FairTraj-COSMOS: Train-Rich, Eval-Pure Missingness Protocols for Calibration-Robust Trajectory Prediction

Wangshu Zhu [1]   Chengbo Zang [1]   Zoran Kostic [1]

Abstract

Progress in trajectory prediction is often confounded by how missing observations are handled. We introduce FairTraj-COSMOS, a missingness-aware framework for the Cosmos dataset that (i) unifies scene conversion and valid-mask semantics, (ii) formalizes four protocols—A (strict filter), B (fill-as-real), C (fill-but-mask), and D (no-fill-mask), and (iii) elevates calibration via Brier–FDE alongside ADE/FDE and miss rate. On a common split with AutoBot, Wayformer, and MTR, we find: B-test is overly optimistic; under Eval-Pure (A/C/D), C-train yields the best calibration without accuracy loss (A slightly lowers ADE but worsens Brier–FDE; B gives the best FDE; D is weaker). Architecture matters: Wayformer calibrates better than AutoBot at similar latency, and MTR adds calibration gains at higher latency. Making missingness policy explicit enables fair, reproducible comparisons and supports the rule *Train-Rich, Eval-Pure*.

1. Introduction

Trajectory prediction systems in autonomous driving are frequently trained and evaluated on data with pervasive missingness—short tracks, dropped frames, and early exits are the norm rather than the exception. Such incomplete observations are typically addressed through time interpolation/extrapolation to fixed horizons or strict filtering of short sequences. However, these preprocessing choices alter task difficulty and target semantics: interpolation smooths targets and simplifies the task, while strict filtering skews the data distribution toward “well-behaved” scenes. The result is often systematically optimistic metrics, reduced comparability across studies, and poor probability calibration of predicted trajectories.

We introduce FAIRTRAJ-COSMOS, a missingness-aware evaluation framework designed to separate modeling ability from preprocessing artifacts. The framework (i) unifies scene conversion and valid-mask semantics across datasets, (ii) defines four explicit and reproducible missingness-handling protocols—A (strict filter), B (fill-as-real), C (fill-but-mask), and D (no-fill-mask), and (iii) elevates probability calibration to a first-class metric via Brier–FDE, alongside conventional ADE/FDE and miss rate. A unified conversion adapter standardizes heterogeneous data and embeds protocol semantics consistently in both training and evaluation.

Controlled experiments on a real-world intersection dataset with AutoBot, Wayformer, and MTR reveal three key findings. (1) **B-test is strongly optimistic**: treating filled targets as real substantially lowers ADE/FDE for all models, masking the true difficulty. (2) Under “Eval-Pure” settings (evaluation only on pure, unfilled targets via A/C/D), training with protocol C delivers the best calibration without sacrificing accuracy; protocol A achieves slightly lower displacement error but worse Brier–FDE, while protocol D is consistently weaker. (3) **Architecture matters**: Wayformer attains stronger calibration than AutoBot at nearly identical single-agent latency, while MTR offers additional calibration gains but at significantly higher latency. These results support the practical guideline—*Train-Rich, Eval-Pure*—for fair and calibration-robust trajectory prediction.

2. Background

2.1. Trajectory Prediction

Trajectory prediction aims to forecast an agent’s future states over F time steps given H observed steps of its past trajectory, optionally including the motion of neighboring agents and a high-definition map. The output is typically a multi-modal distribution over possible future trajectories. Standard evaluation benchmarks report top-k displacement errors—such as minimum Average Displacement Error (minADE) and minimum Final Displacement Error (minFDE)—and often augment them with probability calibration metrics like Brier score or negative log-likelihood (NLL) under standardized driving datasets (Rudenko et al., 2020; Sun et al., 2020; Gujarathi & Frossard, 2023).

Representative paradigms in trajectory prediction include:

- **Physics/Social rules** — Hand-crafted interaction models that enforce physically and socially plausible behav-

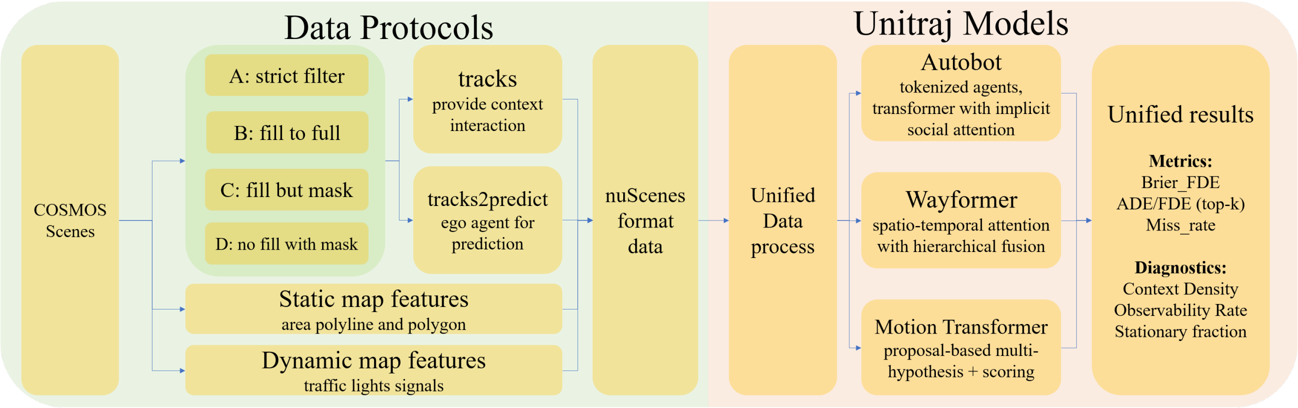


Figure 1. Project workflow. COSMOS scenes are processed under four missingness protocols (A–D) to produce `tracks` (context) and `tracks2predict` (targets), together with static and dynamic map features. A unified data process feeds UniTraj models (AutoBot, Wayformer, MTR), and evaluation reports unified metrics (Brier–FDE, ADE/FDE@6, Miss rate) and diagnostics (context density, observability ratio, stationary fraction).

ior, such as repulsive or attractive forces in pedestrian motion (Helbing & Molnár, 1995).

