UniTraj: A Unified Framework for Scalable Vehicle Trajectory Prediction Supplementary Materials

Lan Feng*,[1], Mohammadhossein Bahari*,[1], Kaouther Messaoud Ben Amor[1], Éloi Zablocki[2], Matthieu Cord[2,3], and Alexandre Alahi[1]

[1] EPFL, Switzerland,
firstname.lastname@epfl.ch
[2] Valeo.ai, France,
[3] Sorbonne Université, France

In this supplementary document, we provide additional content to complement our main paper. This includes a set of experiments that further illuminate the capabilities of the UniTraj framework for the research community (see Section 1), additional results including cross-checking the performance of models and reporting other metrics (see Section 2), and more details about UniTraj framework (see Section 3).

1 Further experimentation with UniTraj

UniTraj is a flexible and comprehensive tool that opens up various research opportunities. While we have shown some of these in the main paper, we demonstrate other opportunities and capabilities of the framework here.

1.1 Continual learning on multiple datasets

Adapting trajectory prediction models to new datasets without erasing prior knowledge, termed catastrophic forgetting [4], is crucial for autonomous driving models. Our results in Table 1 illustrate this challenge, where we train AutoBot on 30k samples of WOMD (resp. Argoverse 2) dataset followed by fine-tuning on Argoverse 2 (resp. WOMD). Fine-tuning a WOMD-trained model with Argoverse 2 data modestly improves performance in Argoverse 2 (improvement +0.92), but significantly hurts its WOMD performance (-0.52). Similarly, fine-tuning an Argoverse 2-trained model with WOMD data notably boosts WOMD performance (+2.06) but severely impacts Argoverse 2 performance (-0.98). This is attributed to the considerable domain gap between WOMD and Argoverse 2, as shown in our cross-dataset generalization experiment. This outcome exemplifies the issue of catastrophic forgetting, highlighting the challenge of balancing adaptation to new datasets without compromising performance on previously learned ones.

Table 1: Continual learning results. This table presents outcomes from two methodologies: initial training on WOMD with subsequent fine-tuning on Argoverse 2, and initial training on Argoverse 2 followed by fine-tuning on WOMD. We report the brier-minFDE metric for the validation sets. The "Improvement" column indicates performance gain on the fine-tuning domain data, and the "Forgetting" column shows the performance drop on the training domain data.

Training domain	→	Finetuning domain	Improvement (finetun. dom.)	Forgetting (train. dom.)
WOMD	→	Argoverse 2	+ 0.92	- 0.52
Argoverse 2	→	WOMD	+ 2.06	- 0.98

Table 2: Generalization from synthetic data to real data. We train AutoBot with nuScenes or/and synthetic dataset, and evaluate on nuScenes.

Training	minADE6 (↓)	brier-minFDE (↓)
PG	8.36	14.35
nuScenes	**1.28**	**3.55**
nuscenes and PG	1.30	3.56

1.2 Synthetic data

Synthetic data serves as a cost-effective and versatile approach, particularly in autonomous driving, thanks to its ease of generation and capacity to simulate various, even rare, driving scenarios during training. In our approach, we leverage a large synthetic dataset created through Procedural Generation (PG) to pre-train our models. PG, a technique used in the MetaDrive simulator [5], facilitates the creation of varied traffic scenarios and maps based on predefined rules. In our experiment, we use the same setting of MetaDrive simulator; traffic density at 15 vehicles per 100 meters and incorporating 2 roadblocks in each scenario. This resulted in the creation of 30,000 unique scenarios.

We train AutoBot using the synthetic PG dataset and subsequently fine-tune it on the nuScenes dataset. The pre-training stage can be helpful for injecting basic behaviors in models, such as lane following and avoiding collisions. Table 2 shows the result of our experiment. The comparison between the first and second rows of the table reveals a notable domain gap between datasets. Specifically, the model trained on the PG dataset underperforms significantly on the nuScenes dataset. Moreover, the model does not show any substantial performance improvement after fine-tuning, compared to those that have not undergone pre-training. This outcome indicates that while synthetic data may potentially serve as a useful starting point for training and introducing models to a range of conditions, its effectiveness in improving real-world performance may be limited. This limitation is attributed to the domain gap between real and synthetic data, as well as the simplicity and limited diversity of the syn-

thetic data. We believe that more realistic and diverse synthetic datasets might be helpful as a pre-training source.

Table 3: Stratified evaluations per trajectory type. We report brier-minFDE for AutoBot [2] and MTR [9] models trained and evaluated on WOMD dataset.

