PPT: Pretraining with Pseudo-Labeled Trajectories for Motion Forecasting

Yihong Xu[*1] **Yuan Yin**[*1] **Éloi Zablocki**[1]
Tuan-Hung Vu[1] **Alexandre Boulch**[1] **Matthieu Cord**[1,2]
[1]Valeo.ai, Paris, France [2]Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
{firstname.lastname}@valeo.com

Abstract: Accurately predicting how agents move in dynamic scenes is essential for safe autonomous driving. State-of-the-art motion forecasting models rely on large curated datasets with manually annotated or heavily post-processed trajectories. However, building these datasets is costly, generally manual, hard to scale, and lacks reproducibility. They also introduce domain gaps that limit generalization across environments. We introduce PPT (Pretraining with Pseudo-labeled Trajectories), a simple and scalable alternative that uses unprocessed and diverse trajectories automatically generated from off-the-shelf 3D detectors and tracking. Unlike traditional pipelines aiming for clean, single-label annotations, PPT embraces noise and diversity as useful signals for learning robust representations. With optional finetuning on a small amount of labeled data, models pretrained with PPT achieve strong performance across standard benchmarks particularly in low-data regimes, and in cross-domain, end-to-end and multi-class settings. PPT is easy to implement and improves generalization in motion forecasting. Code and data will be released upon acceptance.

Keywords: Motion forecasting, Pretraining, Pseudo-labeled trajectories

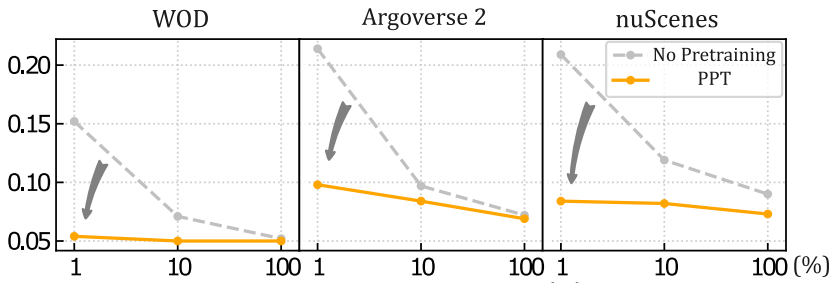


Figure 1: **No pretraining (gray) vs. Pretraining (orange)** with Pseudo-labeled Trajectories (PPT). PPT improves the motion forecasting performance (e.g., MissRate↓) through pretraining with pseudo-labeled trajectories, especially well in annotation-efficient regimes where only 1∼10% of the ground-truth labeled trajectories are used for finetuning.

1 Introduction

Anticipating how a scene will evolve is a key requirement in human-robot interaction scenarios, such as assisted and autonomous driving. Forecasting the future motions of nearby agents is critical for safe trajectory planning and avoiding collisions in these settings. To support progress in motion forecasting (MF), large-scale datasets like nuScenes (NUS) [1], Waymo Open Dataset (WOD) [2], and Argoverse 2 (AV2) [3] have become widely adopted. These datasets provide vectorized driving

[*]Equal contribution.

scenes with ground-truth agent trajectories, typically obtained through manual annotation and heavy post-processing [3, 4, 5]. This curated data has enabled strong supervised forecasting models [6, 7, 8].

Relying on curated datasets comes with significant limitations. First, dataset creation is expensive and labor-intensive, limiting scalability. Second, the curation pipelines are often opaque and dataset-specific, involving hand-tuned post-processing steps that are difficult to reproduce or apply to new datasets. Third, differences in how datasets are built can introduce domain gaps: models trained on one dataset often perform poorly on others [9, 10].

To address these challenges, we introduce PPT—Pretraining with Pseudo-labeled Trajectories—a simple and scalable alternative for motion forecasting. Instead of relying on curated labels, PPT pretrains models on pseudo-labeled data, followed by optional finetuning on a smaller amount of curated ground truth. PPT uses a fully automatic pipeline to generate pseudo-labeled trajectories. We apply off-the-shelf 3D object detectors [11, 12, 13, 14, 15, 16] to estimate agent positions in individual frames, and then apply a lightweight, non-learning tracker [17, 16] to associate these detections over time. The resulting pseudo-labeled trajectories exhibit two key properties: they are noisy and diverse. Indeed, compared to curated annotations, pseudo-labeled trajectories are less precise, reflecting the realistic imperfections of detector outputs and simple tracking. Moreover, instead of producing a single 'ground-truth' trajectory per agent, we aggregate multiple pseudo-trajectories from different detectors. Unlike traditional dataset annotation pipelines, aiming to produce clean, unique annotations per agent [5, 4], PPT embraces noise and diversity. While these traits are usually viewed as weaknesses, we show that they are surprisingly useful for pretraining. The variability acts as a regularizer, encouraging models to learn robust and generalizable representations.

A major advantage of PPT is its flexibility. Because our pseudo-labeling pipeline is fully automated and dataset-agnostic, we can combine data from multiple sources without manual harmonization. This enables large-scale pretraining that enhances generalization—a trend also observed in recent works [9]. Moreover, while most motion forecasting models are trained from scratch, PPT provides a valuable initialization. It allows models to first learn general forecasting dynamics from noisy and unlabeled data, and then specialize to any target domain with limited labeled data.

In summary, our contributions are as follows:

- We introduce PPT, a simple and novel pretraining approach for motion forecasting with diverse pseudo-labeled trajectories from multiple off-the-shelf detectors and a non-learning tracker.
- PPT significantly reduces the need for curated annotations, and is especially effective in annotation-efficient regimes, where only 1∼10% of the ground-truth labeled trajectories are used for finetuning, as shown in Fig. 1.
- PPT improves generalization across domains, including strong performance on out-of-distribution scenarios, end-to-end and multi-class motion forecasting settings.

2 Related Work

Motion forecasting consists of predicting the future trajectory for the agents of interest (e.g. vehicles or 10 different classes in AV2 MF [5]) based on their past trajectories, agent interactions, and map information. Conventional motion prediction methods [18, 19, 6, 20, 8, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31] focus on architectural design to improve trajectory prediction and assume that past trajectories are obtained from curated datasets with human annotations [2, 1, 3]. In contrast, end-to-end forecasting methods [10, 32, 33, 34] focus on the real-world deployment condition, where past trajectories are estimated from a perception pipeline, consisting of detection and tracking. However, as noted in [10], current end-to-end *trained* models [33, 32] exhibit lower performance compared to modular approaches [34], and effective strategies to address perception model imperfections remain unclear. In this work, we aim to propose a simple and powerful pretraining framework that improves both paradigms with diverse pseudo-labeled trajectories and minimum annotation cost.

