Towards Generalizable Trajectory Prediction using Dual-Level Representation Learning and Adaptive Prompting

Supplementary Material

This supplementary document provides additional insights and experiments to complement the main paper. It includes detailed explanations of the Dynamic Weighted Aggregation (DWA) strategy and Prompt-Based Fine-Tuning techniques, along with implementation specifics to ensure consistency in evaluating baselines. Furthermore, we present experiments demonstrating the scalability of our model with increasing data size in multi-dataset training and comparing transfer learning strategies, such as prompt tuning and full fine-tuning, for cross-dataset adaptation.

7. Methodology Details

7.1. Dynamic Weighted Aggregation (DWA)

To balance the six diverse losses during training, we employ a **Dynamic Weighted Aggregation (DWA)** strategy. Each loss component, denoted as $\mathcal{L}_i$, is assigned a weight w_i that dynamically adjusts based on the relative difficulty of the task at the current stage of training. This approach ensures that harder tasks receive greater emphasis, enabling balanced optimization across all objectives.

The six losses are as follows:

- **Past Reconstruction Loss ($\mathcal{L}_{\text{past}}$):** Reconstruction loss for masked past trajectory points.
- **Future Reconstruction Loss ($\mathcal{L}_{\text{future}}$):** Reconstruction loss for masked future trajectory points.
- **Lane Reconstruction Loss ($\mathcal{L}_{\text{lane}}$):** Reconstruction loss for masked lane polyline points.
- **Prediction Loss ($\mathcal{L}_{\text{pred}}$):** Gaussian Mixture Model (GMM)-based loss to evaluate multimodal trajectory predictions.
- **Cross-Entropy SD Loss ($\mathcal{L}_{\text{cross-entropy}}$):** Aligns the student encoder's latent space with the teacher encoder's outputs.
- **KoLeo Regularization Loss ($\mathcal{L}_{\text{KoLeo}}$) [12]:** This regularizer encourages diversity and uniformity among feature representations within a batch. Given a set of n feature vectors $\{x_1, \ldots, x_n\}$, the regularizer is defined as:

$$\mathcal{L}_{\text{KoLeo}} = -\frac{1}{n}\sum_{i=1}^{n}\log(d_{n,i}), \tag{14}$$

where $d_{n,i} = \min_{j\neq i}\|x_i - x_j\|$ is the minimum distance between x_i and any other feature in the batch. To ensure consistency and stability, all feature vectors are ℓ_2-normalized before computing the regularization term.

Each task's loss is dynamically balanced using the DWA. The smoothed loss value $\tilde{L}_i$ for each task i is computed as:

$$\tilde{L}_i = 0.9\, L_i^{(t-1)} + 0.1\, L_i^{(t-2)}, \tag{15}$$

where $L_i^{(t-1)}$ and $L_i^{(t-2)}$ are the loss values from the previous and second-to-last iterations, respectively. The relative importance ratio r_i for task i is then calculated as:

$$r_i = \frac{\tilde{L}_i}{L_i^{(t-2)} + \epsilon}, \tag{16}$$

where ϵ is a small constant to avoid division by zero. Using these ratios, initial task weights w_i are computed as:

$$w_i = \frac{n\, r_i}{\sum_{j=1}^{n} r_j}, \tag{17}$$

where n is the total number of tasks.

To account for task-specific priorities, biases are applied to the weights, and the biased weights are clipped within predefined bounds $[w_{\text{min}}, w_{\text{max}}]$ to ensure stability. Finally, the weights are normalized again to ensure their sum equals the total number of tasks:

$$w_i = \frac{n\, w_i}{\sum_{j=1}^{n} w_j}. \tag{18}$$

7.2. Prompt-Based Fine-Tuning

Clustering and Prompt Selection During fine-tuning, the clustering head, trained during pretraining, assigns each input scene to a specific cluster. The clustering head outputs class logits, and the cluster is identified using the `argmax` operation:

$$\text{cluster\_id} = \arg\max(\text{MLP}(\mathbf{X})), \tag{19}$$

where $\mathbf{X}$ represents the input features, and the cluster with the highest probability is selected. Each cluster corresponds to a unique prompt sequence $\mathbf{p}_k$ from the prompt pool, where k denotes the cluster index.

