PriMoTraj: Motion-Prior Gating for Lightweight GPS-Derived Trajectory Forecasting

En Chen, Yuanbo Xu, Yang Li
Jilin University, Changchun, China
{chenen5523, liyang5523}@mails.jlu.edu.cn, yuanbox@jlu.edu.cn

Weitong Chen
The University of Adelaide, Adelaide, Australia
t.chen@adelaide.edu.au

***Abstract*—For online mobility tasks, global positioning system (GPS)-derived trajectory prediction is evaluated by meter-level average displacement error (ADE) and final displacement error (FDE), while many models are trained in normalized coordinates. PriMoTraj targets a lightweight GPS-only setting with short-to-moderate histories and no road graph, point-of-interest features, or pretraining. It combines a compact temporal encoder with a sample-wise gate over seven deterministic motion priors, then applies a bounded full-window residual head to correct the weighted prior. On Porto with $T = 12$, PriMoTraj uses 11.1K parameters at $H = 6$ and achieves the lowest ADE among input-matched baselines, 234.08 $\pm$ 0.18 versus 237.50 $\pm$ 2.38 for long short-term memory sequence-to-sequence with attention (LSTM-Seq2Seq-Attn), while using less than half the parameters. At $H = 12$, it reaches 432.97 $\pm$ 0.43 ADE, trailing LSTM-Seq2Seq-Attn by 16.7 m. Controlled Porto downsampling further shows that preferred motion priors change with sampling interval, motivating joint reporting of ADE/FDE, latency, forecast duration, and input assumptions.**

***Index Terms*—Trajectory Prediction, Lightweight Forecasting, Motion Prior, GPS-Derived Forecasting, Average Displacement Error**

I. Introduction

Trajectory prediction supports route continuation, dispatching, estimated time of arrival (ETA) inference, and risk-aware planning [1], [2]. In these settings, the quantity used by downstream systems is displacement in meters, after the predicted coordinates have been inverse-transformed. The loss optimized during training is often a normalized coordinate error. That distinction is easy to overlook, but it can change the empirical ordering of models: two forecasts that are close in normalized space may differ noticeably after geographic scaling, and average/final displacement error (ADE/FDE) is computed only on latitude-longitude outputs. The issue is most visible in short-window urban forecasting, where predictions are queried repeatedly and where both meter-level error and online cost matter.

Recent time-series models, including PatchTST [3], iTransformer [4], TimeMixer [5], and AMD [6], have improved generic forecasting through patching, inverted tokenization, decomposition, and multi-scale mixing. These backbones are useful comparisons, but coordinate forecasting has additional structure. A short taxi trajectory may be well described by a constant-velocity extrapolation, a stop, a damped motion, or a smoother long-window trend. Which motion prior is useful depends on the sampling interval and on the recent local motion. A fixed number of forecast steps does not always mean the same future duration, and the same future duration can correspond to different values of H under different sampling intervals. This makes the sampling regime part of the modeling problem, rather than a detail that can be left implicit.

Corresponding author: Yuanbo Xu.

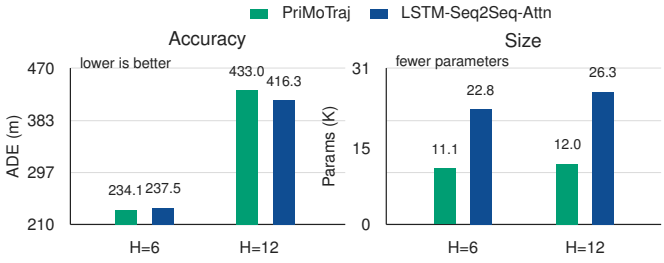

Fig. 1. Porto $T = 12$ result preview in the input-matched GPS-derived setting. ADE bars show three-seed mean $\pm$ one standard deviation; the parameter panel is deterministic. Full baselines, latency, and bootstrap tests are in Sec. IV-D.

These properties make short-window trajectory forecasting less a generic sequence-to-sequence task than a data-mining problem about matching trajectory structure to the evaluation and sampling regime. In this paper, *sampling-regime-aware* refers to evaluation and prior allocation under explicitly specified sampling intervals, rather than to a single interval-conditioned model trained across all regimes.

Road graphs, point-of-interest (POI) features, user histories, and pretrained trajectory encoders can be valuable, but they change what information is available to the model. This paper studies a narrower GPS-derived input setting: prediction from recent trajectory features alone, with no road graph, POI features, pretrained trajectory encoder, or future context. Under this restriction, simple constant-position and constant-velocity predictors are not uniformly weak, but one motion prior cannot cover stops, acceleration, sparse observations, and noisy short-window velocity estimates. A natural alternative is to choose among several closed-form extrapolations and learn only a bounded correction, instead of producing coordinates entirely from a neural head.

PriMoTraj follows this route. Its encoder combines a Parallel Mixer (PM), using depthwise temporal convolution and pointwise feature mixing, with a gated residual from Multi-scale Decomposable Mixing (MDM). The prediction head builds seven deterministic motion priors from

the observed coordinates: constant-position, constant-velocity, smoothed-velocity, mixed-velocity, damped-velocity, constant-acceleration, and long-window smoothed-velocity. A sample-wise softmax gate averages these candidates, and a learned full-window residual head supplies a bounded per-step correction. The neural part is therefore used to select and adjust a motion prior, while the coordinate forecast remains tied to an explicit extrapolation.

The empirical picture is horizon-dependent, as previewed in Figure 1. On Porto with $T = 12$ and $H = 6$, PriMoTraj (11.1K parameters, no knowledge distillation) has the best ADE among input-matched baselines, 3.4 m below tuned LSTM-Seq2Seq-Attn while using less than half the parameters, and remains faster than PatchTST and TimeMixer in CPU batch-size-one online latency. At $H = 12$, the same design has 12.0K parameters and trails LSTM-Seq2Seq-Attn by 16.7 m in ADE. The main empirical claim is therefore anchored on short-horizon Porto GPS-derived forecasting; controlled Porto downsampling analyzes sampling-regime effects, while T-Drive and Argoverse 2 (AV2)-derived provide sparse and high-frequency supporting sampling regimes rather than broad cross-city evidence.

The contributions are threefold:

- We establish an input-matched GPS-derived evaluation that reports meter-level ADE/FDE, CPU batch-size-one online latency, parameter count, and forecast duration, while separating baselines that require road graphs, POI features, user histories, or pretraining.
- We introduce PriMoTraj, a compact predictor that combines a temporal encoder with a sample-wise gate over seven deterministic motion priors and a bounded full-window residual head.
- We analyze sampling-regime effects through fixed-step and fixed-duration Porto downsampling, with ablations and gate diagnostics showing how prior selection and residual correction contribute to the final forecast.

Code availability. The implementation, the preprocessing and evaluation pipeline, and the analysis scripts behind the statistical tables and the trade-off figure are available at https://github.com/David-Chen31/PriMoTraj.

II. Related Work

A. Forecasting Backbones

Modern time-series forecasting spans attention, decomposition, patching, inverted tokenization, and compact linear or mixing models [3]–[9]. These backbones provide useful baselines for short-window coordinate forecasting, but their usual normalized losses do not by themselves specify geographic displacement, online latency, or the available input context. For urban trajectories, this distinction matters because the same normalized coordinate error can correspond to different meter-level errors after inverse transformation.

B. Trajectory Prediction with External Context

Trajectory prediction often uses additional structure beyond GPS-derived inputs. Classical mobility studies and taxi destination prediction emphasize spatial continuity [2], [10], while graph- or attention-based models exploit road networks, transition structure, or user-history context [11]–[14]. Pretraining-based trajectory representation models such as START and TrajMamba learn from road-level or semantic contexts before downstream forecasting [15], [16]. These methods define a different input setting and are discussed separately in Appendix C.