- **RNN/LSTM with pooling** — Sequence models predicting per-agent futures with recurrent encoders, combined with social pooling to capture spatial interactions (Alahi et al., 2016).
- **Graph-based interaction models** — Agents are modeled as nodes connected by learned edges; message passing captures heterogeneous interactions and map context (**?**).
- **Transformer scene models** — Self- or cross-attention across agents and map tokens enables joint modeling of interactions and long-range dependencies (Girgis et al., 2021; Nayakanti et al., 2022).
- **Anchor/candidate (detection-style) forecasting** — Predict a discrete set of candidate trajectories or goals and refine them for efficient, high-recall multi-modality (Shi et al., 2021; 2023).
- **Generative models (VAE/GAN/Diffusion)** — Latent-variable or score-based approaches explicitly sample diverse futures, with recent diffusion-based methods enabling controllable multi-agent forecasting (Gupta et al., 2018; Jiang et al., 2023).

In this work, we evaluate three representative models covering diverse architectural families: AutoBot (Transformer-based sequence modeling), Wayformer (efficient attention over agents and map), and MTR (anchor/candidate detection). This selection ensures that our findings on missingness-handling protocols are not tied to a single modeling paradigm but hold across distinct approaches to trajectory forecasting.

2.2. Missingness Processing

Real-world trajectory datasets contain substantial missingness: agents may enter or leave the field of view, be occluded, have broken tracks, or suffer from multi-sensor asynchrony, leading to variable-length histories. Many benchmarks expose this via availability masks indicating whether a state is physically observed at each time step (e.g., Argoverse provides per-step availability for forecasting; WOMD reports agent- and step-wise presence) (Chang et al., 2019; Sun et al., 2020).

Three common strategies are widely used to handle missingness:

- **Filtering** — Discard trajectories with short histories or truncated futures to enforce fixed observation and prediction windows (e.g., $H \geq 8, F \geq 12$). This preserves native kinematics but reduces context density and sample size, skewing the data toward stationary or easier cases.
- **Imputation (interpolation/extrapolation)** — Reconstruct missing states to a fixed horizon using linear or spline interpolation, or classical smoothing methods such as Kalman or Rauch–Tung–Striebel filters (**??**). This stabilizes optimization and increases context exposure, but treating imputed points as ground truth can redefine task difficulty and lead to optimistic displacement errors by smoothing high-curvature segments.
- **Explicit masking in learning and attention** — Modern architectures can process variable-length sequences together with masks, computing losses only on available targets and masking out invalid tokens in attention layers (e.g., Transformers or set-based encoders)

(?Nayakanti et al., 2022; Shi et al., 2021). This preserves evaluation purity while still leveraging dense context through auxiliary masked inputs.

A critical consideration here is calibration. When imputed segments are treated as observed targets, strictly proper scoring rules (e.g., Brier score, NLL) (Brier, 1950; Gneiting & Raftery, 2007) can be biased toward overconfident predictions on artificially smoothed trajectories. Consequently, recent practice favors protocols that (i) respect availability masks during evaluation and (ii) distinguish between using imputation as context versus as targets. These distinctions underpin our later definition of "Eval-Pure" and the comparative analysis of missingness-handling protocols.

2.3. Technical Challenges

Missingness — Real-world data inevitably contain short tracks, dropped frames, and early exits caused by occlusion and sensor asynchrony. In practice, three main handling strategies dominate: (i) strict filtering that enforces fixed H/F horizons, (ii) imputation via interpolation or extrapolation to a fixed length, and (iii) explicit valid masking so that models learn only from observed states with availability preserved. Prior studies have shown that seemingly minor choices at this stage can materially affect optimization difficulty and reported accuracy (Rudenko et al., 2020; Gujarathi & Frossard, 2023). Treating filled points as ground truth smooths curvature and can yield optimistic displacement errors; pure filtering reduces context diversity and biases the distribution toward "well-behaved" scenes. Modern forecasters may even incorporate re-sampling inside the decoder (e.g., MTR++ goal/anchor refinement), underscoring that this is part of the modeling stack rather than a benign preprocessing step (Shi et al., 2023).

Context density — For a target agent, context density measures the strength of potential interactions during the observation window, often approximated by the number of neighboring agents within a fixed spatial radius. Different architectures consume this context in different ways: pooling in RNNs (Alahi et al., 2016), learned message passing on graphs (Salzmann et al., 2020), or self-/cross-attention over scene tokens in Transformers (Girgis et al., 2021; Nayakanti et al., 2022). Performance is often systematically density-sensitive: imputation may dilute challenging short-cut interactions, filtering may drop rare but difficult cases, while masking preserves them but requires stronger model support.

Evaluation fairness — Reported metrics are not directly comparable across codebases and workflows. Unified frameworks that standardize data formats, models, and metrics have shown that rankings and absolute scores can shift significantly under a consistent pipeline (Feng et al., 2024). Beyond implementation, the composition of the test set itself is a fairness variable: if imputed points are included as targets, errors can be systematically underestimated; if training labels include filled segments, models may regularize around artifacts. Recent benchmarking work also highlights that both protocol choice and scenario composition can bias rankings (Chen et al., 2023), motivating the need to make exposure policy explicit.

These challenges motivate a controlled, missingness-aware protocol that separates training exposure from evaluation purity while making context density an explicit factor in analysis.

3. Data Protocols

3.1. Conversion Adapter

COSMOS scenes are deterministically converted into a MetaDrive-compatible scenario. Each scene (H=8 observed, F=12 forecast; total 20 steps) is expanded to per-agent tracks. Coordinates are transformed from pixels to meters by vertical flip and constant scaling; timestamps are $\{0, \ldots, 19\}/\text{fps}$ with fps=2.5; velocities are computed as first-order differences in meters × fps. The valid mask propagates native observations and futures, while missingness handling for targets follows the protocols described in Section 3.2. Dataset splits are performed *per scene* prior to agent expansion to prevent context or map leakage across splits.

The detailed conversion logic is shown in Algorithm **??**, which iterates over all agents to standardize coordinates, timestamps, and dynamic map states, and assigns valid masks according to the chosen protocol. The resulting MetaDrive scenario schema, illustrated in Figure **??**, includes per-agent states, dynamic map states, and map features in a reproducible format for downstream training and evaluation.