Motion forecasting pretraining. Current motion forecasting heavily relies on the curated annotations provided by the datasets and trains the models from scratch. Recently, several self-supervised

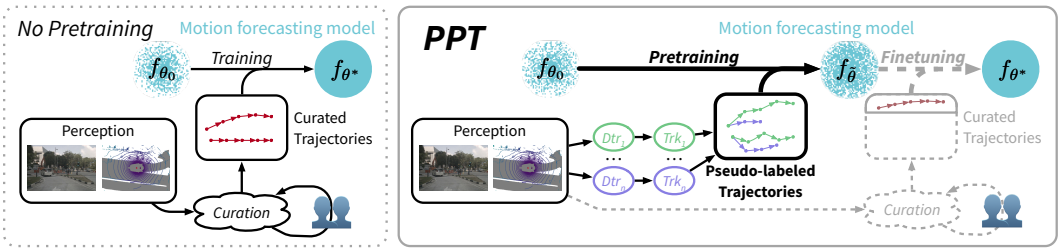


Figure 2: **Illustration of PPT.** On the left, we show the conventional approach to training a motion forecasting model from scratch with annotated labels, often from human curation. By contrast, on the right, we present a pretraining pipeline with pseudo-labeled trajectories from different 3D detectors and (non-learning) trackers (*Dtr*, *Trk*), followed by an optional finetuning phase (shaded in gray).

methods for motion forecasting have emerged [35, 36, 37]. Specifically, Xu et al. [35] pretrain map and trajectory encoders through contrastive learning, encouraging feature similarity between positive trajectory-map pairs. Bhattacharyya et al. [37] define several pretext tasks, such as lane masking, lane node distance prediction, and maneuver classification of agents of interest. Cheng et al. [36] mask the lane and past/future trajectories and train a network to reconstruct the masked values. Notably, these pretraining methods *still rely on curated annotations* and generally improve features of a sub-module (encoders) through contrastive learning or masking. Yang et al. [38] pretrain a perception backbone using images for various downstream tasks but focus only on single-dataset settings. Instead, we position our proposed strategy as *task-specific pretraining*, which performs pretraining using pseudo-labeled data with the same optimization objective of motion forecasting. Our goal is to explore diverse off-the-shelf trajectories for motion forecasting pretraining without directly using any annotated trajectories.

Pseudo-labeling. Traditionally, driving datasets [3, 1, 2] rely heavily on human annotation processes to generate agent trajectory labels, a costly approach that limits scalability. To address this, recent motion forecasting datasets such as WOMD [4] and AV2 MF [5] use automatic annotation pipelines coupled with complex post-processing steps to generate high-quality, near-human single-label trajectories. Instead, PPT embraces the raw tracking trajectories with various camera- [39, 14], LiDAR- [16, 11, 12, 13] or LiDAR-camera-based [40, 15] 3D detectors and a *non-learnable* tracker based on geometry cues [17, 16]. We note that the pseudo-labeling in PPT differs fundamentally from WOMD and AV2 MF in three key regards: (1) we do not aim to achieve clean labels, but embrace noisy and diverse trajectories for MF pretraining. (2) No post-processing like track smoothing [5, 4] is performed to reduce noise. (3) No manual selection of a single trajectory per agent discarding the trajectory diversity, which we show important for pretraining. (4) Moreover, with diverse pseudo labels, PPT exhibits higher generalization ability across domains (Sec. C.4).

3 PPT Framework

3.1 Conventional Training for Motion Forecasting

Motion forecasting models are typically trained on curated datasets composed of human-annotated trajectories within a specific domain. We denote such a dataset as $\mathcal{T} = \{\tau_i\}_i$, where τ_i is a trajectory of an object i (in a given scene). The trajectory is defined by temporally ordered state vectors $s_{i,t}$, i.e., $\tau_i = (s_{i,t})_t$. Each state $s_{i,t}$ encapsulates the attributes of the object (e.g., dimensions, heading) and center-point.

We consider a motion forecasting model $f_\theta \colon s_{i,t-L:t} \mapsto s_{i,t+1:t+M}$, mapping L past states and the current state of an object to its M future states, parameterized by model weights θ. Other contextual information in the scene, i.e., other agents' past trajectories and the map, can also be included in the input. We omit it for simplicity. The model is optimized over the curated trajectory set $\mathcal{T}$ as follows:

$$\theta^* = \arg\min_\theta \mathcal{L}(f_\theta; \mathcal{T}), \tag{1}$$

where $\mathcal{L}(f_\theta; \mathcal{T}) = \mathbb{E}_{\tau_i \sim p(\mathcal{T})} \mathbb{E}_{t \sim \mathcal{U}\{L, \text{len}(\tau_i) - M\}} \ell(f_\theta(s_{i,t-L:t}), s_{i,t+1:t+M})$ is the prediction cost computed over the trajectory set. A commonly used loss ℓ is the Average Distance Error (ADE):

$$\ell(f_\theta(s_{i,t-L:t}), s_{i,t+1:t+M}) = \frac{1}{M} \sum_{k=1}^{M} \| f(s_{i,t-L:t})_m - s_{i,t+m} \|_2^2. \tag{2}$$

We notice that optimizing Eq. 1 relies solely on the curated trajectory set $\mathcal{T}$, which implies a dependence on its curation specificity.

3.2 PPT Training Strategy via Pseudo-Labeling

To reduce dependence on curated, human-annotated trajectories, we propose a novel approach—PPT—to leverage unannotated perception data and optionally curated trajectories. PPT uses perception data by generating raw, non-curated trajectories for pretraining motion forecasting models. We detail the PPT strategy in the following and illustrate it in Fig. 2.

Pseudo-labeling via non-learning tracking. We assume access to a perception dataset $\mathcal{D} = \{(o_t)_t\}$, consisting of temporally ordered sensor data in different scenes at each time step t. Our goal is to construct a pseudo-labeled trajectory set $\tilde{\mathcal{T}}$ directly from this perception dataset $\mathcal{D}$. To achieve this, PPT leverages 3D multi-object tracking (MOT) methods to generate the pseudo labels from the sensor data sequence $(o_t)_t$. MOT methods typically follow the Tracking-by-Detection paradigm: At a given time t, a detector $Dtr\colon o_t \mapsto \delta_t$ processes the sensor data o_t to produce a set of *detected objects* $\delta_t = \{\tilde{s}_{k,t}\}_k$, each defined as the state vector mentioned above. We then initialize each track i by assigning it an initial state $s_{i,0} \in \delta_0$. The tracker iteratively extends the trajectory by searching for the most likely object in the detected-object set δ_t at the next time step, i.e., $Trk\colon (s_{i,t-1}, \delta_t) \mapsto s_{i,t}$. Current 3D object-tracking methods—like AB3DMOT [17] and others—typically rely on a non-learned tracker for *Trk*, that links past tracks to detections via a one-to-one assignment. Consequently, these methods fundamentally operate as a form of pseudo-labeling for agent trajectories, since they have not been trained to *predict* temporal waypoints. After processing each scene in the perception dataset $\mathcal{D}$, we obtain a set of pseudo-labeled trajectories $\tilde{\mathcal{T}}$.

Pretraining with pseudo-labeled trajectories. Using the pseudo-labeling method described above, we can generate a variety of pseudo-labeled trajectory sets for any dataset, given a detector and tracker. Although these trajectories may not exactly match the curated ones, they still serve well for pretraining. We provide a detailed analysis of their quality in App. C, covering the datasets and detectors used in our experiments. Moreover, thanks to the abundance of available perception datasets, detectors, and trackers, it is convenient to scale this method to produce diverse trajectories with different domains, noise levels, and other variations beyond the curated ground-truth annotations. To perform pretraining, we simply optimize Eq. 1 over our pseudo-labeled trajectory set $\tilde{\mathcal{T}}$. That gives us pretrained weights $\tilde{\theta}$ that capture the distribution of those pseudo-labeled trajectories.

Finetuning on human-annotated data. From there, one can optionally use the pretrained model as an initialization when training on the actual annotated data, such that it adapts to the specificity of the target domain for predictions. The finetuning still follows the same optimization in Eq. 1 but on a curated trajectory subset in a chosen domain $\mathcal{T}' \subseteq \mathcal{T}$. This subset can be smaller than the original curated set. Note that the finetuning is conducted independently of the data domains encountered during pretraining, enhancing the flexibility of the PPT strategy.

4 Experimental Results

We evaluate PPT across a range of settings. Sec. 4.1 introduces the experimental setup (with details in App. A). We compare PPT with a 'No-Pretraining' baseline under different annotation budgets in Sec. 4.2, and assess its generalization in cross-domain, end-to-end, and multi-class settings in Sec. 4.3 and, Sec. 4.4 examines scalability. Further analyses are in Sec. 4.5 and qualitative results in App. D.