C. Motion Priors and Lightweight Online Forecasting

Simple extrapolation remains a meaningful reference point in the GPS-derived input setting. Constant-position and constant-velocity predictors have no trainable parameters, yet they can outperform neural models when observations are sparse and recent velocity is stable [17]. They are also relevant operational baselines when batch-size-one latency and parameter count matter alongside ADE/FDE. This input setting is common in streaming scenarios where road matching, map construction, and semantic enrichment are unavailable or too costly for each query.

Classical trackers extend these assumptions with acceleration, turn-rate, or switching dynamics through Kalman filters and interacting multiple-model (IMM) filters [18]–[20]. PriMoTraj can be viewed as a constrained feed-forward conditional ensemble. Unlike a standard mixture-of-experts (MoE), its experts are parameter-free closed-form trajectory hypotheses rather than independently learned functions, and learning is restricted to sample-wise routing plus a bounded correction. Unlike an IMM, it makes one open-loop forecast: it maintains no recursive mode posterior or mode-conditioned filter state and performs no measurement update over the prediction horizon.

Closed-form dynamics, softmax routing, and residual learning are each established; the novelty we claim is their constrained combination under an input-matched GPS-derived protocol, not a new general MoE or switching-filter principle. Within that scope PriMoTraj occupies a narrow point: it keeps the closed-form priors of classical trackers, routes among them with a learned sample-wise gate, and adds a bounded residual instead of a full neural decoder. Distributional and multi-modal predictors, including mixture density and social forecasting models, address route ambiguity more directly [21], [22]; these probabilistic formulations answer a different question from the deterministic setting studied here.

III. Methodology

A. Problem Definition

Given an observed trajectory window

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{T \times F}, \tag{1}$$

where each step contains latitude, longitude, speed, heading features, and time features, the goal is to predict

$$\widehat{\mathbf{Y}} = [\widehat{\mathbf{y}}_1, \ldots, \widehat{\mathbf{y}}_H] \in \mathbb{R}^{H \times F}. \tag{2}$$

The geographic metrics are computed on the first two channels after inverse normalization. The main Porto short-history

experiments use $T = 12$ and $H \in \{6, 12\}$; longer-history and dataset-support experiments are treated as robustness or appendix evidence rather than as the basis of the short-history analysis.

PriMoTraj anchors the latitude-longitude prediction to an explicit bank of motion priors and learns both the prior weights and a bounded correction:

$$\begin{aligned}\widehat{\mathbf{Y}}_{1:2}(h) = \sum_i w_i(\mathbf{X})\mathbf{P}_i(h;\mathbf{X}) \\ + \boldsymbol{\lambda}_r(h)\tanh(R_\theta(\mathbf{X})(h)), \\ h = 1, \ldots, H.\end{aligned} \tag{3}$$

Here $\mathbf{P}_i$ are candidate motion priors, $w_i(\mathbf{X})$ are sample-wise gate weights, R_θ is the learned correction, and $\boldsymbol{\lambda}_r(h)$ is the positive residual scale at horizon step h.

B. Architecture Overview

Figure 2 summarizes the data flow. A compact temporal encoder maps the normalized input window to $\mathbf{Z}_o$ and a neural head produces a preliminary forecast $\mathbf{U}$. The preliminary head supplies non-coordinate target channels and auxiliary training signal; the final latitude-longitude output instead follows Eq. (3). Speed, heading, and time features enter the encoder and gate, while the priors remain coordinate-based. This split lets the learned part route and correct explicit extrapolations rather than generate coordinates directly, keeping the coordinate output tied to interpretable motion hypotheses.

C. Motion-Prior Gate

The motion-prior module operates on normalized latitude-longitude coordinates. Given $\mathbf{p}_T$, $\mathbf{v}_r = \mathbf{p}_T - \mathbf{p}_{T-1}$, $\mathbf{v}_p = \mathbf{p}_{T-1} - \mathbf{p}_{T-2}$, $\mathbf{v}_s = (\mathbf{p}_T - \mathbf{p}_{T-2})/2$, and $\mathbf{v}_\ell = (\mathbf{p}_T - \mathbf{p}_{T-4})/4$, the prior bank is defined as

$$\mathbf{P}_1(h) = \mathbf{p}_T, \tag{4}$$

$$\mathbf{P}_2(h) = \mathbf{p}_T + h\mathbf{v}_r, \tag{5}$$

$$\mathbf{P}_3(h) = \mathbf{p}_T + h\mathbf{v}_s, \tag{6}$$

$$\mathbf{P}_4(h) = \mathbf{p}_T + h(\gamma\mathbf{v}_r + (1-\gamma)\mathbf{v}_s), \tag{7}$$

$$\mathbf{P}_5(h) = \begin{cases} \mathbf{p}_T + \frac{1-\rho^h}{1-\rho}\mathbf{v}_r, & \rho \neq 1, \\ \mathbf{p}_T + h\mathbf{v}_r, & \rho = 1, \end{cases} \tag{8}$$

$$\mathbf{P}_6(h) = \mathbf{p}_T + h\mathbf{v}_r + \tfrac{1}{2}h(h-1)\,\mathbf{a}^\star, \tag{9}$$

$$\mathbf{P}_7(h) = \mathbf{p}_T + h\mathbf{v}_\ell, \tag{10}$$

where clipping is applied independently to each coordinate,

$$a_j^\star = \max(-|v_{r,j}|, \min(v_{r,j} - v_{p,j}, |v_{r,j}|)), \quad j \in \{\text{lat}, \text{lon}\}. \tag{11}$$

The operation is performed in dataset-normalized coordinate space after the internal RevIN inversion and before inverse standardization to latitude-longitude degrees. The seven priors cover stops, recent velocity, short-window smoothing, mixed velocity, damped motion, clipped constant acceleration, and long-window smoothing. In implementation, ρ is clipped to $[0, 0.99]$ for the damped prior.

Here "standalone" denotes a separately evaluated zero-parameter comparator. P4 mixes one- and two-step velocities, whereas standalone CVMix-3 mixes one- and three-step velocities; thus the two uses previously sharing "CVMix" are not the same predictor.

TABLE I
UNIFIED NAMES FOR INTERNAL PRIORS AND STANDALONE BASELINES. "NONE" MEANS NO FORMULA-IDENTICAL STANDALONE ROW IS REPORTED.

ID	Formal name	Fig. 4	Standalone relation
P1	Constant position	P1 CP	Same as Last
P2	Recent constant velocity	P2 CV-R	Same as CV
P3	Two-step smoothed velocity	P3 CV-S2	None
P4	Mixed 1/2-step velocity	P4 CV-M(1,2)	Not CVMix-3
P5	Damped recent velocity	P5 D-CV	None
P6	Clipped constant acceleration	P6 CA-C	Not standalone CA
P7	Four-step long-window velocity	P7 CV-L4	None

The gate is a two-layer multilayer perceptron (MLP) with configurable input horizon K_g; unless otherwise stated, $K_g = 3$. It flattens the last $\min(K_g, T)$ observed steps using all input channels, giving a Porto short-history default input dimension of $3 \times 9 = 27$. With hidden width h_g and Gaussian error linear unit (GELU) activation, the output layer produces n_p logits followed by a softmax:

$$\mathbf{w} = \text{softmax}(\phi(\mathbf{X}_{T-K_g+1:T})), \quad \phi : \mathbb{R}^{K_g F} \to \mathbb{R}^{h_g} \to \mathbb{R}^{n_p}. \tag{12}$$