3.2. Missingness Protocols

Let $\mathbf{x}_{1:H} \in \mathbb{R}^{H \times d}$ and $\mathbf{y}_{1:F} \in \mathbb{R}^{F \times d}$ denote observed and future states, with a binary availability mask $\mathbf{m} \in \{0,1\}^{H+F}$.

We define an imputation operator $\mathcal{I}(\cdot)$ that temporally completes a sequence to $H+F$ steps (e.g., via linear/spline interpolation). Let `valid` be the evaluation mask. Each policy specifies two orthogonal choices: *Context exposure*—what enters tracks (training input). *Target purity*—what enters `tracks_to_predict` and how `valid` is set.

A. Strict Filter (native-only). Targets must satisfy $H \geq 8$, $F \geq 12$, and context contains only native spans. Maximizes evaluation purity but reduces context density and sample size.

Algorithm 1 COSMOS → Metadrive

```
Input: COSMOS scene
constants $H$=8, $F$=12, fps=2.5, div=20.
Output: MetaDrive scenario
initialize tracks[``center''] at the intersection center with constant position and valid$\equiv$ 1.
for $i = 0$ n_objs$-1$ do
    typ $\leftarrow$ VEHICLE or PEDESTRIAN from lbs[$i$]
    $(\mathbf{o}, \mathbf{f}) \leftarrow$ (obs[$i$], tgt[$i$]) in pixels
    convert each $(x, y)$ by $(x, (H_{px} - y))/$div {flip-$y$, scale to meters}
    $s \leftarrow \left\lfloor \frac{\text{frm}[i]-\text{horizon}}{\text{skip}} \right\rfloor + H$ {start index}
    allocate pos$\in \mathbb{R}^{20\times3}$, vel$\in \mathbb{R}^{20\times2}$, valid$\in \{0,1\}^{20}$
    write $\mathbf{o}$ to pos[$s : s{+}H, 0 : 2$], $\mathbf{f}$ to pos[$s{+}H : s{+}H{+}F, 0 : 2$]
    set valid true on the written indices
    vel[1 :] $\leftarrow$ (pos[1 :,0:2] $-$ pos[0 : $-1$,0:2]) $\times$ fps; vel[0] $\leftarrow$ vel[1]
    write tracks[oid] with type, state {pos,vel,valid}, and metadata
end for
tracks_to_predict $\leftarrow$ SELECTTARGETS(tracks, policy) {A/B/C/D in Sec. ??}
dynamic_map_states $\leftarrow$ per-step lane traffic-light states; map_features $\leftarrow$ polygons (meters) with derived centerlines
return MetaDrive scenario
```

B. Perfect Fill (filled-as-ground-truth). All agents are completed to $H{+}F$ by $\mathcal{I}(\cdot)$, and $\texttt{valid} = 1$ on imputed steps for both context and targets. Densest context and smooth optimization at the cost of label realism.

C. Selective Fill (missingness-aware). Context is completed by $\mathcal{I}(\cdot)$ but imputed steps are marked $\texttt{valid} = 0$; targets are native-only as in A. Train-time rich context with evaluation-pure targets; separates exposure from scoring.

D. Zero-Impute (native masked). No temporal completion; missing steps remain absent with $\texttt{valid} = 0$ in both context and targets. Most faithful to raw data; sparse optimization signals can make training brittle.

3.3. Dataset Diagnostics

This section analyzes how the four missingness strategies (A/B/C/D) reshape the *context-agent* distribution along four complementary axes: *observability*, *context density*, *kinematics*, and *risk profile*. Unless otherwise stated, all statistics are computed on *context agents* (i.e., all agents in the track field visible to the model), not only the prediction targets.

```
scenario = {
  "id": f"Cosmos_{horizon}",
  "version": "MetaDrive␣v0.3.0.1",
  "length": 20,
  "metadata": {
    "ts": np.arange(20) / fps,
    "metadrive_processed": False,
    "coordinate": "metadrive",
    "source_file": file,
    "dataset": "cosmos",
    "scenario_id": str(horizon),
    "sdc_id": "center",
    "tracks_to_predict": {
        "center": {
        "track_index": 0,
        "track_id": "center",
        "difficulty": 0,
        "object_type": "PEDESTRIAN"
        }
    },
    "object_summary": object_summary,
    "number_summary": number_summary
  },
  "tracks": tracks,
      # per-agent states {pos, vel, ...}
  "dynamic_map_states": traffic_lights,
      # per-step lane states + stop points
  "map_features": map_features
      # polygons (m) + centerlines
}
```

Figure 2. Scenario schema produced by the adapter (abbreviated).

Let $T{=}H{+}F$ be the total temporal horizon. For agent i at time t, let $\mathbf{p}_{i,t} \in \mathbb{R}^2$ denote its 2D position, $m_{i,t} \in \{0, 1\}$ the valid-mask indicator (1 if natively observed, 0 if filled or missing), and

$$\mathcal{N}_{i,t}(r) = \left\{\, j \neq i : \, \|\mathbf{p}_{j,t} - \mathbf{p}_{i,t}\|_2 \leq r, \; m_{j,t} = 1 \,\right\}$$

the set of *valid* neighbors within radius r.