Table 1: **PPT vs. No Pretraining.** Pretraining with PPT consistently improves performance, especially with limited labeled data. Models are either trained from scratch ("No Pretraining") or initialized with PPT. All models are then trained on 1%, 10%, or 100% of labeled data. All evaluations are on the full labeled valset. Relative improvements (%, teal) are shown over "No Pretraining".

% labeled trajs.	MTR [6]	*WOD* [2] Brier-FDE↓	*WOD* [2] min-ADE↓	*WOD* [2] min-FDE↓	*WOD* [2] Miss-Rate↓	*AV2* [3] Brier-FDE↓	*AV2* [3] min-ADE↓	*AV2* [3] min-FDE↓	*AV2* [3] Miss-Rate↓	*NUS* [1] Brier-FDE↓	*NUS* [1] min-ADE↓	*NUS* [1] min-FDE↓	*NUS* [1] Miss-Rate↓
1%	No Pretraining	5.358	1.905	5.154	0.152	6.258	1.560	5.768	0.214	6.039	2.411	5.764	0.209
	PPT (Ours)	**0.585**	**0.217**	**0.438**	**0.054**	**1.181**	**0.399**	**0.913**	**0.098**	**0.914**	**0.339**	**0.707**	**0.084**
		-89%	-89%	-92%	-64%	-81%	-74%	-84%	-54%	-85%	-86%	-88%	-60%
10%	No Pretraining	0.800	0.279	0.644	0.071	1.084	0.387	0.834	0.097	1.310	0.472	1.109	0.119
	PPT (Ours)	**0.563**	**0.209**	**0.419**	**0.050**	**1.044**	**0.334**	**0.756**	**0.084**	**0.856**	**0.325**	**0.650**	**0.082**
		-30%	-25%	-35%	-30%	-4%	-14%	-9%	-13%	-35%	-31%	-41%	-31%
100%	No Pretraining	0.562	0.219	0.416	0.052	0.905	0.307	0.644	0.072	0.923	0.327	0.722	0.090
	PPT (Ours)	**0.546**	**0.205**	**0.401**	**0.050**	**0.894**	**0.291**	**0.624**	**0.069**	**0.776**	**0.298**	**0.583**	**0.073**
		-3%	-6%	-4%	-4%	-1%	-5%	-3%	-4%	-16%	-9%	-19%	-19%

4.1 Experimental Setting

Datasets. We use three standard *perception* datasets[2] with both raw sensors (for pseudo-labeling) and trajectory data (for baselines and finetuning) available. Namely, nuScenes (NUS) [1], Waymo Open Dataset (WOD) [2] and Argoverse 2 (AV2) [3].

Pseudo-labeled trajectories. We collect detections from 9 state-of-the-art detectors and their variants [40, 16, 41, 40, 15, 11, 12, 13] and employ non-learning tracking methods [16, 17] for detection association, which form 8.6M pseudo-labeled trajectories for PPT pretraining. With further specification, we focus on the 4-wheel vehicle objects. Detailed statistics, their forecasting quality and, the impact of pseudo-labeling quality to motion forecasting performance are provided in App. C.

Forecasting method. To conduct experiments on different datasets, we leverage UniTraj [9], which provides a reliable implementation of multiple models, such as MTR [6], Wayformer [8], Autobot [42], that predict 6-second future trajectories from 2-second past ground-truth history at 10 Hz. Our main results use MTR; additional results for Wayformer and Autobot are provided in App. E.

Evaluation metrics. We evaluate the predicted trajectories with respect to the ground-truth future trajectories with: • minADE_k: The minimum Average Displacement Error over k predictions, computed as the mean L^2 distance between each predicted trajectory location and the ground-truth one. • minFDE_k: The minimum Final Displacement Error over k predictions, which measures the L^2 distance between the predicted endpoint and the ground truth. • Brier-FDE: The sum of minFDE_k and $(1-\delta)^2$, where δ represents the forecasting confidence score, quantifying the model's confidence in the trajectory prediction that most closely matches the ground truth. • $\text{MissRate}_{k@x}$: The proportion of forecasts where minFDE_k exceeds a threshold x, set to 2 meters. We set $k = 6$ for all metrics.

4.2 Annotation Efficiency

In the section, we show that PPT generally improves the baselines via the pretraining strategy with diverse off-the-shelf trajectories. This performance gap is especially pronounced when only a small fraction of curated data is available, demonstrating the strong annotation efficiency of PPT.

PPT vs. No Pretraining in different labeled data regimes. Tab. 1 reports the results of single-dataset training where the training and validation are done on the same dataset. Models trained from scratch ('No Pretraining') consistently underperform compared to those initialized with PPT, especially in low-label regimes (1%, 10%). Notably, models pretrained with PPT and finetuned on

[2]WOMD [4] and AV2 MF [5] datasets are omitted in our experiments due to the lack of off-the-shelf detectors and full perception data for the pseudo label extraction. We however show results on such datasets.

Table 2: **PPT *w/o* finetuning for adaptation.** Pretraining only on in-domain pseudo labels outperforms training on human-labeled data from a different domain.

Target		Source	Brier-FDE↓	min-ADE↓	min-FDE↓	Miss-Rate↓
WOD	No Pretraining	AV2	0.886	0.318	0.718	0.063
		NUS	1.539	0.577	1.362	0.093
	PPT (Ours)	pseudo WOD	**0.586**	**0.229**	**0.449**	**0.055**
AV2	No Pretraining	WOD	**1.302**	0.441	1.007	0.118
		NUS	1.483	0.529	1.205	0.124
	PPT (Ours)	pseudo AV2	1.311	**0.430**	**0.995**	**0.102**
NUS	No Pretraining	WOD	1.427	0.625	1.152	0.150
		AV2	1.128	0.557	0.882	0.105
	PPT (Ours)	pseudo NUS	**1.003**	**0.356**	**0.793**	**0.089**

Table 3: **Cross-domain generalization.** PPT pretraining with diverse pseudo-labeled trajectories improves cross-domain generalization.

Target		Pre-train	Fine-tune	Brier-FDE↓	min-ADE↓	min-FDE↓	Miss-Rate↓
WOD	No Pretraining	×	AV2	0.886	0.318	0.718	0.063
	PPT (Ours)	pseudo AV2		**0.798**	**0.310**	**0.611**	**0.053**
	No Pretraining	×	NUS	1.539	0.577	1.362	0.093
	PPT (Ours)	pseudo NUS		**0.988**	**0.371**	**0.836**	**0.079**
AV2	No Pretraining	×	WOD	**1.302**	**0.441**	**1.007**	0.118
	PPT (Ours)	pseudo WOD		1.363	0.445	1.074	**0.109**
	No Pretraining	×	NUS	1.483	0.529	1.205	0.124
	PPT (Ours)	pseudo NUS		**1.453**	**0.485**	**1.182**	**0.114**
NUS	No Pretraining	×	WOD	1.427	0.625	1.152	0.150
	PPT (Ours)	pseudo WOD		**1.207**	**0.458**	**0.959**	**0.102**
	No Pretraining	×	AV2	1.128	0.557	0.882	0.105
	PPT (Ours)	pseudo AV2		**1.071**	**0.470**	**0.822**	**0.093**

just 10% of labeled data match or surpass models trained from scratch on 100% of labeled data. In the extreme case of using only 1% of labeled trajectories, the performance gap becomes even larger.