The main Porto table uses one deployed gate configuration at both horizons: $h_g = 64$, $n_p = 7$ (2,247 gate parameters), $K_g = 3$, and one PM block ($N = 1$). The gate input horizon K_g is the only capacity knob in the routing module: the gate flattens the last $\min(K_g, T)$ steps, so its first-layer width grows linearly in K_g while the priors, residual head, and inference path are unchanged; the $K_g \in \{2, 3, 6, 12\}$ sweep is in Appendix D. The mixing and damping coefficients $\gamma = 0.5$ and $\rho = 0.8$ are fixed global hyperparameters; gate weights remain sample-wise and learned end-to-end. The residual bound $\boldsymbol{\lambda}_r \in \mathbb{R}^H = \text{softplus}(\tilde{\boldsymbol{\lambda}})$ is initialized to 0.15. The exact parameter budget (backbone and preliminary head / gate / residual head including $\boldsymbol{\lambda}_r$) is $1{,}120 + 2{,}247 + 7{,}762 = 11{,}129$ at $H = 6$ and $1{,}198 + 2{,}247 + 8{,}548 = 11{,}993$ at $H = 12$. The horizon-dependent difference comes from the preliminary and residual output widths. Appendix B maps each family of reported numbers to its configuration and evaluation population.

Let $\bar{\mathbf{P}}(h) = \sum_{i=1}^{n_p} w_i(\mathbf{X})\mathbf{P}_i(h)$ and let $R_\theta(\mathbf{X}) \in \mathbb{R}^{H\times 2}$ be a learned residual head. R_θ is an independent two-layer MLP $\mathbb{R}^{TF} \to \mathbb{R}^r \to \mathbb{R}^{2H}$ that reads the full input window (108-dim for $T = 12, F = 9$) with hidden width $r = 64$. Its final layer is zero-initialized so training starts from pure prior weighting, and the residual is bounded per step by the learned positive scale $\boldsymbol{\lambda}_r = \text{softplus}(\tilde{\boldsymbol{\lambda}})$ defined above:

$$\widehat{\mathbf{Y}}_{1:2}(h) = \bar{\mathbf{P}}(h) + \boldsymbol{\lambda}_r(h)\ \tanh(R_\theta(\mathbf{X})(h)). \tag{13}$$

The learned per-step scale allows larger corrections at horizons where the prior error is larger.

The prior bank and bounded residual are applied only to latitude and longitude. The neural head still outputs all F target channels, including speed, heading, and time-derived channels, and the normalized regression loss is computed over

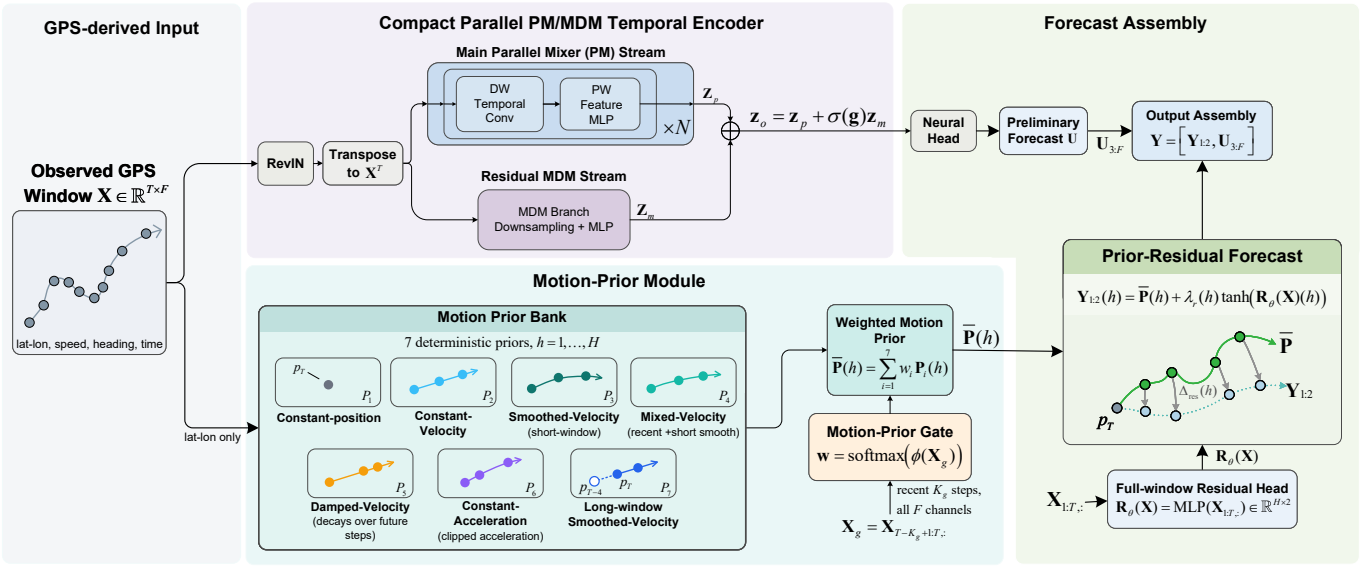

Fig. 2. Overview of PriMoTraj. The temporal encoder processes the normalized GPS window, while a motion-prior bank builds deterministic coordinate extrapolations from recent latitude-longitude history. A lightweight gate averages the priors, and a bounded residual head corrects the coordinate forecast. The same deployed architecture is used at $H = 6$ and $H = 12$ on Porto.

all channels. Auxiliary channels provide training signal to the encoder and gate, while ADE/FDE and the final coordinate forecast are evaluated only on inverse-normalized coordinate channels. No future auxiliary feature is used as model input.

D. Compact Temporal Encoder

The encoder uses a compact PM main path with an MDM context branch inspired by multi-scale decomposable mixing [5]. Reversible instance normalization (RevIN) is applied before feature mixing and inverted before the motion-prior module [23]. Let $\mathbf{X}^\top \in \mathbb{R}^{F\times T}$ denote the transposed normalized input. The MDM branch produces

$$\mathbf{Z}_m = \mathrm{MDM}(\mathbf{X}^\top), \tag{14}$$

and the main path applies $N = 1$ PM blocks to the original coordinate-preserving representation:

$$\mathbf{Z}_p = \mathrm{PM}^{(N)}(\mathbf{X}^\top). \tag{15}$$

Each PM block combines a depthwise temporal convolution with a pointwise feature MLP. Here DWConv/BN/Drop/PWMLP denote depthwise convolution, batch normalization, dropout, and pointwise MLP:

$$\mathbf{H}_t = \mathrm{Drop}(\mathrm{GELU}(\mathrm{DWConv}(\mathrm{BN}(\mathbf{Z})))), \tag{16}$$

$$\mathbf{Z}' = \mathbf{Z} + \mathbf{H}_t, \tag{17}$$

$$\mathbf{H}_f = \mathrm{PWMLP}(\mathrm{BN}(\mathbf{Z}')^\top)^\top, \tag{18}$$

$$\mathrm{PM}(\mathbf{Z}) = \mathbf{Z}' + \alpha\mathbf{H}_f. \tag{19}$$

The MDM context is added through a scalar gate:

$$\mathbf{Z}_o = \mathbf{Z}_p + \sigma(g)\mathbf{Z}_m. \tag{20}$$

Here $g \in \mathbb{R}$ is a learned scalar initialized to zero, so $\sigma(g) = 0.5$ at initialization. PM-only removes the MDM branch and is therefore prediction-equivalent to a zero MDM contribution, although the ablation is retrained rather than produced by clamping a deployed checkpoint. PM remains the coordinate-preserving stream, while MDM supplies residual multi-scale context.