3.3.1. Observability and Context Density

The observability ratio (OR) and fill ratio (FR) of a target agent i are

$$\mathrm{OR}_i = \frac{1}{T} \sum_{t=1}^{T} m_{i,t}, \qquad \mathrm{FR}_i = 1 - \mathrm{OR}_i. \tag{1}$$

Context density (CD) measures the mean number of valid neighbors within radius r, and the context-OR normalizes by neighbor presence:

$$\mathrm{CD}_i(r) = \frac{1}{T} \sum_{t=1}^{T} \big|\mathcal{N}_{i,t}(r)\big|, \qquad \rho_i(r) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\big[\, \big|\mathcal{N}_{i,t}(r)\big| > 0 \,\big], \tag{2}$$

$$\mathrm{OR}_i^{\mathrm{ctx}} = \frac{1}{T} \sum_{t=1}^{T} \frac{\sum_{j \in \mathcal{N}_{i,t}(r)} m_{j,t}}{\max\!\big(1, \, \big|\mathcal{N}_{i,t}(r)\big|\big)}. \tag{3}$$

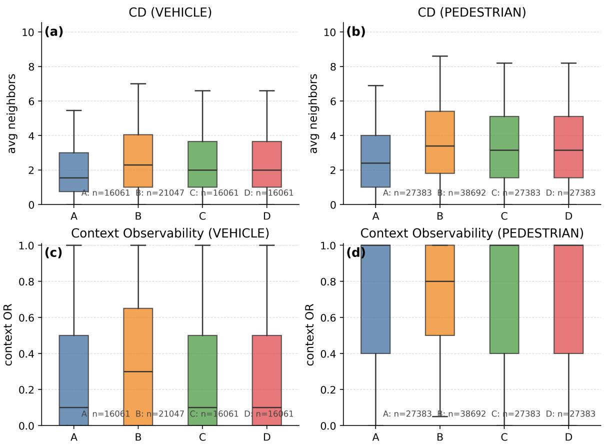


Figure 3. **Context density and observability (train, targets).** (a,b) Average valid neighbors (r=10 m). (c,d) Context OR (valid fraction among neighbors). Box limits show IQR with Tukey whiskers; sample counts are annotated.

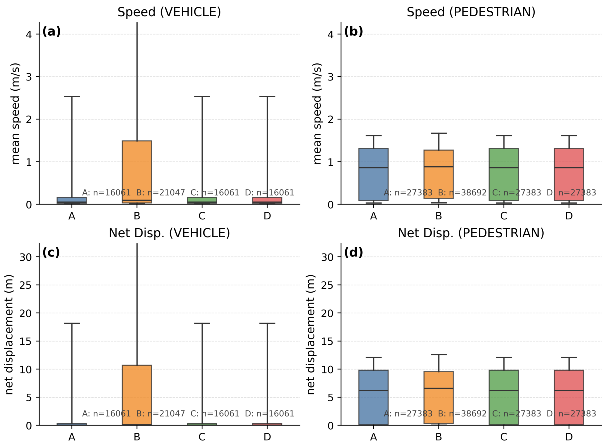


Figure 4. **Kinematics (train, targets).** (a,b) Mean speed $\bar{v}$; (c,d) net displacement Δ. Protocol B exhibits heavier upper tails for vehicles due to extrapolated long segments, while A/C/D remain close.

B yields the highest CD and context OR by filling all trajectories, producing densely observed neighborhoods; A is the sparsest due to strict filtering; C/D retain high CD through context-only completion while preserving native target spans. This distinction is critical: C/D expose the model to rich multi-agent context without label contamination, supporting better generalization than B and providing denser interaction cues than A.

3.3.2. Kinematics

For agent i, instantaneous *speed* and net displacement are

$$v_{i,t} = \frac{\|\mathbf{p}_{i,t} - \mathbf{p}_{i,t-1}\|_2}{\Delta t}, \qquad \bar{v}_i = \frac{1}{T-1}\sum_{t=2}^{T} v_{i,t}, \qquad \Delta_i = \|\mathbf{p}_{i,T} - \mathbf{p}_{i,1}\|_2. \tag{4}$$

An agent is deemed *stationary* if its displacement throughout the entire sequence remains below a category-specific velocity threshold for all frames:

$$\max_t v_{i,t} \le \varepsilon_{\text{type}}, \quad \varepsilon_{\text{veh}} = 1.0\ \text{m/s}, \ \ \varepsilon_{\text{ped}} = 2.0\ \text{m/s}.$$

B produces heavy-tailed $\bar{v}$ and Δ distributions for vehicles due to extrapolated long segments; A/C/D preserve native kinematics. B substantially reduces stationary fractions by converting idle spans into smooth motion; C/D maintain A-like stationary composition, avoiding unrealistic motion bias.

3.3.3. Risk Profile

For neighbor $j \in \mathcal{N}_{i,t}(r)$, define along-ray closing speed using displacement and *velocity vectors*

$$\mathbf{d}_{ij,t} = \mathbf{p}_{j,t} - \mathbf{p}_{i,t}, \qquad \Delta\mathbf{v}_{ij,t} = \mathbf{v}_{i,t} - \mathbf{v}_{j,t},$$

where $\mathbf{v}_{i,t} = (\mathbf{p}_{i,t} - \mathbf{p}_{i,t-1})/\Delta t$. The along-ray closing speed and time-to-collision (TTC) are

$$u_{ij,t} = \max\left(0, \frac{\Delta\mathbf{v}_{ij,t}^{\top}\mathbf{d}_{ij,t}}{\|\mathbf{d}_{ij,t}\|_2}\right), \qquad \mathrm{TTC}_{ij,t} = \frac{\|\mathbf{d}_{ij,t}\|_2}{u_{ij,t} + \epsilon}. \tag{5}$$

We report the p10 and median finite TTC, and the isolation fraction (agents with no closing neighbors).

Time-to-collision proxy (TTC). For each neighbor $j \in \mathcal{N}_{i,t}(r)$, let $\mathbf{d}_{ij,t} = \mathbf{p}_{j,t} - \mathbf{p}_{i,t}$ and $\Delta v_{ij,t} = \mathbf{v}_{i,t} - \mathbf{v}_{j,t}$ with $\mathbf{v}_{i,t} = (\mathbf{p}_{i,t} - \mathbf{p}_{i,t-1})/\Delta t$. Define the along-ray closing speed $u_{ij,t} = \max\big(0, \frac{\Delta v_{ij,t}^{\top}\hat{\mathbf{d}}_{ij,t}}{\|\hat{\mathbf{d}}_{ij,t}\|}\big)$, then a finite TTC is

$$\mathrm{TTC}_{ij,t} = \frac{\|\mathbf{d}_{ij,t}\|_2}{u_{ij,t} + \epsilon}, \quad \epsilon{=}10^{-3}. \tag{6}$$

Report $\min_{j,t} \mathrm{TTC}_{ij,t}$ over the window; ∞ indicates non-closing configurations.

