LATENT-TYPE POSTERIOR CONDITIONING FOR HETEROGENEOUS MULTI-AGENT TRAJECTORY PREDICTION UNDER NOISY AND MISSING TYPE LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterogeneous trajectory predictors condition each agent's representation on a semantic type label (pedestrian, cyclist, vehicle) and treat it as ground truth. In deployment the label is a perception output that may be wrong, missing, or inconsistent with the agent's motion, which raises the question of when a predictor should rely on it. We propose posterior type conditioning: a dynamics-based type classifier and a learned, context-dependent noisy-label channel are combined by Bayes' rule into a per-agent posterior over types, which conditions the predictor. Both are trained jointly from the prediction loss and the label likelihood, without supervision of label reliability. On a controlled benchmark and three real-world datasets spanning metric- and pixel-space trajectories and hand-annotated and perception-derived labels (COSMOS, Stanford Drone, Argoverse 2), we find that the value of the label is strongly class-dependent: relative to a type-agnostic model it improves prediction for a behaviorally distinct minority class by 8–15% (pedestrians among vehicles) and by 1–3% where motion already identifies the type. Naive conditioning degrades by 11–34% at 50% label corruption; label-noise augmentation removes most of this loss but forfeits part of the minority-class gain. Posterior conditioning retains most of that gain, degrades more gracefully, and is statistically indistinguishable from the best model on held-out scenes, cities, and recording dates. Its per-agent trust score detects corrupted labels with AUROC 0.91–0.97 on two of three datasets and recovers interpretable label–motion relationships, such as crediting a cyclist label only at cycling speeds.

1 INTRODUCTION

Multi-agent trajectory predictors for mixed traffic increasingly condition on a per-agent type label (Ma et al., 2019; Salzmann et al., 2020; Mo et al., 2022; Rainbow et al., 2021). A pedestrian, a cyclist, and a car exhibit distinct motion dynamics, so the agent type is a strong and readily available prior (Ma et al., 2019; Mo et al., 2022). On benchmarks it is also free: class labels are annotated and supplied at test time (Zhang et al., 2020). In deployment, however, the type is inferred by an upstream perception system and can be wrong, temporarily unavailable, or inconsistent across frames, and a newly detected agent may carry no reliable semantic evidence at all (Liang et al., 2020). Whether a predictor that has learned to rely on the type degrades gracefully when it becomes unreliable, and whether it can indicate when this is happening, remains insufficiently characterized.

This paper addresses that gap through three questions. First, how much does type information contribute to forecasting accuracy, and for which classes? Second, how does performance change when the type is corrupted or unavailable, and how much of that degradation can be mitigated? Third, can the predictor estimate, per agent, how much confidence to place in the label it was given? The answers are more specific than a binary view of type information as available or unavailable would suggest: the value of the label is concentrated in one kind of class, most of the robustness to label noise comes from the training procedure rather than the conditioning mechanism, and the per-agent trust estimate, while accurate about the label, carries no information about the accuracy of the prediction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To answer these questions we build a controlled testbed. A single conditioning module with one parameterization implements every variant compared—no type information (NONE), the observed label (HARD), a type inferred from motion alone (SOFT), and the proposed method (POSTERIOR)—so that differences are attributable to the conditioning rule and nothing else. We evaluate on three real-world datasets chosen to span regimes of type informativeness, from Argoverse 2 (AV2)(Wilson et al., 2021), where pedestrians are a rare and kinematically distinct minority, to COSMOS (Zang et al., 2024), where vehicle and pedestrian speeds overlap almost entirely, with the Stanford Drone Dataset (SDD) (Robicquet et al., 2016) in between; each is evaluated on a held-out split by scene, city, or recording date. A synthetic benchmark adds ground truth: types differ in future behavior while overlapping in observed speed, and a known fraction of agents move according to another type's dynamics while retaining a correct label (Appendix D). Labels are perturbed at test time by uniform flips, class-confusion flips, and dropout at rates up to 50%; every configuration is trained five times (three for secondary baselines) and compared by paired tests.

Our method treats the observed label as probabilistic evidence about a latent type. A probe estimates the type from motion; a learned observation channel, with a class-confusion matrix and a per-agent reliability, models how the provided label depends on the true type; Bayes' rule combines the two into a posterior over types, which conditions the predictor through feature-wise modulation and yields, as a by-product, a trust score—the posterior probability of the label that was given.