E. Training Objective

The base objective is normalized mean squared error (MSE) over all target channels:

$$\mathcal{L}_{\mathrm{norm}} = \frac{1}{BHF}\sum_{b=1}^{B}\sum_{h=1}^{H}\|\widehat{\mathbf{Y}}_{b,h} - \mathbf{Y}_{b,h}\|_2^2. \tag{21}$$

The deployed configuration, used at both $H = 6$ and $H = 12$ on Porto, down-weights normalized MSE and adds a Haversine-FDE term for the last predicted coordinate:

$$\mathcal{L} = \beta_{\mathrm{mse}}\mathcal{L}_{\mathrm{norm}} + \beta_{\mathrm{fde}}\frac{1}{B}\sum_{b=1}^{B}\frac{d_{\mathrm{geo}}(\widehat{\mathbf{y}}_{b,H}, \mathbf{y}_{b,H})}{1000}, \tag{22}$$

with $\beta_{\mathrm{mse}} = 0.1$ and $\beta_{\mathrm{fde}} = 1.0$; d_{geo} is Haversine distance after inverse scaling. The deployed recipe uses five epochs, batch size 2048, AdamW with base learning rate 10^{-3} and weight decay 10^{-4}, gradient-norm clipping at 1.0, cosine annealing to $\eta_{\min} = 10^{-6}$, and checkpoint selection by validation ADE in meters. An optional KD term against a frozen LSTM-Seq2Seq-Attn teacher (latitude-longitude MSE, two-epoch warmup, weight $\beta_{\mathrm{KD}} = 0.5$) leaves inference unchanged; deployed checkpoints use KD off.

The two loss terms have different direct scopes: normalized MSE supervises every target channel and step, whereas Haversine-FDE acts only on the last geographic point. Its gradient nevertheless updates the shared encoder, gate, and residual head used by all steps, so it can indirectly change ADE. We did not perform a controlled loss-weight sweep and therefore do not claim insensitivity to the fixed 0.1/1.0 weighting. The weaker $H = 6$ FDE and $H = 12$ results also

TABLE II
DATASETS USED IN THE FORMAL EXPERIMENTS.

Dataset	Points	Trips	Step	Split
Porto [10]	72.7M	1.40M	15 s	trip 70/10/20
AV2-derived [24]	1.01M	9.16K	0.1 s	scenario 70/10/20
T-Drive [25]	0.68M	7.73K	300 s	segment 70/10/20

show that adding the FDE term does not guarantee endpoint superiority.

IV. EXPERIMENTS

The experiments test the accuracy-size trade-off, sampling-regime effects, component contributions, latency, and supporting sampling regimes under the input-matched GPS-derived setting.

A. Datasets and Input Protocol

Table II lists the three datasets used in the experiments. Porto is the primary benchmark, with latitude-longitude coordinates sampled every 15 seconds. T-Drive is segmented into fixed 5-minute intervals and provides sparse sampling-regime support. For AV2-derived, focal trajectories are converted from local meter coordinates to proxy degrees so the same coordinate pipeline can compute meter-level displacement; these results are used only as high-frequency sampling-regime support and are not official AV2 leaderboard numbers.

For the main comparison, all Porto models use the same 9-dimensional GPS/history-derived/time feature vector: latitude, longitude, speed, heading sine/cosine, hour sine/cosine, and day-of-week sine/cosine. Methods requiring road graphs, POI, road embeddings, or pretraining are excluded from the main table. Porto and T-Drive use trip-level splits; AV2-derived uses a scenario-sorted derived split.

For Porto downsampling, each variant is produced within trips, preserving the original split assignment and recomputing normalization statistics from training only. Speed and heading are recomputed after downsampling; trips too short for a given (T, H) are removed and retained-sample counts are tracked.

B. Baselines, Metrics, and Statistical Protocol

The main Porto comparison covers Last, constant velocity (CV), fixed velocity mixture (CVMix), constant-velocity Kalman filter (CV-KF), long short-term memory sequence-to-sequence with attention (LSTM-Seq2Seq-Attn) [26], [27], DLinear, PatchTST, TimeMixer, and PriMoTraj. CV-KF uses a normalized state with process noise, observation noise, and damping selected on the validation split [18]. Deterministic constant-acceleration (CA) and coordinated-turn (CT) baselines test richer physics using the same coordinate history. PatchTST and TimeMixer serve as representative short-window time-series backbones in the main table; iTransformer and AMD are cited as recent forecasting context rather than numerical baselines. START, TrajMamba, and HGT-RN are discussed as protocol-separated methods rather than numerical baselines because their original formulations use road graphs, road/POI semantics, traffic-flow graphs, or pretraining [15], [16], [28].

TABLE III
RANK MISMATCH BETWEEN NORMALIZED MSE AND METER-LEVEL ADE. TOP-3 OVERLAP COMPARES THE THREE BEST MODELS BY NORMALIZED MSE WITH THE THREE BEST MODELS BY ADE.

Dataset	H	Spearman ρ	Kendall τ	Top-3 overlap	Worst rank gap
AV2-derived	6	-0.542	-0.429	0/3	CV: 13→1
AV2-derived	12	-0.552	-0.451	0/3	CV: 14→1
AV2-derived	24	-0.486	-0.385	0/3	CV: 14→1
Porto-15s-s12	6	0.550	0.389	2/3	DLinear: 5→9
Porto-15s-s12	12	0.683	0.500	2/3	DLinear: 5→9
T-Drive-s6	3	-0.633	-0.389	0/3	LSTM: 1→9
T-Drive-s6	6	-0.333	-0.222	0/3	LSTM: 1→9
T-Drive-s6	12	-0.083	0.000	1/3	LSTM: 1→9

ADE and FDE are computed in meters after inverse normalization. For test windows $b = 1, \ldots, B$ and Haversine distance d_{geo} on the coordinate channels,

$$\text{ADE} = \frac{1}{BH} \sum_{b=1}^{B} \sum_{h=1}^{H} d_{\text{geo}}(\widehat{\mathbf{y}}_{b,h}^{1:2}, \mathbf{y}_{b,h}^{1:2}), \tag{23}$$

$$\text{FDE} = \frac{1}{B} \sum_{b=1}^{B} d_{\text{geo}}(\widehat{\mathbf{y}}_{b,H}^{1:2}, \mathbf{y}_{b,H}^{1:2}). \tag{24}$$

Shared GPU batch latency gives throughput context, while CPU batch-size-one latency is measured end-to-end with 20 warmup and 100 timed repetitions, reporting p50 and p95 rather than mean±std, because batch-size-one CPU timings are sensitive to cold-cache and scheduling outliers that inflate the standard deviation without changing the percentile latency relevant to deployment. Neural baselines use seeds 2024, 2025, and 2026 with a common validation-loss checkpoint rule; deterministic baselines are profiled through the same script. Neural search spaces are in Table XII; CV-KF process noise, observation noise, and damping are tuned on the validation split. The trip-level paired bootstrap uses 131,072 matched Porto windows (119,110 trips at $H = 6$, 114,208 at $H = 12$) and 5,000 resamples, with whole trips drawn with replacement so that within-trip correlation is not counted as independent evidence.

C. Metric Alignment Before Model Comparison

Because normalized MSE can mis-rank meter-level ADE, we present the rank diagnostic before interpreting model comparisons. Table III reports rank agreement across all evaluated regimes. Porto is positively but imperfectly aligned, and DLinear still moves from middle MSE rank to worst ADE rank; AV2-derived and T-Drive are negatively correlated, with top-3 overlap falling to 0/3. We therefore select PriMoTraj checkpoints by validation ADE and report test ADE/FDE after inverse transform.