PPT *w/o* finetuning for adaptation. We also evaluate models trained *only* with pseudo-labeled data, without any supervised finetuning. As shown in Tab. 2, pretraining on in-domain pseudo-labeled trajectories outperforms models trained on human-labeled but out-of-domain datasets. This result is particularly useful for practitioners working with new domains without labeled data. In such cases, PPT offers a compelling solution: train directly on pseudo labels from the target domain instead of transferring from unrelated labeled data, which often suffers from domain gaps.

4.3 Improved Generalization with PPT

Cross-domain generalization with PPT. Pretraining on diverse (Fig. 3) pseudo-labeled trajectories from a single dataset leads to better generalization to unseen domains. As shown in Tab. 3, models pretrained with PPT on pseudo-labeled data from a source dataset outperform those trained from scratch when evaluated on a different target dataset. For instance, when we pretrain a model using pseudo-labeled trajectories from the NUS dataset (via PPT), then finetune it on the human-annotated labels of NUS, and finally evaluate it on the WOD dataset, it outperforms a model that was trained only on the labeled NUS data. We attribute this to the noise and diversity of pseudo-labeled trajectories, which act as regularization and expose the model to a broader distribution of behaviors, ultimately improving robustness and generalization.

Toward end-to-end forecasting with PPT. End-to-end (E2E) forecasting uses *predicted* past trajectories—generated by a perception model—as input, reflecting real-world deployment where clean, labeled inputs are unavailable [43, 10, 34]. To jointly evaluate perception and forecasting, the metric mAP_f [43] is used, similar to the detection AP, matching predicted future trajectories to ground truth. We integrate PPT into the winning solution [34] of the AV2 E2E forecasting challenge, training the model either on labeled AV2 data (No pretraining) or solely on pseudo-labeled AV2 data [40] (PPT without finetuning). As shown in Tab. 4, the model trained with PPT outperforms its supervised

Table 4: **E2E forecasting on AV2.** With PPT, the end-to-end forecasting model [34] becomes more robust against imperfect perception inputs in the end-to-end forecasting paradigm. Teal shows relative gains (%) over 'No Pretraining'.

Modular4Cast [34]	Source	AV2 E2E MF mAP_f ↑	minADE↓	minFDE↓
No Pre-training	AV2	0.367	2.868	4.807
PPT (Ours)	pseudo AV2	**0.595**	**0.874**	**1.430**
		+62.13%	-69.53%	-70.25%

Table 5: **Multi-class motion forecasting.** PPT boosts performance on the AV2 MF [5] multi-class benchmark. Evaluated on the testset via the official server. Teal shows relative gains (%) over 'No Pretraining'.

	AV2 MF Testset Brier-FDE↓	minADE↓	minFDE↓	MissRate↓
No Pretraining	2.282	0.832	1.658	0.286
PPT (Ours)	**2.246**	**0.820**	**1.630**	**0.272**
	-1.58%	-1.44%	-1.69%	-4.90%

counterpart, despite using no ground-truth labels. This highlights PPT as a promising strategy for improving forecasting robustness in E2E settings with the presence of imperfect perception.

Toward multi-class motion forecasting with PPT. We extend our evaluation to a multi-class setting, predicting trajectories for 10 distinct agent types as defined in the AV2 Motion Forecasting (AV2 MF) benchmark [5]: `VEHICLE`, `PEDESTRIAN`, `MOTORCYCLIST`, `CYCLIST`, `BUS`, `BACKGROUND`, `STATIC`, `CONSTRUCTION`, `RIDERLESS_BICYCLE` and `OTHER`. To do this, we pretrain the MTR model [6] with PPT on pseudo-labeled data of the perception datasets [1, 3, 2], including all classes, and then finetune it on the AV2 MF [5] training set. We evaluate by submitting predictions to the AV2 MF test server. As shown in Tab. 5, pretraining with PPT on pseudo-labeled perception datasets also improves performances in standard motion forecasting benchmarks in the multi-class setting.

4.4 Scaling Pretraining and Finetuning

We study how performance changes when increasing the amount of pretraining or finetuning data. We note that we use the *same* amount of data (i.e., the same number of agents) in pseudo-labeled trajectories and their ground-truth counterparts, but the pseudo-labeled trajectories are extracted from different detectors, automatically creating diverse and realistic training examples. In the setting noted 'Single', the pretraining or finetuning is done on the same labeled data as the evaluation. The model is then evaluated on the same dataset, and we report the average performance across the three datasets. In the setting noted 'All', we combine the three motion forecasting datasets [1, 3, 2] for the pretraining or finetuning. The evaluation is then performed on the respective evaluation sets of these datasets, and the average performance is reported.

Pretraining on more pseudo-labeled data. A key advantage of PPT is that pseudo-labeled trajectories generated from different datasets are naturally compatible: no manual annotation, alignment, or post-processing is required. This makes it straightforward to combine data from multiple sources, something that is far more challenging with curated datasets due to differing formats and annotation protocols [9]. As shown in Tab. 6, pretraining on the combined dataset ('All') consistently outperforms pretraining on any single dataset. This scaling effect highlights the value of large and diverse pseudo-labeled data in improving motion forecasting performance, and further increases the gap between models trained from scratch and those using PPT.

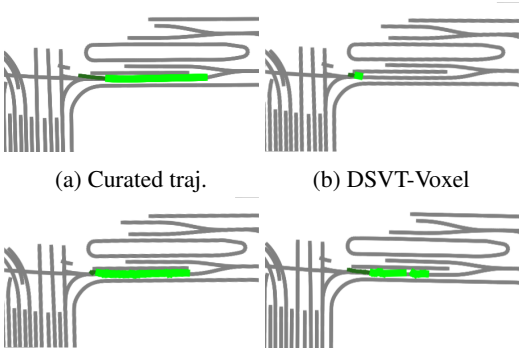
Finetuning on more labeled data. While PPT consistently outperforms the baseline when both pretraining and finetuning use the combined datasets ('All'), increasing the amount of labeled data for finetuning does not always yield additional gains compared to using only a 'Single' dataset for finetuning. We hypothesize that the model already captures strong motion priors from diverse pseudo-labeled trajectories during the pretraining phase. Finetuning then mainly serves as a lightweight adaptation step, helping the model align with human-curated labels. This is supported by our analysis in App. B, where we observe that performance often plateaus after only 1–2 epochs of finetuning.

Table 6: **Scaling pretraining and finetuning.** Comparison of models trained from scratch ('No Pretraining') versus those using PPT pretraining on a single dataset ('Single') or all three combined ('All'). We report the average evaluation metrics computed separately on each dataset. Relative gains (%, in teal) are given against the 'No Pretraining' baseline.

	Pretrain	Finetune	*Average* Brier-FDE↓	minADE↓	minFDE↓	MissRate↓
No Pretraining	×	Single	0.797	0.284	0.594	0.071
PPT (Ours)	Single	Single	0.739	0.265	0.536	0.064
PPT (Ours)	All	Single	**0.700**	**0.251**	**0.507**	**0.061**
			-12.18%	-11.61%	-14.59%	-14.49%
No Pretraining	×	All	0.739	0.295	0.553	**0.060**
PPT (Ours)	All	All	**0.713**	**0.264**	**0.522**	0.062
			-3.52%	-10.51%	-5.61%	+3.33%

Table 7: **Pretraining *w/o* HD maps.** Using PPT, we pretrain the forecasting model *w/* or *w/o* HD maps, compared to no pretraining. We note that all experiments pretrain and finetune using 'All' three datasets and HD maps are used during the finetuning phase. Relative gains (%, in teal) are given against the 'No Pretraining' baseline.