Findings: (1) Agent-type information is most valuable for a behaviorally distinct minority class and provides limited benefit elsewhere. On AV2, conditioning on the label improves pedestrian prediction by 8–15% relative to a type-agnostic model while producing no measurable gain for vehicles; on SDD and COSMOS the pooled improvement is 1–3%. (2) Naive type conditioning is sensitive to label corruption, degrading by 11–34% at a 50% corruption rate. Training with label corruption eliminates most of this degradation: across all three datasets and five independent training runs, the augmented HARD baseline is statistically indistinguishable from every more elaborate conditioning strategy in pooled prediction error. This negative result narrows the practical design space and is a central finding of the study. (3) Beyond augmentation, posterior conditioning provides additional benefit where type information is both informative and unreliable. On AV2 pedestrians it improves over augmented HARD by 2–7% ($p \leq 0.047$ in every condition, $p < 0.001$ under 50% label dropout), recovering 74% of the loss induced by missing labels; label smoothing of the type input does not reproduce this effect. (4) The trust score identifies inconsistencies between the observed label and motion evidence, with AUROC 0.91–0.97 on SDD and AV2 and 0.74–0.80 on four of five COSMOS runs (Section 5.4). The signal emerges even when training labels are clean and can be calibrated by post-hoc fitting. (5) On real labels, evaluated with an audit procedure whose verdicts are regenerated from the raw annotations, the trust score does not encounter any annotation error—none exists among the 130 audited windows. What it identifies are agents whose observed motion is atypical for the learned class distribution, such as moving vehicles in a class dominated by parked vehicles. After controlling for speed, trust is uncorrelated with prediction error. We therefore interpret it as a label–motion consistency signal rather than as a confidence measure for selective prediction.

2 Related work

Heterogeneous trajectory prediction: Early social models did not represent agent type (Alahi et al., 2016; Gupta et al., 2018). Later heterogeneous predictors incorporate it through type-specific encoders, heterogeneous interaction structures, or class embeddings (Ma et al., 2019; Salzmann et al., 2020; Rainbow et al., 2021; Mo et al., 2022), and recent driving predictors condition on the category attribute provided by the dataset (Zhou et al., 2022). All treat the supplied type as observed and do not model uncertainty in its correctness. We keep the backbone fixed and vary only the conditioning rule, so that the contribution of the type itself can be isolated.

Learning with noisy labels: Noisy *target* labels are well studied: loss correction under class-dependent noise (Natarajan et al., 2013; Patrini et al., 2017), sample selection (Han et al., 2018), and label smoothing (Szegedy et al., 2016; Müller et al., 2019); see Frénay & Verleysen (2014) for a survey. Our setting differs in that the uncertain label is an *input*: the model forecasts a continuous future conditioned on a categorical attribute that may be wrong or missing. The closest classical analogue is

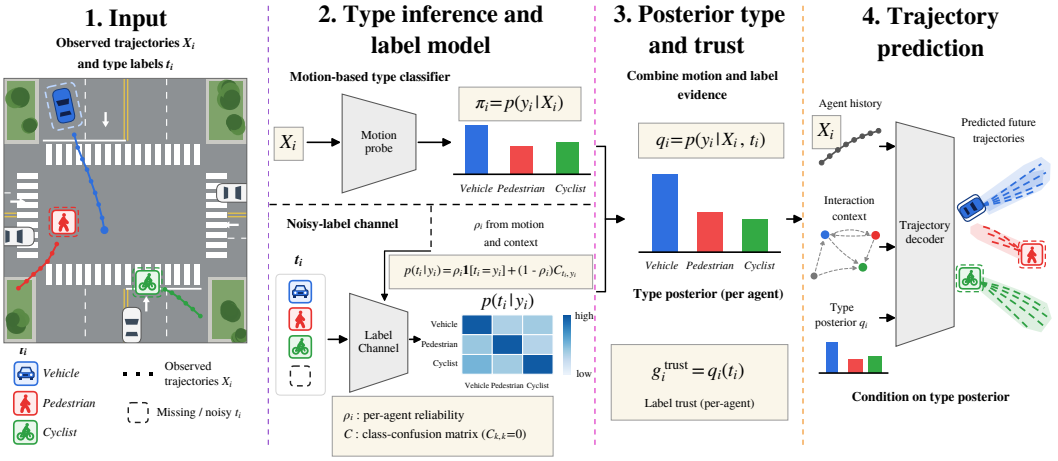


Figure 1: Posterior type conditioning: a motion probe (π_i) and a learned label channel (ρ_i, C) are combined by Bayes' rule into q_i, which conditions the predictor and yields the trust score.

annotator-noise modeling (Dawid & Skene, 1979), which we adapt into a learned, context-dependent, per-agent observation channel.