D. Input-Matched Porto Results

The main Porto comparison asks whether the prior-residual design changes the accuracy-size trade-off under the input-matched GPS-derived setting. At $H = 6$, PriMoTraj (11.1K parameters; residual head over the full input window, expanded prior bank, learned per-step bound; trained without knowledge distillation) has the lowest ADE among input-matched methods, 3.4 m below LSTM-Seq2Seq-Attn (234.082 ± 0.181 vs

TABLE IV

TRIP-LEVEL PAIRED BOOTSTRAP ON MATCHED PORTO TEST WINDOWS. PRIMOTRAJ AND LSTM-SEQ2SEQ-ATTN ERRORS ARE AVERAGED ACROSS 3 SEEDS BEFORE BOOTSTRAPPING; OTHER BASELINES USE SEED 2026. ΔADE = ADE(BASELINE) $-$ ADE(PRIMOTRAJ), SO POSITIVE FAVORS PRIMOTRAJ. WINDOW/TRIP COUNTS, THE RESAMPLING SCHEME, AND THE p-VALUE DEFINITION ARE IN SEC. IV-B.

Comparison	H	ΔADE (m)	Tail	p-value
PriMoTraj vs LSTM-Seq2Seq-Attn	6	$+3.6$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs LSTM-Seq2Seq-Attn	12	-16.6	0/5000	$< 4\times10^{-4}$
PriMoTraj vs TimeMixer	6	$+41.1$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs TimeMixer	12	$+71.7$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs CV-KF	6	$+45.9$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs CV-KF	12	$+90.6$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs PatchTST	6	$+78.8$	0/5000	$< 4\times10^{-4}$
PriMoTraj vs PatchTST	12	$+112.7$	0/5000	$< 4\times10^{-4}$

237.498 $\pm$ 2.383), while using less than half its parameter count. The FDE of PriMoTraj is 11.1 m higher than the LSTM-Seq2Seq-Attn FDE. At $H = 12$, the same design with horizon-dependent output width has 12.0K parameters, but LSTM-Seq2Seq-Attn has lower ADE, with PriMoTraj trailing by 16.7 m (432.971 $\pm$ 0.434 vs 416.262 $\pm$ 2.636, 3 seeds) at roughly half the parameter count. Across both horizons, PriMoTraj has lower ADE than PatchTST, TimeMixer, CV-KF, CVMix, CV, CA, CT, DLinear, and Last under the same input-matched setting.

The paired bootstrap in Table IV uses the same 131,072 matched Porto test windows. For PriMoTraj and LSTM-Seq2Seq-Attn, per-window errors are averaged across the three training seeds before trips are resampled. The $H = 6$ delta is $+3.6$ m and the $H = 12$ delta is -16.6 m. All observed two-sided tail counts are zero at $N_{\text{boot}} = 5000$, so every value reaches the same conservative reporting floor $p < 4 \times 10^{-4}$. That floor reflects the resampling budget, not the effect size, so Table IV reports the opposite-tail resample count alongside each p-value and should be read from the signed ΔADE column, which separates the $+3.6$ m gap from the $+112.7$ m one. Raising the budget separates them: at $N_{\text{boot}} = 50{,}000$ on the released checkpoints, 87 of 50,000 trip resamples fall in the opposite tail for the LSTM-Seq2Seq-Attn contrast ($p \approx 3.5 \times 10^{-3}$) while the other three stay at the floor, and the 95% trip-level intervals are 1.1–1.9 m wide at $H = 6$ and 1.9–3.1 m at $H = 12$. The released analysis script reports both at a configurable N_{boot}.

Tables V and IV therefore support a short-horizon ADE advantage for PriMoTraj at 11.1K parameters. The two ADE comparisons are also separated by more than the cross-seed spread: at $H = 6$ the $\pm$one-standard-deviation intervals $[233.90, 234.26]$ for PriMoTraj and $[235.12, 239.88]$ for LSTM-Seq2Seq-Attn are disjoint by 0.85 m, and at $H = 12$ the intervals $[432.54, 433.41]$ and $[413.63, 418.90]$ are disjoint by 13.64 m in the opposite direction. The 3.4 m short-horizon gap therefore survives seed variation, although it remains small in absolute terms. LSTM-Seq2Seq-Attn still has the best FDE at $H = 6$ and both the best ADE and FDE at $H = 12$, so parameter count alone does not explain the longer-horizon gap.

Both the FDE gap and its growth with the horizon follow from how the two designs allocate capacity over the horizon.

TABLE V

PORTO RESULTS IN THE INPUT-MATCHED GPS-DERIVED SETTING, $T = 12$. UNPREFIXED LAST/CV/CVMIX/CA/CT DENOTE STANDALONE COMPARATORS; INTERNAL PRIORS USE P1–P7 (TABLE I). NEURAL ROWS REPORT THREE SEEDS; DETERMINISTIC ROWS HAVE NO TRAINABLE PARAMETERS. CPU BATCH-SIZE-ONE LATENCY IS IN TABLE VII.

H	Model	Seeds	Params	GPU ms	ADE (m)	FDE (m)
6	Last	1	0	0.095	397.789	648.866
	CV	1	0	0.107	328.645	595.177
	CVMix	1	0	0.192	312.892	557.591
	CV-KF	1	0	2.099	279.701	475.520
	LSTM-Seq2Seq-Attn	3	22.8K	5.376	237.498	**377.651**
	DLinear	3	156	3.299	674.118	963.453
	PatchTST	3	268.5K	19.681	322.952	520.180
	TimeMixer	3	34.5K	20.434	274.271	460.354
	PriMoTraj	3	11.1K	2.828	**234.082**	388.783
12	Last	1	0	0.096	714.388	1227.421
	CV	1	0	0.110	669.746	1313.346
	CVMix	1	0	0.192	626.660	1216.448
	CV-KF	1	0	2.190	524.681	948.279
	LSTM-Seq2Seq-Attn	3	26.3K	5.730	**416.262**	**729.387**
	DLinear	3	312	4.163	998.189	1536.968
	PatchTST	3	270.0K	24.218	561.076	981.540
	TimeMixer	3	34.7K	27.950	504.375	904.774
	PriMoTraj	3	12.0K	3.386	432.971	764.117

TABLE VI

LATENCY PROFILING ENVIRONMENT.

Item	Value
CPU	AMD EPYC 9754, 128 cores / 256 threads
GPU	NVIDIA GeForce RTX 4090 D, 24 GB
OS	Linux 5.15.0-94-generic
Python	3.8.10
PyTorch	2.4.1+cu121
CUDA driver/runtime	13.0 driver / 12.1 runtime
CPU latency protocol	batch 1, end-to-end, 20 warmup, 100 repeats

Every prior in the bank extrapolates forward from $\mathbf{p}_T$, so its error is smallest at the first predicted step and grows with the step index: ADE averages over all H steps and is weighted towards the steps the priors fit best, whereas FDE is read only at step H, where the prior is weakest. PriMoTraj can correct that step only within the learned bound $\boldsymbol{\lambda}_r$, which matches short futures well but offers less corrective freedom than a fully recurrent decoder once the prior error grows. At $H = 12$, the longer extrapolation magnifies small velocity-estimation errors, where the stepwise generation of LSTM-Seq2Seq-Attn is more flexible. This reading follows from the construction rather than from a per-step error decomposition, which we did not run.

Table VI gives the profiling environment, and Table VII reports CPU batch-size-one online latency. LSTM-Seq2Seq-Attn is fastest in this profile despite its 22.8–26.3K parameter count; PriMoTraj is slower than LSTM-Seq2Seq-Attn but below PatchTST/TimeMixer in p50 latency at both horizons (≈ 1.1 ms end-to-end).