Main findings.

- **A**: Clean, native-only supervision but context-sparse, limiting interaction learning.
- **B**: Maximally densified and smoothed context, but distorts kinematics and risk; excels when test-time distribution matches fill pattern but overfits otherwise.
- **C**: Best trade-off—rich training context without target contamination; maintains realistic motion and risk structure.
- **D**: Similar to C but with stricter masking, sometimes underutilizing available context.

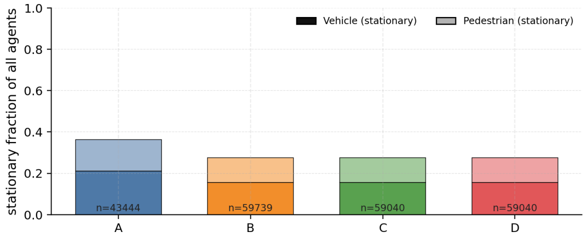


Figure 5. **Stationary composition (train, targets).** Bars show fractions of stationary vehicles and pedestrians. Protocol A yields the highest stationary share for vehicles; B reduces stationarity via time-filling.

	A	B	C	D
Vehicles				
p10 finite TTC ↓	2.81	0.91	1.67	0.86
Median finite TTC ↓	10.0	5.67	7.59	7.05
Isolation frac. ↓	0.377	0.214	0.258	0.253
Pedestrians				
p10 finite TTC ↓	0.99	0.79	0.87	0.39
Median finite TTC ↓	3.81	2.30	2.50	2.22
Isolation frac. ↓	0.195	0.106	0.119	0.118

Table 1. **TTC and isolation (train, targets).** Lower is riskier. B (time-filling with `valid=true`) consistently increases proximity; C preserves risk while avoiding test-time fill.

4. Model Architectures

4.1. UniTraj Overview

UniTraj is a unified research framework for trajectory prediction that eliminates discrepancies in data interfaces, masking semantics, and evaluation pipelines, enabling fair, apples-to-apples comparisons across models under different missingness protocols. This unification allows us to focus on how architectures adapt to context sparsity or densification, rather than on dataset-specific preprocessing or metric idiosyncrasies.

Specifically, UniTraj standardizes three key aspects:

- **Scenario adapter.** All datasets are converted into a shared per-agent tensorization with explicit valid masks for missing frames. Geometric units and timestamps are normalized, and map elements are represented as polylines with fixed sampling. This is essential in our experiments, as it ensures that the definition of "missing context" is strictly consistent across the four protocols (A/B/C/D), isolating the effect of the data protocol from the model itself.

- **Batched inputs.** Agents, map tokens, and temporal windows are packed into fixed shapes using padding and masks, guaranteeing that all three representative architectures operate under the same temporal resolution and neighborhood scope. This uniformity is key for our later comparison in Section 5, where we evaluate how different interaction modeling mechanisms respond to sparse versus dense context.

- **Unified evaluation harness.** The framework provides common metrics (ADE, FDE, miss rate, and Brier score) and a single post-processing pipeline for ground-truth alignment and metric aggregation. This removes evaluator-level bias and ensures that any performance differences in Section 5 can be attributed to architectural behavior rather than evaluation discrepancies.

4.2. Representative Architectures

We evaluate three representative architectures—AutoBot, Wayformer, and MTR—which, when run under UniTraj's unified I/O and masking rules, collectively span three major paradigms of interaction modeling in modern trajectory prediction. This diversity not only improves the generality of our findings but also enables us to investigate how different interaction mechanisms exhibit robustness (or fragility) under varying degrees of context availability.

AutoBot — Implicit, set-based interaction. Encodes agents and map polylines as temporal tokens and applies stacked self-/cross-attention to capture motion history and context. Decoding is set-based: a fixed set of learnable queries attends to the encoded scene to produce multi-modal predictions. Interactions between agents are modeled implicitly via token attention, without handcrafted pooling. While highly flexible for aggregating long-range dependencies, its reliance on implicit attention can make it sensitive to severe context sparsity—an effect we later observe in B → A/C/D transfer experiments.

Wayformer — Hierarchical, explicit cross-stream fusion. Uses hierarchical attention to fuse multiple heterogeneous streams (agent–agent, agent–map, temporal) and employs multi-scale feature aggregation to improve long-horizon reasoning (Nayakanti et al., 2022). Its structured cross-stream fusion excels in dense contexts but shows diminishing returns when neighborhood information is heavily missing, as the computation budget does not translate proportionally into predictive gains.

MTR — Joint multi-agent reasoning with coarse-to-fine refinement. Predicts candidate endpoints or modes via a coarse stage, followed by trajectory refinement conditioned on scene context (Shi et al., 2021; **?**). This design offers strong stability in dense scenes, while the coarse-to-fine approach helps preserve performance even under partial missingness (C/D). Our analysis later highlights that MTR's stability in sparse settings aligns closely with the robustness of its decoding strategy.

4.3. Metrics

We evaluate all models under UniTraj's unified metric suite, ensuring that performance differences stem from model–protocol interactions rather than inconsistencies in computation or aggregation. The metrics span three complementary dimensions:

Geometric accuracy — Average/Final Displacement Error (ADE/FDE). For each predicted mode k, ADE measures the mean pointwise distance to the ground truth over the prediction horizon, while FDE captures the endpoint error. Following common multi-modal evaluation, we report the oracle top-K errors:

$$\mathrm{ADE}_k = \frac{1}{T}\sum_{t=1}^{T}\left\|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(k)}\right\|_2, \qquad \mathrm{FDE}_k = \left\|\mathbf{y}_T - \hat{\mathbf{y}}_T^{(k)}\right\|_2, \tag{7}$$

$$\min \mathrm{ADE}@K = \min_{1\le k\le K} \mathrm{ADE}_k, \qquad \min \mathrm{FDE}@K = \min_{1\le k\le K} \mathrm{FDE}_k. \tag{8}$$

These capture the best achievable geometric match among K proposals (K=6 in our experiments). In Section 5, we leverage these to isolate purely geometric degradation under varying missingness protocols.

