., He, T., Wang, L.: Tra-MoE: Learning trajectory prediction model from multiple domains for adaptive policy conditioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6960–6970 (2025)
70. Yang, J., Jin, D., Tang, A., Shen, L., Zhu, D., Chen, Z., Zhao, Z., Wang, D., Cui, Q., Zhang, Z., et al.: Mix data or merge models? balancing the helpfulness, honesty, and harmlessness of large language model via model merging. arXiv preprint arXiv:2502.06876 (2025)
71. Yang, P., Zheng, Y., Zhang, Q., Zhu, K., Xing, Z., Lin, Q., Liu, Y.F., Su, Z., Zhao, D.: UncAD: Towards safe end-to-end autonomous driving via online map uncertainty. arXiv preprint arXiv:2504.12826 (2025)
72. Yoshida, K., Naraki, Y., Horie, T., Yamaki, R., Shimizu, R., Saito, Y., McAuley, J., Naganuma, H.: Mastering task arithmetic: $\tau$Jp as a key indicator for weight disentanglement. In: The Thirteenth International Conference on Learning Representations (2025)
73. Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., He, Y.: Boosting continual learning of vision-language models via mixture-of-experts adapters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23219–23230 (2024)
74. Yu, L., Yu, B., Yu, H., Huang, F., Li, Y.: Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: Forty-first International Conference on Machine Learning (2024)
75. Yu, Y., Han, W., Wu, L., Liu, B., Wang, E., Zhang, Z.: Enduring, efficient and robust trajectory prediction attack in autonomous driving via optimization-driven multi-frame perturbation framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17229–17238 (2025)
76. Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. IEEE Access **8**, 58443–58469 (2020)
77. Zeng, F., Guo, H., Zhu, F., Shen, L., Tang, H.: RobustMerge: Parameter-efficient model merging for MLLMs with direction robustness. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
78. Zeng, W., Wang, S., Liao, R., Chen, Y., Yang, B., Urtasun, R.: DSDNet: Deep structured self-driving network. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16. pp. 156–172. Springer (2020)

79. Zhang, D., Liang, J., Guo, K., Lu, S., Wang, Q., Xiong, R., Miao, Z., Wang, Y.: CarPlanner: Consistent auto-regressive trajectory planning for large-scale reinforcement learning in autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17239–17248 (2025)
80. Zheng, Y., Liang, R., Zheng, K., Zheng, J., Mao, L., Li, J., Gu, W., Ai, R., Li, S.E., Zhan, X., et al.: Diffusion-based planning for autonomous driving with flexible guidance. arXiv preprint arXiv:2501.15564 (2025)
81. Zhong, Y., Feng, C., Yan, F., Liu, F., Zheng, L., Ma, L.: RoboTron-Nav: A unified framework for embodied navigation integrating perception, planning, and prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6416–6425 (2025)
82. Zhou, Y., Wu, X., Wu, J., Feng, L., Tan, K.C.: HM3: Hierarchical multi-objective model merging for pretrained models. arXiv preprint arXiv:2409.18893 (2024)
83. Zhou, Z., Cai, T., Zhao, S.Z., Zhang, Y., Huang, Z., Zhou, B., Ma, J.: AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. arXiv preprint arXiv:2506.13757 (2025)
84. Zhou, Z., Wang, J., Li, Y.H., Huang, Y.K.: Query-centric trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17863–17873 (2023)
85. Zhou, Z., Ye, L., Wang, J., Wu, K., Lu, K.: HiVT: Hierarchical vector transformer for multi-agent motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8823–8833 (2022)