E. Richer GPS-Derived Physics Baselines

Deterministic CA and CT baselines in Table VIII test whether the motion-prior bank is too limited. CA uses the last three coordinates to estimate acceleration with norm clipping. CT estimates a signed turn rate from the last two displacements, clips the turn rate, and falls back to straight motion

TABLE VII
CPU BATCH-SIZE-ONE ONLINE LATENCY ON PORTO $T = 12$. VALUES ARE P50/P95 MILLISECONDS OVER 100 TIMED REPETITIONS AFTER 20 WARMUP RUNS.

H	Model	p50	p95
6	Last	0.035	0.038
6	CV	0.043	0.046
6	CVMix	0.078	0.086
6	CV-KF	0.755	0.862
6	LSTM-Seq2Seq-Attn	**0.211**	**0.223**
6	DLinear	0.102	0.114
6	PatchTST	1.323	66.059
6	TimeMixer	1.981	2.596
6	PriMoTraj	1.104	1.152
12	Last	0.036	0.042
12	CV	0.043	0.046
12	CVMix	0.078	0.086
12	CV-KF	0.740	0.857
12	LSTM-Seq2Seq-Attn	**0.216**	**0.348**
12	DLinear	0.104	0.108
12	PatchTST	1.322	73.564
12	TimeMixer	1.967	58.586
12	PriMoTraj	1.138	1.165

TABLE VIII
RICHER DETERMINISTIC GPS-DERIVED PHYSICS BASELINES ON PORTO $T = 12$. LOWER ADE/FDE IS BETTER.

H	Model	Accel	Turn	ADE (m)	FDE (m)
6	CA	yes	no	700.961	1548.587
6	CT	no	yes	**408.591**	**733.966**
12	CA	yes	no	2189.796	5562.549
12	CT	no	yes	**781.810**	**1433.833**

at very low normalized speed. Both baselines use only GPS-derived coordinate history and have no learned parameters.

More expressive deterministic dynamics are not automatically more robust under noisy short-window GPS. CA performs poorly because second-order differences amplify observation noise, especially at $H = 12$. CT is more stable than CA but still trails CV-KF and PriMoTraj. Simple stable priors therefore form the default bank, while acceleration-aware or curved priors are treated as validation-selected extensions.

F. Sampling-Regime Analysis

The Porto downsampling study in Table IX separates fixed-step and fixed-duration effects. Following the definition in the introduction, each variant is evaluated under an explicitly stated sampling interval and forecast duration; PriMoTraj is not trained as a single interval-conditioned model across all regimes. Protocol A fixes the number of future steps ($H = 6$), so forecast duration grows from 90 s at 15 s sampling to 30 min at 300 s sampling. Protocol B fixes 3 min and 6 min futures across 15 s, 30 s, and 60 s variants. All variants preserve the original trip-level split assignment and recompute train-only normalization statistics.

Under fixed-step forecasting, increasing the sampling interval eventually changes the best fixed prior from CV-KF to Last as the forecast duration grows and recent-velocity extrapolation becomes less reliable. PriMoTraj has lower ADE than the best fixed prior from 15–120 s; at 300 s, where the retained sample shrinks to a few thousand trips with a dominant stop ratio, the deterministic Last prior is lower. When the future duration is fixed, CV-KF is the best fixed prior across 15–60 s and PriMoTraj has lower ADE. Retained-sample statistics also matter: the 60 s variant keeps 249K trips while 300 s keeps only 2.6K trips with a much higher stop ratio.

TABLE IX
PORTO DOWNSAMPLING SUMMARY. PROTOCOL A FIXES FUTURE STEPS; PROTOCOL B FIXES FORECAST DURATION. 'BEST PRIOR" IS THE BEST OF LAST, CV, CVMIX, AND CV-KF.

Protocol	Setting	Interval	H	Best prior ADE	PriMoTraj ADE
A	fixed $H = 6$	15 s	6	279.701 (CV-KF)	**234.082**
	fixed $H = 6$	30 s	6	564.228 (CV-KF)	**456.829**
	fixed $H = 6$	60 s	6	929.771 (CV-KF)	**803.988**
	fixed $H = 6$	120 s	6	1047.410 (Last)	**948.547**
	fixed $H = 6$	300 s	6	**1001.197** (Last)	1529.113
B	3 min future	15 s	12	524.681 (CV-KF)	**432.971**
	3 min future	30 s	6	564.228 (CV-KF)	**456.829**
	3 min future	60 s	3	582.133 (CV-KF)	**490.455**
B	6 min future	15 s	24	994.430 (CV-KF)	**820.179**
	6 min future	30 s	12	983.052 (CV-KF)	**810.341**
	6 min future	60 s	6	929.771 (CV-KF)	**803.988**

The shift from CV-KF to Last in Protocol A illustrates how the preferred motion prior changes with the sampling regime. The sample-wise gate is meant to absorb this regime dependence within each setting by spreading weight across the seven priors; the learned allocation in Fig. 4 follows the empirically best prior as the interval changes. The 300 s case marks the sparse, stop-heavy limit, where the parameter-free Last prior wins.

G. Accuracy-Efficiency Trade-off

Figure 3 summarizes the ADE-efficiency trade-off for the Porto comparison. At $H = 6$, PriMoTraj has lower ADE than LSTM-Seq2Seq-Attn while using $\sim$half its parameters; at $H = 12$, LSTM-Seq2Seq-Attn has lower ADE. For GPU batch throughput, PriMoTraj has the lowest latency among learned models in Table V (≈ 2.8–3.4 ms at batch 2048), about $2\times$ faster than LSTM-Seq2Seq-Attn and 7–$10\times$ faster than PatchTST and TimeMixer. Table VII gives the complementary CPU batch-size-one profile, where LSTM-Seq2Seq-Attn has the lowest p50 latency. The two profiles favor different operating points: batched GPU inference favors PriMoTraj among learned models, whereas strictly serial CPU inference favors LSTM-Seq2Seq-Attn. The two latency views are therefore reported separately.

H. Core Ablation

The ablation in Table X uses the deployed configuration trained from scratch (no KD) and separates the gate, prior bank, bounded residual, and LSTM-Seq2Seq-Attn knowledge-distillation (KD) signal. Adding the KD term ($\beta_{\mathrm{KD}} = 0.5$) keeps ADE within 1 m at both horizons (3-seed mean delta $+0.83$ m at $H = 6$ and $+0.81$ m at $H = 12$, both inside the cross-seed standard deviation of the deployed configuration), with the no-KD mean slightly lower. The full model is lower than a fixed CVMix prior by 78.8 m at $H = 6$ and 193.7 m at $H = 12$, so a fixed velocity mixture is not sufficient here. Replacing the entire prior bank with the neural backbone alone ("no prior") costs 34.3 m at $H = 6$ and 68.8 m at $H = 12$; removing only the learned residual head while keeping the