	Pretrain	Finetune	*Average* Brier-FDE↓	minADE↓	minFDE↓	MissRate↓
No Pretraining	×	All	0.739	0.295	0.553	0.060
PPT (Ours)	All	All	0.713	**0.264**	**0.522**	0.062
			-3.52%	-10.51%	-5.61%	+3.33%
PPT *w/o* Maps (Ours)	All	All	**0.706**	0.266	0.528	**0.059**
			-4.50%	-10.06%	-4.62%	-2.19%



(a) Curated traj. (b) DSVT-Voxel

(c) PV-RCNN++ (d) VoxelNeXt

Figure 3: **Diverse pseudo-labeled trajectories.** Compared to a single curated annotation in WOD [2], with pseudo-labeled trajectories, we provide not only the training example close to the ground truth but other feasible trajectories.

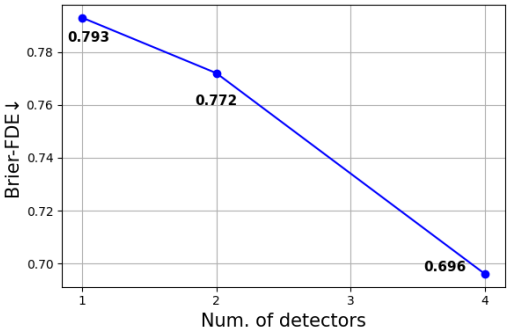


Figure 4: **Data diversity in pretraining matters.** MTR models are pretrained without finetuning with exactly 15,290 samples from 100 scenarios of WOD [2]. These samples come from one, two or four detection models. All models are evaluated on the full validation set of WOD.

4.5 Further Analyses

Data diversity matters. The pseudo-labeled trajectories produced by PPT are characterized by two key properties: they are noisier than curated annotations (see App. C), but also significantly more diverse, as illustrated in Fig. 3. We hypothesize that this diversity acts as a strong form of regularization, helping models learn richer and more generalizable motion representations. To quantify how diversity impacts performance, we conduct a controlled experiment where we fix the scenarios and the total number of pretraining trajectories and vary only the diversity of their sources. Specifically, on the same 100 WOD scenarios, we uniformly sample 15,290 pseudo-labeled trajectories from (i) a single detector [12], (ii) two different detectors [13, 12], or (iii) four detectors [13, 12, 11][3]. We then pretrain MTR [6] using these different pseudo-labeled trajectories and evaluate the impact to motion forecasting. As shown in Fig. 4, models pretrained on more diverse pseudo-labeled trajectories consistently outperform those trained on data from fewer sources. These results show the crucial role of diversity in pseudo-labeled trajectories for better effective motion forecasting pretraining.

Pretraining *w/o* HD maps. In all previous experiments, we assume the availability of HD maps, which are commonly used in motion forecasting research, to isolate the effect of pseudo-labeled trajectories during pretraining. In Tab. 7, we relax this assumption by pretraining the model without HD maps, using only the pseudo-labeled trajectories. Specifically, we use empty HD maps during pretraining. The results show that pretraining without HD maps achieves performance on par with the pretraining that incorporates them. This suggests that the primary benefit of pretraining comes from the trajectory dynamics and agent interactions, rather than the agent-map interactions.

5 Conclusion

This paper introduces PPT, a novel pretraining strategy for motion forecasting leveraging pseudo-labeled trajectories collected from off-the-shelf detectors and tracking methods. Without aiming for single-label clean annotations, PPT embraces noise and diverse trajectories without human effort, offering significant advantages: (1) PPT achieves superior results across various datasets, in particular with limited labeled data for finetuning. (2) Improved generalization capabilities of models trained with PPT, as tested with cross-domain, end-to-end, and multi-class settings. Further analyses show the importance of data diversity and the independence of HD maps. Overall, we establish PPT as a simple and effective framework toward robust motion forecasting in diverse driving contexts.

[3]DSVT [11] includes two variants: DSVT-Voxel and DSVT-Pillar.

Limitations

Currently, our proposed framework PPT operates on detectors with a fixed number of classes—this limits PPT from exploring diverse object classes in the real world. Therefore, with the flexibility and dataset-agnostic nature of PPT, we are motivated to extend PPT toward open-world motion forecasting, with the help of existing open-world 3D detectors [44, 45].

Acknowledgments

This work was supported by the ANR grant MultiTrans (ANR-21-CE23-0032). This work was performed using HPC resources from GENCI-IDRIS (Grant 2023-GC011015459). This research received the support of EXA4MIND project, funded by a European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement N°101092944. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[2] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[3] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

[4] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *ICCV*, 2021.

[5] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

[6] S. Shi, L. Jiang, D. Dai, and B. Schiele. Motion transformer with global intention localization and local movement refinement. In *NeurIPS*, 2022.

[7] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022.

[8] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *ICRA*, 2023.

[9] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *ECCV*, 2024.

[10] Y. Xu, L. Chambon, É. Zablocki, M. Chen, A. Alahi, M. Cord, and P. Pérez. Towards motion forecasting with real-world perception inputs: Are end-to-end approaches competitive? In *ICRA*, 2024.

[11] H. Wang, C. Shi, S. Shi, M. Lei, S. Wang, D. He, B. Schiele, and L. Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023.

[12] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *IJCV*, 2023.

[13] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, 2023.

[14] H. Liu, Y. Teng, T. Lu, H. Wang, and L. Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*, 2023.

[15] J. Yin, J. Shen, R. Chen, W. Li, R. Yang, P. Frossard, and W. Wang. Is-fusion: Instance-scene collaborative fusion for multimodal 3d object detection. In *CVPR*, 2024.

[16] T. Yin, X. Zhou, and P. Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021.

[17] X. Weng, J. Wang, D. Held, and K. Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *IROS*, 2020.

[18] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020.

[19] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi. Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents. In *CVPR*, 2021.

[20] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *ICRA*, 2023.

[21] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020.

[22] Y. Yuan, X. Weng, Y. Ou, and K. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, 2021.

[23] J. Gu, C. Sun, and H. Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021.

[24] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019.

[25] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. Lam, D. Anguelov, and B. Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, 2022.

[26] H. Ben-Younes, É. Zablocki, M. Chen, P. Pérez, and M. Cord. Raising context awareness in motion forecasting. In *CVPR workshop*, 2022.

[27] Y. C. Tang and R. Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2019.

[28] H. Cui, V. Radosavljevic, F. Chou, T. Lin, T. Nguyen, T. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *ICRA*, 2019.

[29] P. Kothari, D. Li, Y. Liu, and A. Alahi. Motion style transfer: Modular low-rank adaptation for deep motion forecasting. In *CoRL*, 2022.

[30] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang. Ganet: Goal area network for motion forecasting. In *ICRA*, 2023.

[31] Y. Xu, V. Letzelter, M. Chen, É. Zablocki, and M. Cord. Annealed winner-takes-all for motion forecasting. In *CoRR*, 2024.

[32] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023.

[33] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*, 2023.

[34] Y. Xu, É. Zablocki, A. Boulch, G. Puy, M. Chen, F. Bartoccioni, N. Samet, O. Siméoni, S. Gidaris, T.-H. Vu, A. Bursuc, E. Valle, R. Marlet, and M. Cord. Valeo4cast: A modular approach to end-to-end forecasting. In *CVPRW*, 2024.

[35] C. Xu, T. Li, C. Tang, L. Sun, K. Keutzer, M. Tomizuka, A. Fathi, and W. Zhan. Pretram: Self-supervised pre-training via connecting trajectory and map. In *ECCV*, 2022.

[36] J. Cheng, X. Mei, and M. Liu. Forecast-mae: Self-supervised pre-training for motion forecasting with masked autoencoders. In *ICCV*, 2023.

[37] P. Bhattacharyya, C. Huang, and K. Czarnecki. Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving. In *CoRL*, 2023.