Traj. Type	Stationary	Straight	Straight right	Straight left	Right u-turn	Right-turn	Left u-turn	Left-turn	All
AutoBot	1.50	2.21	2.77	2.69	8.06	2.99	4.32	2.69	2.47
MTR	1.09	2.13	2.86	2.90	5.96	2.83	4.58	2.64	2.12

Table 4: Fine-grained evaluation Kalman difficulty. We report the brier-minFDE metric across three chunks of Kalman difficulties. We compare AutoBot [2] and MTR [9] models trained and evaluated on WOMD dataset.

Kalman difficulty	Easy $\in [0, 30[$	Medium $\in [30, 60[$	Hard $\in [60, 100[$	All
AutoBot	2.52	2.46	3.52	2.47
MTR	2.05	2.40	2.45	2.12

1.3 Comparing different models' performances using fine-grained evaluations:

We compare the performance of multiple trajectory prediction models in Table 2 of the main paper. The table provides a comparison between different models. Specifically, it shows performances of AutoBot and MTR on the WOMD dataset, where MTR achieves a lower error rate of 2.37, outperforming AutoBot's 2.47. However, this is a course comparison without any details about the strengths and weaknesses of each model. Thanks to the fine-grained evaluation approach available in UniTraj, we provide a more detailed comparative analysis in Tables 3 and 4. The first table compares performances across different trajectory types. While MTR generally outperforms AutoBot, it falls behind in specific trajectories such as straight right, straight left, and left U-turns. The second table also reveals that AutoBot exceeds MTR in medium difficulty scenarios. These insights offer a more thorough comparison and help pinpoint specific areas of weakness in the models.

1.4 Other potential future directions

UniTraj framework paves the way toward building foundation models for trajectory forecasting. Foundation models are typically trained on vast datasets,

which, as of now, are not readily available in the trajectory forecasting domain. An effective workaround is to utilize extensive synthetic data for initial model training, followed by fine-tuning on real-world datasets. UniTraj facilitates this approach by enabling the integration of various synthetic datasets alongside its collection of the largest real data currently available in this field. This direction of research is already gaining traction and some recent studies in trajectory forecasting have adopted the foundation model concept by tokenizing the action space and focusing on predicting the subsequent token [7, 8].

The framework also opens up opportunities for other research studies such as coreset selection and dataset distillation [3, 10]. This involves creating a compact subset of the trajectory data that encapsulates the majority of information from the combined dataset.

2 Complementary results

In this section, we present additional results that complement those featured in the main paper. We first provide performances of the prediction models integrated into the framework compared with their officially reported performance to verify our correct integration. Then, we report metrics beyond brie-minFDE for the results we presented in the paper.

2.1 Cross-checking the performance of baselines with original papers

For MTR, we report numbers on WOMD. We trained MTR on WOMD and in the same setting as the official setting, e.g., 1 second of past trajectories and 8 seconds of future trajectories. Then, we evaluate the model on the official WOMD validation set, using the official evaluation tool to report the numbers. The results in Table 5 show that the integrated MTR can achieve similar performance compared to the original implementation.

Table 5: Performance evaluation of MTR in the WOMD setting compared to the original implementation.

	mAP	minADE	minFDE	MissRate
Vehicle	0.44	0.78	1.55	0.16
Pedestrian	0.43	0.35	0.73	0.07
Cyclist	0.36	0.72	1.45	0.19
Avg (ours)	0.41	0.62	1.24	0.14
Avg-original	0.42	0.60	1.23	0.14

Similarly, for Wayformer, we report numbers on WOMD with the same data setting as the official setting. Since the original paper has not reported numbers

on validation set, we compare our performance with MTR. The results in Table 6 show that our internal implementation of Wayformer outperforms MTR in terms of minADE and minFDE metrics. However, due to the absence of certain details from the Wayformer's non-public implementation, there remains a discrepancy in mAP metric performance.

Table 6: Performance evaluation of Wayformer in the WOMD setting.

	mAP	minADE	minFDE	MissRate
Vehicle	0.28	0.67	1.39	0.14
Pedestrian	0.25	0.32	0.67	0.09
Cyclist	0.24	0.68	1.40	0.21
Avg (ours)	0.26	0.56	1.15	0.15

For Autobot, please refer to our results on the nuScenes leaderboard in in Table 4 in the main paper.

2.2 Reporting other metrics

We extend our examination of model generalization capabilities, previously reported in the main paper using the brierFDE metric (Table 2), by presenting additional evaluation metrics: minFDE in Table 7, minADE in Table 8 and MR in Table 9. We report the metrics of the same experiments; training models on each dataset individually and evaluating their performance across all others.

Our expanded analysis confirms the observed trend of reduced model performance on unfamiliar datasets, a consistent result across the considered model architectures and datasets. It also reinforces the conclusions drawn in the main paper, illustrating the generalization hierarchy where WOMD-trained models outperform others, followed by Argoverse 2 and nuScenes.