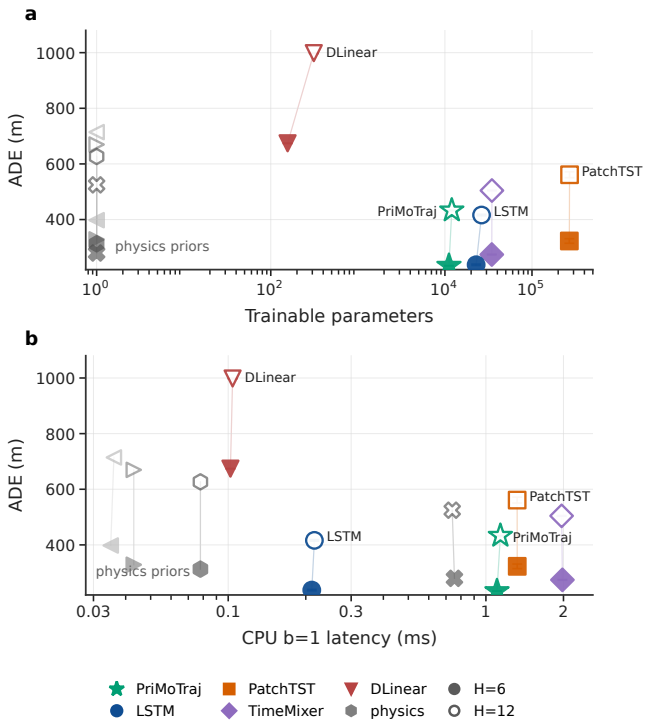

Fig. 3. Porto $T = 12$ ADE-parameter/GPU-batch-latency trade-off for input-matched models. Lower-left is better. Vertical bars show $\pm$ one standard deviation over three seeds for neural models; deterministic baselines have no seed error bar. Filled markers: $H = 6$; hollow: $H = 12$.

prior bank costs 21.1 m at $H = 6$ and 47.5 m at $H = 12$. Both components affect ADE, with the larger ablation gap coming from the prior bank. The "no prior" row removes the routing structure together with the experts, so it cannot separate the two. We therefore ran a control that holds the encoder, the gate, the anchoring at $\mathbf{p}_T$ and the parameter budget fixed (11,100 and 11,915 against 11,129 and 11,993) and replaces the seven closed-form priors with seven trainable heads over a shared trunk. At the 15 s interval the learned head is ahead by 10.6 m at $H = 6$ and 18.8 m at $H = 12$, with 95% trip-level intervals $[10.3, 11.0]$ and $[18.2, 19.4]$; restoring the bounded residual to it at the same budget changes this by under 0.6 m.

Transferred without retraining, that advantage does not hold. The paired difference is $+0.1$ and -16.7 m at 30 s and -8.1 and $+24.2$ m at 60 s for $H = 6$ and $H = 12$: one win, two losses and one tie, with the learned error growing by a larger factor in all four ($4.24\times$ against $4.01\times$ at 60 s, $H = 6$). The priors read velocity and acceleration from the input window and so follow the interval, while the learned experts fit the displacement scale they saw in training. The deterministic bank buys regime-independence and interpretable routing weights rather than accuracy at a fixed interval.

Two further ablation families in Appendix D locate the remaining design choices. Varying the gate input horizon K_g shifts ADE monotonically but by only a few meters at a $1.5\times$ parameter cost, and removing RevIN, MDM, or the PM-input path leaves ADE within 1.1 m at $H = 6$ and 0.7 m at $H = 12$ (Table XIV). Both effects are substantially smaller than the

TABLE X
CORE PORTO ABLATION WITH $T = 12$. ΔADE IS RELATIVE TO FULL PRIMOTRAJ.

H	Variant	ADE (m)	ΔADE
6	Full model (no KD)	**234.082**	**0.000**
	+ KD	234.910	0.828
	Standalone CVMix-3	312.892	78.810
	Standalone Last	397.789	163.707
	Standalone CV	328.645	94.563
	No prior	268.354	34.272
	Gate no residual	255.211	21.129
12	Full model (no KD)	**432.971**	**0.000**
	+ KD	433.778	0.807
	Standalone CVMix-3	626.660	193.689
	Standalone Last	714.388	281.417
	Standalone CV	669.746	236.775
	No prior	501.729	68.758
	Gate no residual	480.443	47.472

TABLE XI
GATE SANITY ON PORTO $T = 12$, SEED 2026, DEPLOYED CONFIGURATION ON THE 131,072-WINDOW MATCHED SUBSET.

Metric	$H = 6$	$H = 12$
Gate entropy	1.116	1.164
Top-gate/oracle agreement	0.209	0.189
Uniform gate ADE	309.997	624.747
Weighted prior ADE	305.665	561.735
Final model ADE	**247.086**	**450.612**

prior-bank and residual-head gaps, so the deterministic prior bank and the bounded residual head carry the forecast while the compact encoder acts as a light, largely swappable context provider.

I. Gate Sanity

Table XI is a diagnostic on one seed and a matched subset, not a second estimate of the full-test mean in Table V. Its final-model ADE is higher by 13.004 m at $H = 6$ (247.086 vs 234.082) and 17.641 m at $H = 12$ (450.612 vs 432.971). The subset preserves the same relative PriMoTraj/LSTM ordering at both horizons, but we did not establish whether particular trip types cause its higher absolute error. Top-prior/oracle agreement is moderate because oracle-best priors are sample-specific and often close in ADE. Within this fixed subset, weighted priors improve over uniform mixing and the bounded residual improves further.

At longer sampling intervals, the higher entropy suggests that sparse trajectories do not induce a single dominant prior; the model spreads probability across position-preserving and damped or hybrid priors, consistent with the ambiguity introduced by longer gaps.

J. Supporting Sampling Regimes

T-Drive and AV2-derived are supporting sampling regimes rather than the basis of the main claim. On T-Drive with $T = 6$, Last is competitive under sparse 5-minute sampling, so PriMoTraj does not win every regime. On AV2-derived ($T = 50$, $H = 10, 30, 50$, i.e. 1/3/5 s futures), CV has the best deterministic-prior ADE at 0.248/1.354/3.196 m; this benchmark uses proxy coordinates from Argoverse 2 and is not an official leaderboard protocol.

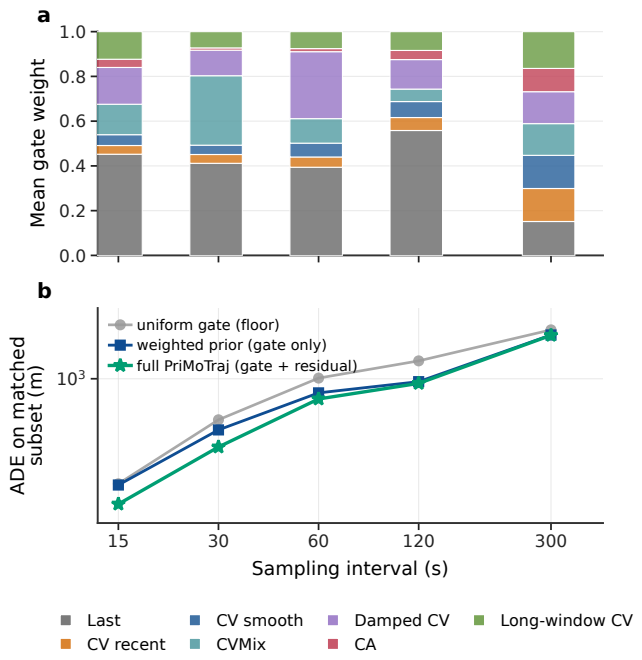

Fig. 4. Gate allocation and nested predictor ADE across Porto downsampling intervals. Panel (a) uses the P1–P7 labels mapped in Table I; panel (b) compares uniform prior mixing, gate-weighted priors, and the full forecast on the 131,072-window matched subset.

K. Protocol-Separated Extended Baselines

START, TrajMamba, and HGT-RN rely on road networks, POI/road semantics, transition information, traffic-flow graphs, or pretraining [15], [16], [28]. They are therefore kept outside the main table, and adapted GPS-derived encoder-plus-head variants are discussed as protocol notes in Appendix C.