Mode recall — Miss Rate (MR@ τ). MR@ τ measures the fraction of target agents for which no predicted mode lies within τ meters of the ground-truth endpoint:

$$\mathrm{MR}_\tau = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left\{\min_k \left\|\mathbf{y}_T^{(i)} - \hat{\mathbf{y}}_T^{(k,i)}\right\|_2 > \tau\right\}. \tag{9}$$

Lower MR indicates better coverage of plausible futures. In sparse contexts (Protocols C/D), MR@ τ reveals whether models can still recover the correct mode despite incomplete histories.

Probabilistic calibration — Brier score (brier_fde). Accuracy alone does not assess whether the predicted probability mass is assigned to the correct mode. We compute the Brier score over the predictive mixture using the oracle-best mode k^* (?Gneiting & Raftery, 2007):

$$\mathrm{Brier} = \frac{1}{K}\sum_{k=1}^{K}\left(p^{(k)} - q^{(k)}\right)^2, \qquad q^{(k)} = \mathbb{I}\{k = k^*\}. \tag{10}$$

This metric complements geometric ones by exposing probability misallocation—a key signal in our analysis of calibration drift across protocols.

Note. All metrics are computed under the same evaluation harness (Sec. 4.1) with consistent units, horizons, and valid-mask semantics. This guarantees that any performance gap in Sec. 5 reflects the interplay between data missingness and architectural design, rather than evaluator bias.

Table 2. Autobot protocol sensitivity.

minADE6 (m)

Train\Test	A	B	C	D
A	0.1981	0.4268	0.1978	0.1979
B	0.2000	0.2775	0.1991	0.1992
C	0.2003	0.4184	0.2000	0.2002
D	0.2026	0.4402	0.2022	0.2023

minFDE6 (m)

Train\Test	A	B	C	D
A	0.3408	0.5399	0.3398	0.3400
B	0.3392	0.4053	0.3384	0.3386
C	0.3459	0.5438	0.3452	0.3450
D	0.3492	0.5597	0.3486	0.3487

Brier-FDE (m)

Train\Test	A	B	C	D
A	0.6621	0.9456	0.6594	0.6600
B	0.6705	0.8025	0.6683	0.6694
C	0.6424	0.9290	0.6407	0.6409
D	0.6536	0.9497	0.6516	0.6521

Miss rate

Train\Test	A	B	C	D
A	0.0286	0.0579	0.0283	0.0287
B	0.0296	0.0335	0.0300	0.0299
C	0.0304	0.0580	0.0300	0.0301
D	0.0308	0.0603	0.0301	0.0301

5. Experimental Results

5.1. Protocol–Sensitive Results

To directly assess how data protocol semantics affect trajectory prediction, we evaluate two representative predictors—AutoBot and Wayformer—each trained on one of the four protocols (A/B/C/D) and tested across all protocols. All results are reported under the unified metrics introduced in Sec. 4.3: geometric accuracy (minADE@6, minFDE@6), probabilistic calibration (Brier–minFDE), and binary endpoint coverage (Miss@2 m). Tables 2 and 3 summarize the cross-protocol performance for the two architectures.

Key observations. Despite architectural differences, both AutoBot and Wayformer exhibit consistent performance patterns across training protocols:

- **A-train minimizes ADE and Miss.** Strictly filtering to only real, high-quality target positions yields the lowest minADE@6 and Miss@2 m in almost all pure-protocol tests (A, C, D).
- **B-train minimizes FDE but hurts calibration.** Filling missing targets to full length produces the best minFDE@6, yet systematically yields the worst Brier–minFDE.

Table 3. Wayformer protocol sensitivity.

minADE6 (m)

Train\Test	A	B	C	D
A	0.2084	0.4695	0.2093	0.2091
B	0.2116	0.2957	0.2113	0.2113
C	0.2169	0.4898	0.2170	0.2168
D	0.2280	0.4921	0.2280	0.2278

minFDE6 (m)

Train\Test	A	B	C	D
A	0.3644	0.6420	0.3670	0.3666
B	0.3651	0.4433	0.3644	0.3646
C	0.3869	0.6646	0.3867	0.3867
D	0.4079	0.6838	0.4075	0.4071

Brier-FDE (m)

Train\Test	A	B	C	D
A	0.6407	0.9987	0.6448	0.6438
B	0.6842	0.8204	0.6824	0.6829
C	0.6259	0.9940	0.6263	0.6257
D	0.6453	1.0137	0.6444	0.6442

Miss rate

Train\Test	A	B	C	D
A	0.0328	0.0898	0.0335	0.0331
B	0.0302	0.0352	0.0302	0.0303
C	0.0345	0.0904	0.0343	0.0342
D	0.0379	0.0988	0.0385	0.0384

- **C-train achieves the best calibration without ADE loss.** Masking filled targets preserves the strong calibration of B-train while retaining ADE competitive with A-train.
- **D-train underperforms overall.** Zero-insertion with masks leads to degradation in all metrics, suggesting that realism of missingness alone is insufficient without context densification.

Protocols account.

- **A-train (Filtered targets).** Protocol A removes short or incomplete tracks, reducing noisy supervision and emphasizing agents with steady, near-linear motion. This lowers geometric error—particularly ADE, which averages deviations over the entire horizon—and reduces the fraction of large deviations exceeding the 2 m miss threshold. However, the reduced context density can slightly weaken long-horizon accuracy on dense-test protocols.
- **B-train (Filled targets).** Protocol B densifies both context and targets via interpolation, yielding smoother endpoints and lower-variance gradients at the horizon. This explains the improved minFDE@6. Yet, the filled endpoints also introduce overconfidence: the model concentrates probability mass on the filled target mode, even when the true target is multi-modal. This increases the penalty term $(1-p)^2$ in Brier–minFDE, degrading calibration across pure-protocol tests.

Table 4. C_{train}: cross–architecture results on four test protocols. Lower is better. Brier–FDE reflects calibration; ADE/FDE/Miss report geometric accuracy.