We further confirm the benefit of training models on combined datasets, which aligns with the main paper's insights, showing notable performance improvements, especially for nuScenes.

Table 7: Cross-dataset generalization and multi-dataset training experiments minFDE metric.

		MTR [9]			Wayformer * [6]			AutoBot [2]		
		← **Evaluation** →								
↓ **Training**	#trajs	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD
nuScenes	32k	**2.33**	3.89	6.72	**2.50**	3.93	6.48	**2.62**	3.70	5.85
Argoverse 2	180k	3.10	**1.68**	4.04	3.07	**1.75**	4.12	3.52	**1.70**	3.59
WOMD	1800k	2.52	3.14	**1.78**	2.51	3.14	**1.46**	2.90	2.41	**1.65**
All	2012k	**1.81**	**1.61**	**1.78**	**1.76**	**1.51**	**1.45**	**2.24**	1.73	1.66

Table 8: Cross-dataset generalization and multi-dataset training experiments minADE metric.

		MTR [9]			Wayformer * [6]			AutoBot [2]		
		← **Evaluation** →								
↓ **Training**	#trajs	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD
nuScenes	32k	**1.06**	1.85	2.85	**1.04**	1.85	2.58	**1.21**	1.62	2.35
Argoverse 2	180k	1.42	**0.85**	1.73	1.44	**0.85**	1.65	1.59	**0.85**	1.43
WOMD	1800k	1.17	1.50	**0.78**	1.17	1.48	**0.65**	1.42	1.15	**0.73**
All	2012k	**0.85**	**0.82**	**0.78**	**0.84**	**0.76**	**0.65**	**1.12**	0.86	0.74

Table 9: Cross-dataset generalization and multi-dataset training experiments miss rate metric.

		MTR [9]			Wayformer * [6]			AutoBot [2]		
		← **Evaluation** →								
↓ **Training**	#trajs	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD	nuScenes	Argoverse 2	WOMD
nuScenes	32k	**0.41**	0.58	0.71	**0.42**	0.61	0.73	**0.40**	0.52	0.65
Argoverse 2	180k	0.47	**0.30**	0.59	0.48	**0.28**	0.61	0.49	**0.27**	0.55
WOMD	1800k	0.43	0.44	**0.22**	0.44	0.45	**0.25**	0.42	0.40	**0.25**
All	2012k	**0.32**	**0.28**	**0.33**	**0.27**	**0.23**	**0.22**	**0.36**	**0.27**	**0.25**

2.3 Experiments using pedestrian trajectories

While our study specifically focuses on vehicle trajectory prediction, the UniTraj framework still supports all types of traffic participants, including cyclists and pedestrians. To demonstrate this, we train Autobot to predict *pedestrian* trajectories in Table 10. Note that nuScenes is not considered as it does not officially support pedestrian trajectory prediction. Overall, similar findings can be made as with vehicles (Tab. 2).

Table 10: Cross-dataset generalization and multi-dataset training using pedestrian trajectories

AutoBot		brier-minFDE		minFDE		minADE		Miss Rate	
Train ↓	/ Val →	AV2	WOMD	AV2	WOMD	AV2	WOMD	AV2	WOMD
AV2		1.95	2.40	1.38	1.75	0.67	0.85	0.13	0.16
WOMD		2.22	1.92	1.51	1.28	0.75	0.57	0.16	0.14
All		1.82	1.92	1.24	1.28	0.64	0.57	0.11	0.14

3 More details about UniTraj framework

Training pipeline The training pipeline within UniTraj is underpinned by PyTorch Lightning [1]. PyTorch Lightning is a sophisticated deep-learning frame-

work that caters to the needs of AI researchers and machine-learning practitioners. In our implementation, we utilize PyTorch Lightning's training module in Distributed Data-Parallel (DDP) mode, enabling efficient multi-GPU acceleration for enhanced training speed and effectiveness.

Training settings In our experiments, we use 8 × A100 GPUs to train all the models. The batch sizes for MTR, Wayformer, and Autobot are 256, 256 and 128 respectively. The training takes approximately 1 hour, 15 minutes and 5 minutes per epoch for MTR, Wayformer, and Autobot, respectively. We pick the checkpoint based on the best min-brierFDE on the validation set.

References

1. William Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019. 6
2. Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *International Conference on Learning Representations*, 2022. 3, 5, 6
3. Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022. 4
4. James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1
5. Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022. 2
6. Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023. 5, 6
7. Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Learning the language of driving scenarios. *arXiv preprint arXiv:2312.04535*, 2023. 4
8. Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 4
9. Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. *Advances in Neural Information Processing Systems*, 35:6531–6543, 2022. 3, 5, 6
10. Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 4