Prompt Initialization The prompt pool $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K]$ consists of K learnable prompt sequences, one for each cluster. Each prompt sequence $\mathbf{p}_k$ is initialized using a uniform distribution:

$$\mathbf{p}_k \sim \mathcal{U}(-1, 1), \quad \forall k \in \{1, 2, \ldots, K\}. \tag{20}$$

This initialization ensures diversity across prompt sequences and avoids bias in the adaptation process.

Table 5. **# Registers and Clusters vs Performance on NuScenes**

	Number	B-FDE ↓	minADE ↓	minFDE ↓	MR ↓
Registers	0	2.71	0.98	2.06	0.37
	10	2.69	0.94	2.03	0.34
	26	2.67	0.94	2.02	0.34
	58	**2.62**	**0.93**	**1.97**	**0.32**
	74	2.67	**0.93**	2.03	0.34
Clusters	8	2.70	0.93	2.05	0.35
	16	2.68	**0.92**	2.03	0.35
	32	2.67	0.93	2.02	0.34
	64	**2.62**	0.93	**1.97**	**0.32**

Integration of Prompts with Queries After selecting the prompt sequence $\mathbf{p}_{\text{cluster\_id}}$ based on the assigned cluster, it is concatenated with the mode queries $\mathbf{Q}_{\text{modes}}$ and the register queries $\mathbf{Q}_{\text{reg}}$. The combined query representation $\mathbf{Q}_{\text{combined}}$ is expressed as:

$$\mathbf{Q}_{\text{combined}} = \text{Concat}(\mathbf{Q}_{\text{modes}}, \mathbf{Q}_{\text{reg}}, \mathbf{p}_{\text{cluster\_id}}), \tag{21}$$

where Concat denotes concatenation along the token dimension. This combined query serves as input to the frozen Perceiver decoder.

8. Implementation Details

In our implementation, the results for MTR [37] and AutoBot [17] are sourced directly from the UniTraj [14] paper for consistency and comparability. For Forecast-MAE [8], the only SSL-based approach with publicly available code, we adapted the implementation to the UniTraj framework and report the results based on our experiments.

9. Additional Experiments

9.1. Role of Register Queries.

As shown in Tab. 5, increasing the number of registers reduces B-FDE to **2.62** at 58 registers on NuScenes (NS). Unlike mode queries, which generate specific trajectories, register queries are *not directly tied* to specific outputs. Instead, they aggregate global and local context from agent trajectories and road graphs, acting as memory tokens that persist across decoder layers to refine contextual information. This enables diverse trajectory generation without explicit mode enumeration, *i.e.,* multi-modality. Other variants of registers [10] are known to improve performance in other tasks.

9.2. Prompt-Based Fine-Tuning.

In Tab. 5, we observe that increasing the number of clusters improves performance. *Why does it work?* During pretraining, a scene clustering head (MLP) groups similar inputs by aligning student-teacher scene vectors with cross-entropy and enhancing cluster separation with KoLeo loss. During fine-tuning, each cluster is assigned a *learnable* prompt, and each input's Scene Vector (SV_x) is assigned a cluster via: $p_{id} = \texttt{argmax}(\texttt{MLP}(SV_x))$. This prompt p_{id} is concatenated with decoder queries, guiding the model to make cluster-specific predictions while freezing the core architecture.

Figure 5. Failure Analysis (train on WOMD, eval on NS)

9.3. Compute and complexity.

Our PerReg model has 19.2M parameters, with 16.7M trainable. Due to SD and MR, it increases (pre-) training time (compared to training) by ∼22% but reduces inference time by ∼**18%** compared to (with NMS) on WOMD. Pre-Training on 4 × A100 GPUs with a batch size of 512 takes ∼30 minutes per epoch.

9.4. Failure Analysis.

In three cases (Fig. 5), low confidence in cluster assignment (3%) results in missed trajectories. As future work, we will explore alternative prompt aggregation methods, such as adaptive weighting.

9.5. Scalability on Multi-Dataset Training

To evaluate the performance of our model, PerReg+, on multi-dataset pretraining, we combine three datasets of varying sizes and test its performance using progressively larger subsets of the combined data. Specifically, we use 20%, 40%, 60%, 80%, and 100% of the total combined dataset, ensuring that each subset contains equal proportions from all three datasets. This setup allows us to assess how the model scales with increasing data availability while maintaining a balanced representation from each dataset. Performance is evaluated using standard trajectory prediction metrics across these varying dataset sizes.