Uncertainty and selective prediction: Deep ensembles quantify confidence in a model's own predictions (Lakshminarayanan et al., 2017), and selective classification uses such confidence to accept or reject them (Geifman & El-Yaniv, 2017). Our trust score concerns a different quantity, whether a conditioning input should be trusted, and we show the two are distinct: after controlling for speed, label trust is uncorrelated with prediction error, whereas ensemble disagreement is strongly associated with it (Section 5.6).

3 Method

Setting: A scene contains N agents (Figure 1). Agent i has observed positions $\boldsymbol{X}_i \in \mathbb{R}^{T_o \times 2}$, a true type $y_i \in \{1, \dots, K\}$ that is never observed, and an observed type label $t_i \in \{1, \dots, K\} \cup \{\varnothing\}$, where $\varnothing$ denotes a missing label; t_i may differ from y_i. The model outputs K_m future trajectories $\hat{\boldsymbol{Y}}_i^{(m)} \in \mathbb{R}^{T_p \times 2}$, expressed as displacements from the last observed position, together with mode logits $\boldsymbol{\ell}_i \in \mathbb{R}^{K_m}$. With $\boldsymbol{Y}_i$ the ground-truth future, the prediction loss is winner-takes-all regression plus mode classification,

$$\mathcal{L}_{\text{pred}} \;=\; \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \Big[\min_m \text{ADE}\left(\hat{\boldsymbol{Y}}_i^{(m)}, \boldsymbol{Y}_i\right) + \text{CE}\left(\boldsymbol{\ell}_i,\; m_i^\star\right) \Big], \qquad m_i^\star = \arg\min_m \text{ADE}\left(\hat{\boldsymbol{Y}}_i^{(m)}, \boldsymbol{Y}_i\right), \tag{1}$$

where $\mathcal{V}$ is the set of valid agents and ADE is the mean Euclidean error over the T_p steps.

Backbone: A GRU encodes each agent's positions relative to its last observation, together with per-step velocities, into a type-agnostic embedding $\boldsymbol{z}_i \in \mathbb{R}^d$. Multi-head attention over the agents of a scene, with a learned encoding of each agent's position relative to the scene centroid, gives a context embedding $\boldsymbol{c}_i \in \mathbb{R}^d$. An MLP decodes $[\boldsymbol{z}_i'; \boldsymbol{c}_i]$ into K_m step sequences whose cumulative sums are the trajectories, and a second MLP produces the mode logits. Type information enters at exactly one point, the conditioning step $\boldsymbol{z}_i \mapsto \boldsymbol{z}_i'$, implemented by feature-wise linear modulation (Perez et al., 2018):

$$\boldsymbol{z}_i' \;=\; \boldsymbol{z}_i + g_i \left(\boldsymbol{\gamma}(\boldsymbol{e}_i) \odot \boldsymbol{z}_i + \boldsymbol{\beta}(\boldsymbol{e}_i)\right), \qquad \boldsymbol{e}_i \in \mathbb{R}^{d_e}, \tag{2}$$

where $\boldsymbol{e}_i$ is a type embedding, $\boldsymbol{\gamma}, \boldsymbol{\beta} : \mathbb{R}^{d_e} \to \mathbb{R}^d$ are linear maps initialized to zero so that training starts from $\boldsymbol{z}_i' = \boldsymbol{z}_i$, and $g_i \in [0, 1]$ is a scalar gate. Every variant shares this architecture, the type probe, and the head ensemble described below, so parameter counts and auxiliary losses are identical up to the conditioning rule.

Conditioning variants: Let $\boldsymbol{E} \in \mathbb{R}^{(K+1)\times d_e}$ be a learned embedding table with one row per type and one row for $\varnothing$. The variants differ only in $(\boldsymbol{e}_i, g_i)$:

$$\begin{aligned}
\text{NONE}: &\quad g_i = 0 \quad \text{(type-agnostic)};\\
\text{HARD}: &\quad \boldsymbol{e}_i = \boldsymbol{E}[t_i],\ g_i = 1 \quad \text{(the row for } \varnothing \text{ if the label is missing)};\\
\text{SOFT}: &\quad \boldsymbol{e}_i = \textstyle\sum_{k=1}^{K} \pi_i(k)\, \boldsymbol{E}[k],\ g_i = 1 \quad \text{(label ignored)};\\
\text{POSTERIOR}: &\quad \boldsymbol{e}_i = \textstyle\sum_{k=1}^{K} q_i(k)\, \boldsymbol{E}[k],\ g_i = 1,
\end{aligned}$$

with π_i and q_i defined next.