L. Limitations

At $H = 6$ on Porto, PriMoTraj has the best ADE among input-matched baselines (Table V); LSTM-Seq2Seq-Attn keeps the best FDE at $H = 6$, the best ADE/FDE at $H = 12$, and the fastest CPU batch-size-one online latency. The result is therefore a short-horizon ADE and parameter-count trade-off (PriMoTraj uses 11.1K vs 22.8K parameters at $H = 6$; 12.0K vs 26.3K at $H = 12$), with LSTM-Seq2Seq-Attn remaining preferable in the longer-horizon Porto case. The deployed checkpoints are trained from scratch with no recurrent teacher; adding a frozen LSTM-Seq2Seq-Attn teacher with $\beta_{\mathrm{KD}} = 0.5$ changes ADE by less than 1 m at both horizons in Table X.

The GPS-derived input setting excludes map or road-graph information, POI features, and pretrained trajectory representations. Main benchmarking is anchored on Porto; T-Drive and AV2-derived are supporting sampling regimes, with proxy coordinates in the AV2 case. We also do not include a parameter-matched MoE with learned expert heads; consequently, the evidence does not establish that closed-form experts are generally superior to learned experts. That direct comparison, broader cross-city validation, time-aware definitions for irregular sampling, and calibrated mixture heads remain future work.

V. Conclusion

PriMoTraj decomposes forecasting into a weighted motion prior and a bounded learned residual. The main evidence comes from the Porto short-horizon GPS-derived setting: at $H = 6$, PriMoTraj achieves the best ADE among input-matched baselines with 11.1K parameters, 3.4 m better than LSTM-Seq2Seq-Attn; at $H = 12$, the same design has 12.0K parameters and trails LSTM-Seq2Seq-Attn by 16.7 m.

The downsampling and gate analyses show that the preferred motion prior changes with the sampling regime. For lightweight GPS-derived forecasting, input assumptions, forecast duration, latency, and meter-level ADE/FDE should be reported together rather than inferred from normalized loss or mismatched comparisons.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 92567204 and No. 62472196.

References

[1] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, 2022.

[2] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, "Artificial neural networks applied to taxi destination prediction," in *Proceedings of the ECML/PKDD 2015 Discovery Challenges*, 2015.

[3] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[4] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

[5] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "TimeMixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations*, 2024.

[6] Y. Hu, P. Liu, P. Zhu, D. Cheng, and T. Dai, "Adaptive multi-scale decomposition framework for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17 359–17 367.

[7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[8] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 22 419–22 430.

[9] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[10] L. Moreira-Matias, M. Ferreira, and J. Mendes-Moreira, "Taxi service trajectory - prediction challenge, ECML PKDD 2015," 2015, dataset artifact. [Online]. Available: https://archive.ics.uci.edu/dataset/339/taxi+service+trajectory+prediction+challenge+ecml+pkdd+2015

[11] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[12] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1459–1468.

[13] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[14] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1907–1913.

[15] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, and J. Wang, "START: Self-supervised trajectory representation learning with temporal regularities and travel semantics," in *2023 IEEE 39th International Conference on Data Engineering*, 2023.

[16] Y. Liu, Y. Lin, S. Guo, Z. Zhou, Y. Lin, and H. Wan, "TrajMamba: An efficient and semantic-rich vehicle trajectory pre-training model," in *Advances in Neural Information Processing Systems*, 2025.

[17] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[19] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2001.

[21] C. M. Bishop, "Mixture density networks," Aston University, Tech. Rep. NCRG/94/004, 1994.

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

[23] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible Instance Normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2022.

[24] 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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. [Online]. Available: https://www.argoverse.org/av2.html

[25] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-Drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 99–108. [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[28] J. Zhang, L. Guo, G. Wang, J. Yu, X. Zheng, Y. Mei, and B. Han, "A dual-level graph attention network and transformer for enhanced trajectory prediction under road network constraints," *Expert Systems with Applications*, vol. 261, p. 125510, 2025.

APPENDIX A
HYPERPARAMETER SEARCH SPACE

Neural baselines are selected on the validation split before three-seed evaluation. The short-window tuning budget includes smaller-capacity variants in addition to the default configurations, as summarized below; the selected configuration for each neural baseline is then retrained with seeds 2024, 2025, and 2026 under the common data split and metric pipeline.

TABLE XII
VALIDATION SEARCH SPACE FOR NEURAL BASELINES.

Model	Validation search space
LSTM-Seq2Seq-Attn	Recurrent hidden size, recurrent layers, dropout, and learning rate within the short-window candidate configurations.
PatchTST	Patch lengths $\{2, 3, 4, 6\}$, strides $\{1, 2\}$, $d_{\text{model}} \in \{16, 32, 64\}$, heads $\{2, 4\}$, and encoder layers $\{1, 2\}$.
TimeMixer	$d_{\text{model}} \in \{16, 32, 64\}$, encoder layers $\{1, 2\}$, downsampling layers $\{1, 2\}$, and moving-average kernels $\{3, 5\}$.

TABLE XIII
CONFIGURATION MAP. FW/PER-STEP DENOTES THE FULL-WINDOW RESIDUAL HEAD WITH A LEARNED PER-STEP BOUND.

Evidence	Model/training	Seeds and population
Main, core ablation	7 priors, FW/per-step, FDE loss; no KD	3 seeds, full test
Gate sanity	same deployed model	seed 2026, matched 131K
Bootstrap	same deployed model	3-seed mean for ours/LSTM; matched 131K
Gate/backbone ablations	deployed architecture; KD on	3 seeds, full test

TABLE XIV
GATE-HORIZON (UPPER BLOCK) AND BACKBONE-COMPONENT (LOWER BLOCK, $K_g = 3$) ABLATIONS, PORTO $T = 12$, MEAN $\pm$ STD OVER SEEDS 2024–2026, KD ON.

Variant	Params	$\text{ADE}_{H=6}$ (m)	$\text{ADE}_{H=12}$ (m)
$K_g = 2$	10553/11417	235.2 ± 0.4	435.1 ± 0.3
$K_g = 3$ (default)	11129/11993	234.9 ± 0.2	433.8 ± 1.5
$K_g = 6$	12857/13721	232.6 ± 0.5	430.8 ± 0.7
$K_g = 12$ (full T)	16313/17177	$\mathbf{231.7} \pm 0.3$	$\mathbf{429.1} \pm 0.8$
w/o RevIN	11111/11975	233.8 ± 0.2	433.1 ± 0.8
PM only (no MDM)	10742/11606	233.9 ± 0.1	433.2 ± 0.7
PM+MDM direct	11128/11992	233.9 ± 0.1	433.2 ± 0.8

APPENDIX B
CONFIGURATION MAP FOR REPORTED EVIDENCE

Table XIII consolidates which configuration and evaluation population produced each family of reported numbers.

APPENDIX C
EXTENDED TRAJECTORY-REPRESENTATION BASELINES

START [15], TrajMamba [16], and HGT-RN [28] rely on road-network, road/POI semantic, or pretraining inputs that are unavailable in our GPS-derived input setting. Their adapted GPS-derived encoder-plus-head variants would change the methods' core design, so we treat them as protocol notes.

APPENDIX D
GATE-HORIZON AND BACKBONE-COMPONENT ABLATIONS

Both blocks are referenced to the "+ KD" row of Table X (234.9 / 433.8); the ≈ 0.8 m offset from the no-KD main result does not affect the conclusions here.

Gate input horizon. The gate-horizon sweep in Table XIV varies $K_g \in \{2, 3, 6, 12\}$ on the deployed configuration. Extending the default $K_g = 3$ to the full window ($K_g = 12$) reduces ADE by 3.2 m at $H = 6$ and 4.7 m at $H = 12$, but increases parameter count by roughly $1.5\times$; $K_g = 2$ is slightly worse. We keep $K_g = 3$ at the trade-off knee, with the full-window setting retained as a higher-capacity option.