[38] Z. Yang, L. Chen, Y. Sun, and H. Li. Visual point cloud forecasting enables scalable autonomous driving. In *CVPR*, 2024.

[39] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2022.

[40] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023.

[41] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C. Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022.

[42] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022.

[43] N. Peri, J. Luiten, M. Li, A. Osep, L. Leal-Taixé, and D. Ramanan. Forecasting from lidar via future object detection. In *CVPR*, 2022.

[44] W. Hu, W. Lin, H. Fang, Y. Wang, and D. Luo. Ow3det: Toward open-world 3d object detection for autonomous driving. In *IROS*, 2024.

[45] Z. Xia, J. Li, Z. Lin, X. Wang, Y. Wang, and M.-H. Yang. Openad: Open-world autonomous driving benchmark for 3d object detection. *CoRR*, 2024.

[46] Q. Li, Z. M. Peng, L. Feng, Z. Liu, C. Duan, W. Mo, and B. Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *NeurIPS*, 2023.

Appendix

A Implementation Detail

A.1 Datasets

We use three major *perception* datasets with both raw sensor (for pseudo-labeling) and trajectory data (for baselines and finetuning) available: • *NUS: nuScenes* [1] provides multi-modal driving scene data with LiDAR and camera perception. We use the train-validation split, consisting of 850 scenes (700 for training and 150 for validation), each sampled at 2 Hz [4] over 20 seconds. • *WOD: Waymo Open Dataset* [2] contains 1000 sequences with LiDAR and image sensor data, including 798 for training and 202 for validation, sampled at 10 Hz. • *AV2: Argoverse 2 Sensor Dataset* [3] contains 700 sequences for training and 150 for validation.

A.2 Pseudo-Labeled Trajectory Construction

We detail the detectors and trackers used in Sec. 3.2 for perception dataset pseudo-labeling.

Detection. Given the sensor data from the above datasets, we perform 3D object detection: • *NUS*: we collect detection results using LiDAR-only models—CenterPoint [16] and TransFusion-LiDAR [41]—as well as LiDAR-Camera bimodal models, BEVFusion [40] and IS-Fusion [15]. The detection results are collected at 2 Hz, which are then upsampled with linear interpolation to 10 Hz. • *WOD*: DSVT [11] with pillar-based voxel (DSVT-Pillar) and voxel-based (DSVT-Voxel) 3D backbones, LiDAR-based PV-RCNN++ [12] and VoxelNeXt [13] are used. • *AV2*: we detect objects using BEVFusion [40]. For each detector, we systematically apply Non-Maximum Suppression with the score threshold of 0.2 to obtain the final set of detections.

Tracking. To associate 3D detections and form the pseudo-labeled tracks/trajectories, we use *non-learning* tracking methods: CenterPoint's tracker [16] for the detection results on NuScenes and AB3DMOT [17] for others.

Object classes. We extract forecasting training examples at each time step of the collected sequences. Without further specification, we focus on the 4-wheel vehicle objects: `REGULAR_VEHICLE`, `LARGE_VEHICLE`, `BUS`, `BOX_TRUCK`, `TRUCK`, `VEHICULAR_TRAILER`, `SCHOOL_BUS`, `ARTICULATED_BUS`, and `VEHICLE`.

Statistics. We collected in total 8.6M pseudo-labeled trajectories for PPT pretraining: 6.9M from WOD, 0.5M from AV2, and 1.2M from NUS, as detailed in Tab. 8. In comparison, these three datasets contain only 2.1M ground-truth trajectories.

A.3 Motion Forecasting Details

Forecasting method. To facilitate the experiments on different datasets, we leverage UniTraj [9], where the input data are unified in the ScenarioNet [46] format. It also provides a reliable implementation of multiple forecasting methods such as [6, 8, 42]. The task is predicting future trajectories over 6 seconds from 2 seconds of past ground-truth history, both sampled at 10 Hz.

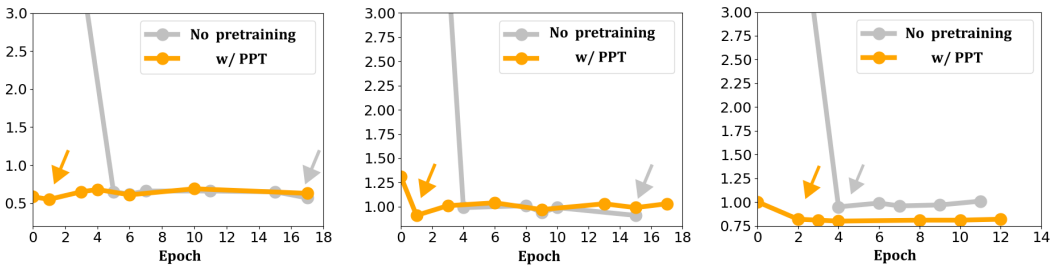
We conduct our experiments with MTR [6], a transformer-based method with 60M parameters, chosen for its superior performance. MTR integrates *intention points* that provide prior information on future endpoints to initialize their queries. We also show the performance of other motion forecasting methods such as Wayformer [8] and Autobot [42] in App. E.

Training. We pretrain the forecasting models for 16 epochs (in practice, convergence may be obtained before). For finetuning, we train for 40 epochs and report early-stopping scores based on validation evaluation. We follow the hyper-parameter setting of UniTraj [9] in all experiments. For finetuning, the learning rate is reduced by an order of magnitude of 0.1; the other hyper-parameters remain unchanged.

[4] We upsample the annotations to 10Hz by linear interpolation, as in [9].

B Efficient Supervised Finetuning

We show in Fig. 5 the brier-FDE curves when training from scratch and from PPT pretrained model. We observe that finetuning after PPT pretraining converges much faster, typically in only 1 or 2 epochs. For reference, one epoch takes around 16 minutes on WOD [2] with the MTR model. These results suggest that most of the forecasting ability was acquired during the pretraining phase, with minimal adaptation required to adjust the model to the labeled data of the benchmarks.



(a) WOD [2] (b) Argoverse 2 [3] (c) nuScenes [1]

Figure 5: **Efficient finetuning**. We show the evolution of brier-FDE on the validation sets during training from scratch/finetuning w/ PPT the forecasting MTR model. Model pretrained with PPT converges faster and achieves better results (indicated by arrows).

C Pseudo Labels: Statistics, Quality and Comparison

Table 8: **Statistics of pseudo-labeled trajectories** extracted from the training splits of the three perception datasets.

	Detector	# Trajs.
AV2	BEVFusion [40]	535K
NUS	CenterPoint [16]	378K
	TransFus.-L [41]	269K
	BEVFusion [40]	284K
	IS-Fusion [15]	276K
WOD	DSVT-Pillar [11]	1,7M
	DSVT-Voxel [11]	1,7M
	PV-RCNN++ [12]	978K
	VoxelNeXt [13]	2,5M
	Total	8.6M

Table 9: **Quality assessment of pseudo-labeled trajectories.** We show here the MF metrics as the proxy to demonstrate the quality of our pseudo-labeled trajectories used for pretraining.