Table 6 summarizes the results for each dataset and data size. As the size of the combined dataset increases, PerReg+ consistently improves its performance across all metrics, indicating effective utilization of additional data. Notably, the performance at 100% data outperforms the 20% subset by significant margins, particularly in B-FDE and minFDE, reflecting the model's ability to generalize with larger, balanced training data.

When compared to training on single datasets, multi-dataset training with 100% data results in better performance on nuscenes [5] and Argoverse 2 [42] across most metrics, highlighting the benefits of diverse domain exposure. For WOMD [13], results are comparable between the single-dataset and multi-dataset approaches, suggesting

Table 6. **Scalability and Performance Evaluation on Multi-Dataset Training.**PerReg+'s performance is evaluated on combined datasets using data sizes from 20% to 100%, with equal proportions from each, and compared to single-dataset training.

Data Size	nuscenes				Argoverse 2				WOMD			
	B-FDE ↓	minADE ↓	minFDE ↓	MR ↓	B-FDE ↓	minADE ↓	minFDE ↓	MR ↓	B-FDE ↓	minADE ↓	minFDE ↓	MR ↓
20%	2.43	0.88	1.80	0.28	2.31	0.86	1.71	0.27	2.28	0.86	1.65	0.26
40%	2.40	0.84	1.75	0.26	2.17	0.80	1.57	0.24	2.16	0.70	1.53	0.23
60%	2.33	0.81	1.69	0.25	2.09	0.76	1.48	0.21	2.11	0.68	1.49	0.22
80%	2.33	0.83	1.68	0.24	2.06	0.76	1.45	0.21	2.07	0.66	1.45	0.21
100%	**2.28**	**0.79**	**1.64**	**0.25**	**2.02**	**0.74**	**1.41**	**0.19**	**2.04**	**0.65**	**1.42**	**0.20**
Single dataset	2.62	0.93	1.97	0.32	2.07	0.77	1.46	0.21	2.05	**0.65**	**1.42**	**0.20**

Table 7. **Transfer Pre-training vs. Direct Pre-training.** Evaluation of different pretraining and fine-tuning strategies for the model on the nuscenes dataset. Strategies include Prompt Tuning (PT) on WOMD, nuscenes, and transfer (WOMD → nuscenes), as well as Full Fine-tuning.

Finetuning Strategy	Evaluation			
	B-FDE ↓	minADE ↓	minFDE ↓	MR ↓
PT (WOMD)	2.75	1.01	2.07	0.36
PT (nuscenes)	2.62	0.93	1.97	0.32
PT (WOMD → nuscenes)	2.53	0.94	1.89	0.32
Full (WOMD → nuscenes)	**2.27**	**0.79**	**1.64**	**0.27**

that the additional data maintains the model's strong performance. The evaluation demonstrates that multi-dataset training improves the generalization of PerReg+, particularly when leveraging the full combined dataset.

9.6. Transfer Pre-training

Table 7 compares various pretraining and fine-tuning strategies on the nuscenes dataset. In all experiments, the model is pretrained on the WOMD dataset and evaluated on nuscenes, except for the PT nuscenes strategy, where the model is both pretrained and fine-tuned exclusively on nuscenes. Prompt tuning on WOMD (PT WOMD) achieves a B-FDE of 2.75, but is outperformed by prompt tuning directly on nuscenes (PT nuscenes). Transfer learning with prompt tuning (PT WOMD → nuscenes) further improves B-FDE, demonstrating the benefits of leveraging large-scale WOMD pretraining. Full fine-tuning after transfer (Full WOMD → nuscenes) achieves the best results across all metrics, highlighting the advantages of fully adapting to the target domain. These results emphasize the importance of combining large-scale pretraining with domain-specific fine-tuning, where prompt tuning offers computational efficiency, and full fine-tuning maximizes performance.

9.7. Decoder Freezing.

Freezing the decoder (∼67% of model params) is favorable because it reduces compute cost. Remarkably, in single-dataset training, PerReg (frozen decoder + finetuned prompts) achieves similar results to decoder finetuning (B-FDE=2.62) despite using only a *fraction* of the compute cost. Moreover, in transfer pretraining (WOMD → NS), a learnable decoder further improves performance, yet the frozen decoder with fine-tuned prompts still *outperforms* single-dataset training. Prompt fine-tuning on a frozen decoder enables targeted adaptation, preserving structure and efficiency.