Model	Metric	Test protocol			
		A	B	C	D
AutoBot	Brier$_FDE$	0.6424	0.9290	0.6407	0.6409
	minADE6	0.2003	0.4184	0.2000	0.2002
	minFDE6	0.3459	0.5438	0.3452	0.3450
	Miss rate	0.0304	0.0580	0.0300	0.0301
Wayformer	Brier$_FDE$	0.6259	0.9940	0.6263	0.6257
	minADE6	0.2169	0.4898	0.2170	0.2168
	minFDE6	0.3869	0.6646	0.3867	0.3867
	Miss rate	0.0345	0.0904	0.0343	0.0342
MTR	Brier$_FDE$	0.5977	0.9540	0.5978	0.6025
	minADE6	0.2520	0.4935	0.2543	0.2557
	minFDE6	0.4743	0.7373	0.4791	0.4815
	Miss rate	0.0405	0.0909	0.0421	0.0420

- **C-train (Context densified, targets masked).** Protocol C separates context densification from target supervision—filling frames for context but masking them for target loss. This enriches scene interactions (denser neighbors, richer map usage) while avoiding overfitting to interpolated endpoints. As a result, Brier–minFDE improves markedly over B-train, while ADE remains within a narrow margin of A-train (differences $\approx$ 2–3 cm). This balance makes C-train the most robust choice under mixed deployment conditions.
- **D-train (Raw missingness).** Protocol D preserves missingness “as-is” with zeros and masks, reducing both target and context supervision density. Sparse gradients and fragmented context hinder both ADE/FDE and calibration, showing that realistic missingness alone is insufficient without structural context augmentation.

Key takeaway. These consistent trends across AutoBot and Wayformer indicate that data protocol semantics—context density, target masking, and supervision quality—dominate performance differences more than architectural details. In particular, C-train with Eval–Pure offers the best trade-off between geometric accuracy and probabilistic calibration, aligning well with deployment requirements for reliable uncertainty estimation.

5.2. Architecture-Sensitive Results

Under a unified training protocol (Protocol C), we compare three representative predictors—MTR, Wayformer, and AutoBot—across four test protocols (A/B/C/D). Table 4 reports results where Brier–FDE quantifies calibration quality, while minADE@6, minFDE@6, and Miss rate measure geometric accuracy.

Key Observations. On unbiased tests (A/C/D), Brier–FDE exhibits a consistent ordering: MTR $<$ Wayformer $<$ AutoBot. For example, on the C-test, the values are 0.5978, 0.6263, and 0.6407, respectively. In contrast, the geometric metrics follow the opposite trend—AutoBot achieves the lowest minADE@6, minFDE@6, and Miss rate, followed by Wayformer, with MTR being least accurate in geometry.

This inversion highlights the pivotal role of the predicted probability p: Brier–FDE augments minFDE with a penalty term $(1-p)^2$, where p is the probability assigned to the mode whose final displacement is closest to the ground truth. Good calibration requires alignment between the *closest* mode and the *highest-scored* mode. Different architectures impose fundamentally different forms and densities of probability supervision, shaping this alignment and driving the observed performance divergence.

Mechanistic Account. AutoBot (best-of-K Hungarian matching). Hungarian matching locks each training sample to a single best hypothesis for regression updates. This produces sparse and indirect supervision for probabilities:

- The chosen mode is geometrically optimal but its probability p may be poorly aligned with the ground truth's multi-modal structure.
- Non-selected modes receive little or no gradient signal, making probability heads less calibrated.
- *Outcome:* Excellent geometric accuracy (lowest minADE/minFDE) but low p, resulting in high Brier–FDE.

Wayformer (multi-head decoding with explicit mode scores). Each head outputs a trajectory and an explicit score, trained via a classification-style objective jointly with regression:

- Provides dense and direct probability supervision with clear gradient flow.
- When a geometrically best head wins, its score rises, improving p and reducing the $(1-p)^2$ penalty.
- However, competition among heads and capacity splitting slightly reduce geometric accuracy compared to AutoBot.

Table 5. Efficiency Summary (single-agent)

Model	Params (M)	Mean (ms)	FPS
AutoBot	1.5	22.23	45.0
Wayformer	16.5	23.14	43.2
MTR	63.9	86.54	11.6

Note: Latency is the wall-clock time of the model's forward pass only; for the same GPU, operator precision, compilation options, and batch size of 1; p90 reflects tail stability. Training throughput depends on batch and optimizer configuration and should be supplemented as needed.

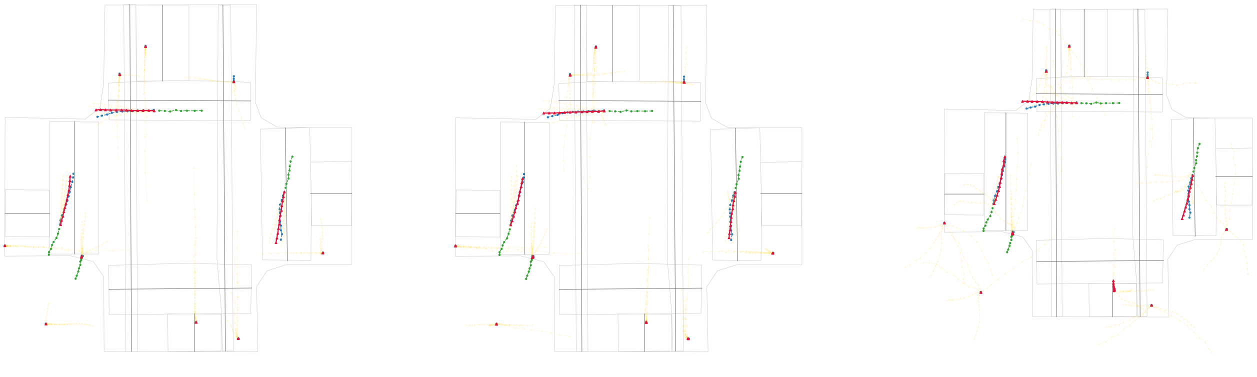
MTR (anchored two-stage with explicit classification). Anchors generate mode proposals, followed by refinement and explicit probability outputs:

- The classification branch receives abundant, clean supervision for p, ensuring strong alignment with the closest mode.
- Produces the lowest Brier–FDE among all three architectures.
- The geometric flexibility of the anchor framework is more limited, leading to slightly worse minADE/minFDE on this dataset.