		Trainset			Valset		
	Tracker	min-ADE↓	min-FDE↓	Miss-Rate↓	min-ADE↓	min-FDE↓	Miss-Rate↓
AV2	BEVFusion [40]	0.274	0.375	0.030	0.295	0.404	0.032
NUS	CenterPoint [16]	0.402	0.593	0.090	0.414	0.600	0.080
	TransFusion-L [41]	0.266	0.326	0.040	0.264	0.323	0.034
	BEVFusion [40]	0.256	0.310	0.035	0.250	0.298	0.026
	IS-Fusion [15]	0.256	0.312	0.037	0.252	0.305	0.027
WOD	DSVT-Pillar [11]	0.383	0.366	0.015	0.385	0.365	0.014
	DSVT-Voxel [11]	0.378	0.362	0.014	0.379	0.360	0.014
	PV-RCNN++ [12]	0.237	0.247	0.011	0.245	0.252	0.011
	VoxelNeXt [13]	0.564	0.532	0.042	0.548	0.511	0.039
	Average	**0.297**	**0.380**	**0.035**	**0.337**	**0.379**	**0.031**

C.1 Pseudo Labels Statistics

We collect pseudo-labeled trajectories by running off-the-shelf detectors [11, 40, 15, 16, 12, 13, 41] and a non-learning tracker [17, 16] on the training splits of the three perception datasets [1, 2, 3]. The pseudo-labeled trajectory contains 2-second past and 6-second future points of the current frame. We do not smooth or manually select the trajectories; all are results of the automated pipeline. Tab. 8 reports the number of pseudo-labeled trajectories collected per detector and per dataset.

C.2 Pseudo Labels Quality Analysis

In Tab. 9, we assess the forecasting performance of the collected tracking trajectories. Precisely, at each time step of a given sequence, we perform a matching between predicted and ground truth based on their past trajectories with a threshold of 2 meters. After that, we calculate forecasting metrics. In

general, the tracking trajectories deviate from the ground truth annotations by less than 0.5 meters on average, generally lower than the average forecast performance of the model used and trained with labeled data. Although they do not fully match the quality of curated annotations, we show that inexpensive and noisy trajectories are beneficial for motion forecasting pretraining, e.g., when ground truth is absent or limited.

C.3 Impact of Pseudo-Labeling Quality to Motion Forecasting

We pretrain MTR [6] with PPT on nuScenes pseudo-labeled trajectories obtained with either BEV-Fusion [40] or CenterPoint [16], with no finetuning; BEVFusion [40] is of higher quality than CenterPoint [16] (Tab. 9). We show in Tab. 10 that better pseudo-labeled trajectories lead to better performance. Still, pseudo-labeled trajectories of varying quality are complementary, and using all leads to optimal performance (Tab. 2).

Table 10: **Impact of pseudo-labeling quality to motion forecasting pretraining.** We show the forecasting performance when we pretrain the model with pseudo-labeled trajectories from NUS [1] of different quality. Without finetuning, all models are evaluated on the full validation set of NUS.

		nuScenes			
	Pretrain	Brier-FDE↓	minADE↓	minFDE↓	MissRate↓
PPT	CenterPoint	1.245	0.421	1.027	0.103
	BEVFusion	**1.143**	**0.409**	**0.943**	**0.100**

C.4 Pseudo-Labeled WOD vs. WOMD Trajectories

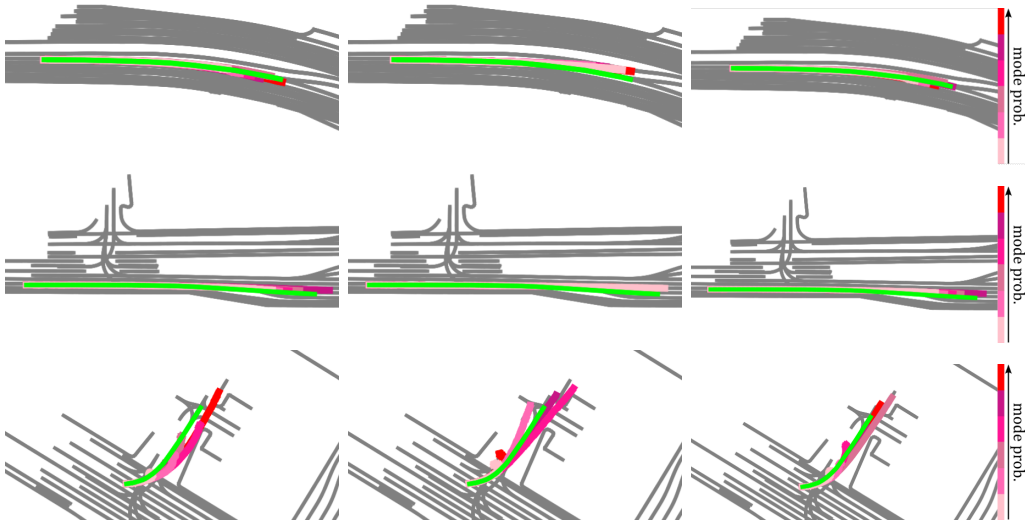
WOMD [4] contains mainly trajectories without the perception data like images or off-the-shelf detectors, which prevents us from directly using WOMD for pseudo-labeling. Also, WOMD employs an automatic annotation pipeline to provide *single-label* annotations close to the ones of human annotators. We assess the cross-domain performance by training models with either our collected WOD pseudo labels or WOMD annotations. As shown in Tab. 11, training with our diverse (several trajectories per agent) WOD pseudo labels without post-processing generalizes better than with curated WOMD trajectories in out-of-domain settings (AV2 and NUS). Those results further consolidate the potential of PPT, especially in diverse driving scenarios.

Table 11: **Pseudo-labeled WOD v.s. WOMD training.** Without finetuning, we show that PPT pretraining on pseudo-labeled of WOD [2] trajectories outperforms training on the WOMD [4] trajectories on the out-of-domain benchmarks. We use MTR [6] as the forecasting model.

Target	Source	Brier-FDE↓	minADE↓	minFDE↓	MissRate↓
AV2	WOMD [4]	1.781	0.607	1.460	0.170
	pseudo WOD	**1.485**	**0.508**	**1.218**	**0.119**
NUS	WOMD [4]	1.576	0.620	1.272	0.177
	pseudo WOD	**1.244**	**0.478**	**1.016**	**0.107**

D Forecasting Qualitative Results

We show some qualitative examples in Fig. 6 from models trained with (a)PPT *w/o* finetuning, (b) No pretraining, and (c) PPT *w/* finetuning. Overall, they perform reasonably well even for the model with only PPT pretraining. In the first two rows of Fig. 6, training with PPT, (a) and (c), provides more diverse and precise forecasts (better overlapped with GT), compared to less diverse modes and less precise predictions of (b) w/o PPT. In the last row, the trajectory of (b) with the highest probability falls behind the ground-truth prediction in green.



(a) PPT *w/o* finetuning (b) No pretraining (c) PPT *w/* finetuning

Figure 6: **Qualitative results.** We show some qualitative results comparing MTR trained with (a) PPT *w/o* finetuning, (b) No pretraining, and (c) PPT *w/* finetuning. Ground-truth future trajectories are shown in green. Predictions are colored in red with different intensities referring to mode probabilities, i.e. the predicted trajectories with the darkest red are predicted at the highest probability.

E Additional Results

We visualize the performance of Tab. 1 for MTR [6] in Fig. 7. Additionally, we show in Fig. 8 and Fig. 9, the results using Autobot [42] and Wayformer [8], respectively. We show that in both methods, PPT generally improves motion forecasting performance compared to "No pretraining", especially in the low-labeled-data regimes.

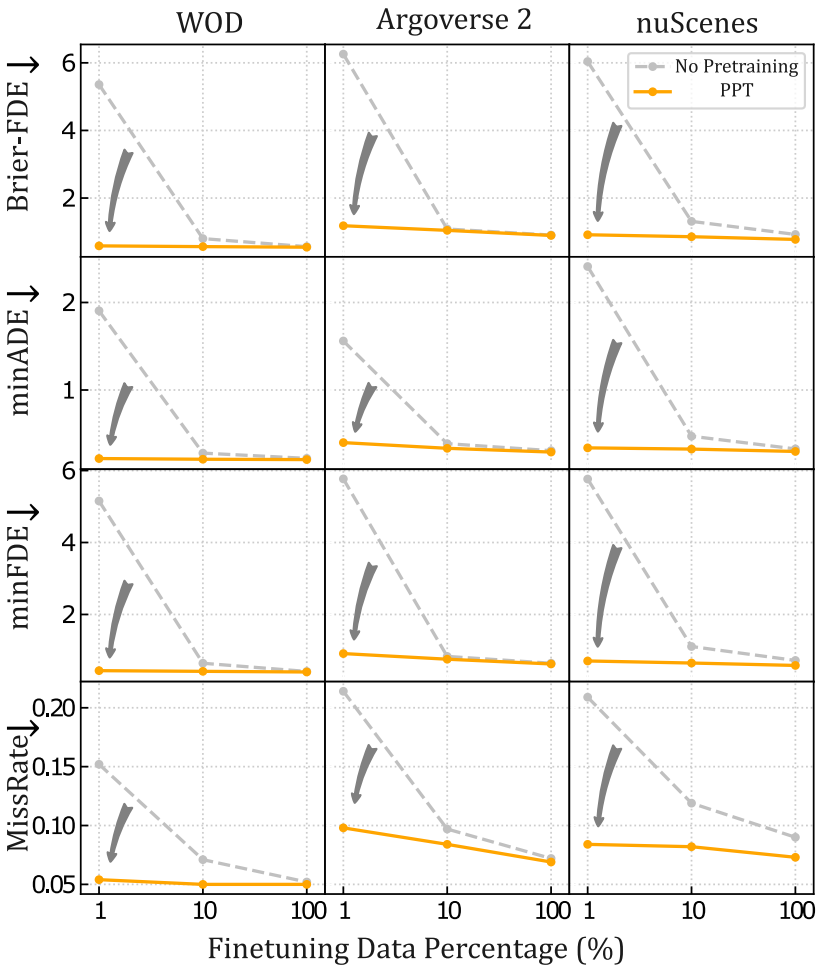


Figure 7: **No pretraining vs. pretraining** with PPT using MTR [6].

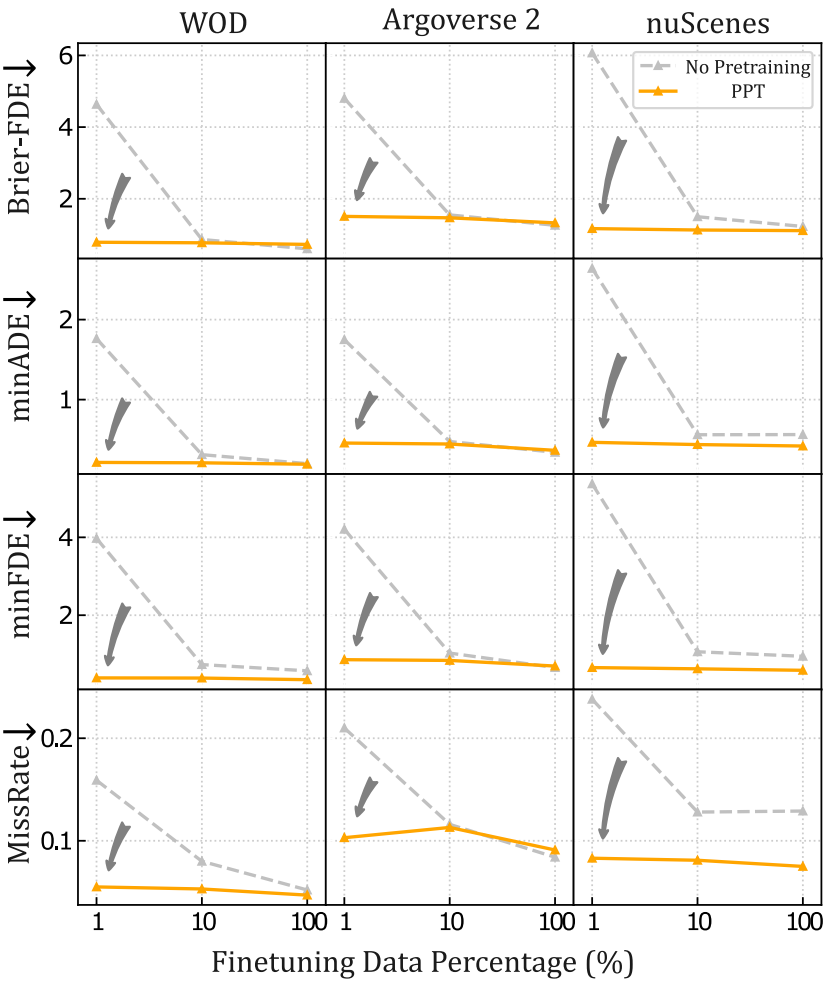


Figure 8: **No pretraining vs. pretraining** with PPT using Autobot [42].

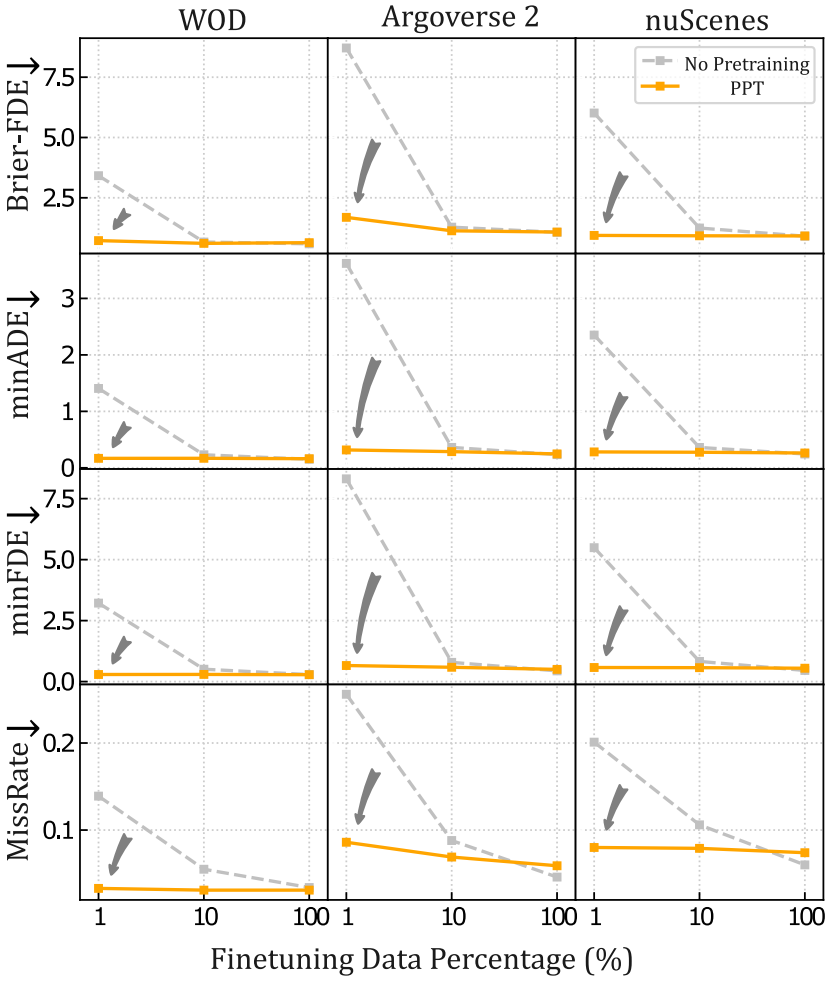


Figure 9: **No pretraining vs. pretraining** with PPT using Wayformer [8].