5.3. Efficiency & Qualitative Behavior

Table 5 summarizes the efficiency profile of the three architectures under a single-agent inference setting. AutoBot is extremely lightweight (1.5 M parameters) and achieves the highest throughput at **45.0 FPS** with minimal latency (22.23 ms). Wayformer is an order of magnitude larger (16.5 M parameters) but maintains similar latency (23.14 ms) and competitive FPS (43.2), benefiting from efficient Transformer-based processing. In contrast, MTR has the heaviest footprint (63.9 M parameters), resulting in significantly higher latency (86.54 ms) and the lowest FPS (11.6), which poses deployment challenges in latency-sensitive applications despite its calibration advantage.

Figure 5 provides qualitative predictions for a single scene from the C-protocol–trained models evaluated on the C test set. The gray lines depict map centerlines, green traces indicate observed history, blue lines mark future ground truth, red traces correspond to the best predicted trajectory, and gold traces denote the remaining five predicted modes (rendered with low opacity to reduce visual clutter). All three models share the same coordinate system and rendering pipeline. From the visualization, AutoBot and Wayformer show tightly clustered predictions near the ground truth, reflecting strong geometric alignment, while MTR produces a more dispersed set of candidate trajectories, reflecting its explicit multi-anchor design and stronger probability calibration, but at the cost of additional spread.



(a) AutoBot

(b) Wayformer

(c) MTR

Figure 6. Qualitative results of the C-protocol trained model on the C test set (single-scene example). Gray represents the centerline of the map; green represents the observed history; blue represents the future ground truth; red represents the best predicted model; gold represents the remaining five models, with low transparency to minimize interference. All three models are rendered using the same coordinate system and the same visualization pipeline.

6. Conclusion and Future Work

This work systematically examined the interplay between data protocol design and model architecture in multi-modal trajectory prediction, introducing a unified evaluation framework that disentangles geometric accuracy from probabilistic calibration. Through controlled experiments across four protocol variants (A/B/C/D) and three representative architectures (AutoBot, Wayformer, MTR), we achieved the objectives outlined in the abstract:

- **Quantifying protocol sensitivity** — Our analysis revealed clear and consistent trends: protocol A favors ADE/Miss, protocol B minimizes FDE but harms calibration, protocol C achieves the best calibration without sacrificing geometric accuracy, and protocol D underperforms overall.
- **Isolating architecture effects** — Under the C-protocol, Brier–FDE consistently ranked MTR $<$ Wayformer $<$ AutoBot in calibration, while geometric accuracy followed the inverse order, highlighting a robust probability–geometry trade-off.
- **Proposing actionable guidelines** — The combined insights from protocol- and architecture-sensitive results support the principle: *Train-Rich, Eval-Pure*, and underscore the importance of dense, explicit probability supervision for calibration-critical deployments.

Lessons learned from this study include:

- Calibration quality is not merely a byproduct of geometric accuracy but is deeply influenced by how data density, masking, and supervision semantics interact with architectural probability heads.
- Certain “quick gains” in endpoint metrics (e.g., from densifying targets) can introduce systematic calibration biases, which may be detrimental in safety-critical applications.
- Visualization analysis confirms that well-calibrated models maintain plausible alternative modes rather than collapsing onto a single high-confidence path.

Future work will extend this analysis to:

- Multi-agent, interactive settings where protocol effects may compound due to agent–agent dependencies.
- Curriculum or hybrid protocols that adapt masking and densification dynamically during training.
- Efficient architectures that preserve calibration benefits while reducing latency, narrowing the gap between MTR’s calibration and AutoBot/Wayformer’s runtime efficiency.

By jointly analyzing data semantics and architectural mechanisms, this work contributes a reproducible foundation for balancing accuracy and reliability in probabilistic motion forecasting.

Acknowledgements

I would like to sincerely thank my advisor Zoran and Chengbo for their invaluable guidance, and my lab colleagues for their constant support, constructive feedback, and encouragement throughout this project. Their insights and collaboration were instrumental in bringing this work to completion.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–971, 2016.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, P., Ramanan, D., and Hays, J. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

Chen, C., Pourkeshavarz, M., and Rasouli, A. Criteria: a new benchmarking paradigm for evaluating trajectory prediction models for autonomous driving. *arXiv preprint arXiv:2310.07794*, 2023.

Feng, L., Bahari, M., Ben Amor, K. M., Zablocki, É., Cord, M., and Alahi, A. Unitraj: A unified framework for scalable vehicle trajectory prediction. In *European Conference on Computer Vision (ECCV) Workshops*, 2024. arXiv:2403.15098.

Girgis, R., Czarnecki, W. M., Pascanu, R., Richemond, P. H., Berner, C., Veličković, P., Simonyan, K., and Heess, N. Latent variable sequential set transformers for joint multi-agent motion prediction. In *Proceedings of the International Conference on Machine Learning*, pp. 3789–3799. PMLR, 2021.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Gujarathi, P. and Frossard, P. Survey of motion prediction methods for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 3788–3797, 2023.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

Helbing, D. and Molnár, P. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

Jiang, B., Ding, M., Wu, Y., Zhao, H., Tomizuka, M., Zhou, Y., and Zhan, W. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9742–9752, 2023.

Nayakanti, N., Hong, J., Sapp, B., Ranjbar, M., Anguelov, D., Sminchisescu, C., and Zhou, Y. Wayformer: Motion forecasting via simple and efficient attention networks. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2022.

Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., and Arras, K. O. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

Salzmann, T., Ivanovic, B., Chakravarty, P., and Pavone, M. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pp. 683–700. Springer, 2020.

Shi, H., Li, X., Ma, Y., Wang, Z., He, C., Gu, X., Li, S., Wang, P., and Yang, R. Mtr: Multi-agent trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14834–14843, 2021.

Shi, H., Li, X., Ma, Y., He, C., Wang, Z., Gu, X., Wang, P., and Yang, R. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9712–9722, 2023.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2443–2451, 2020.