Type probe: An MLP on the encoder output, with gradient stopped at $\boldsymbol{z}_i$ so that the probe reads the encoder but does not shape it, gives $\pi_i(k) = p(y_i = k \mid \boldsymbol{z}_i)$, a distribution over types inferred from motion alone. It is trained in every variant.

Noisy-label channel: The observed label is modeled as a noisy observation of the true type,

$$p(t_i = t \mid y_i = k) \;=\; \rho_i \mathbb{1}[t = k] \;+\; (1 - \rho_i)\, C_{k,t}, \qquad p(t_i = \varnothing \mid y_i = k) = 1 \;\; \forall k, \tag{3}$$

where $C \in \mathbb{R}^{K\times K}$ is a learned confusion matrix with $C_{k,k} = 0$ and $\sum_{t\neq k} C_{k,t} = 1$ (a softmax over the off-diagonal entries of each row), and

$$\rho_i \;=\; \tfrac{1}{2} + \tfrac{1}{2}\,\sigma\big(f_\rho([\boldsymbol{z}_i; \boldsymbol{c}_i; u_i])\big) \;\in\; [\tfrac{1}{2}, 1) \tag{4}$$

is a per-agent reliability produced by an MLP f_ρ from the dynamics embedding, the scene context, and the ensemble disagreement u_i (below). The floor $\rho_i \geq \frac{1}{2}$ removes a mirror solution in which the channel inverts the labels and the type branches permute; without it we observed this collapse on two-class data. A missing label carries no evidence (likelihood 1 under every type). The bias of f_ρ is initialized so that $\rho_i \approx 0.9$.

Posterior and trust: Bayes' rule combines the two sources of evidence,

$$q_i(k) \;=\; \frac{\pi_i(k)\, p(t_i \mid y_i = k)}{\sum_{k'} \pi_i(k')\, p(t_i \mid y_i = k')}, \tag{5}$$

and POSTERIOR conditions on the expected embedding $\sum_k q_i(k)\boldsymbol{E}[k]$. The *trust score* is

$$g_i^{\text{trust}} \;=\; q_i(t_i) \quad \text{if } t_i \neq \varnothing, \qquad g_i^{\text{trust}} = 0 \text{ otherwise}, \tag{6}$$

the posterior probability that the label provided is the true type. When motion and label agree, q_i concentrates on t_i and POSTERIOR reduces to HARD; when they disagree, mass moves to the types the motion supports. (For POSTERIOR the FiLM gate in Eq. 2 is $g_i = 1$; the trust score is a separate quantity and is not applied as a gate.)

Head ensemble: $M = 5$ light heads on the stopped-gradient embedding each predict the normalized endpoint of the future (Lakshminarayanan et al., 2017). Each head is trained by squared error under an independent Bernoulli(0.7) bootstrap mask, and the mean of these losses is $\mathcal{L}_{\text{ens}}$. Their disagreement $u_i = \log\big(1 + \sum_{j\in\{x,y\}} \mathrm{Var}_m\,[\hat{y}_{i,j}^{(m)}]\big)$ is used as an input to f_ρ. The ensemble is present in every variant.

Training: The total loss is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{pred}} + \lambda_{\text{ch}}\,\mathcal{L}_{\text{ch}} + \lambda_{\text{ens}}\,\mathcal{L}_{\text{ens}}, \qquad \mathcal{L}_{\text{ch}} = -\frac{1}{|\mathcal{V}_{\text{obs}}|} \sum_{i\in\mathcal{V}_{\text{obs}}} \log \sum_k \pi_i(k)\, p(t_i \mid y_i = k), \tag{7}$$

where $\mathcal{V}_{\text{obs}}$ is the set of valid agents with an observed label and $\lambda_{\text{ch}} = \lambda_{\text{ens}} = 0.5$. $\mathcal{L}_{\text{ch}}$ is the marginal likelihood of the observed label under the channel; training the probe through it, rather than through a cross-entropy to t_i, lets π_i estimate the clean type rather than the noisy-label distribution. Gradient flow is asymmetric by design: the probe receives gradient only from $\mathcal{L}_{\text{ch}}$ (its input is detached and π_i is detached inside Eq. 5), whereas f_ρ and C receive gradient from both $\mathcal{L}_{\text{ch}}$ and, through q_i, from $\mathcal{L}_{\text{pred}}$, so the channel learns which labels help prediction. In the variants without a channel, $\mathcal{L}_{\text{ch}}$ is replaced by the cross-entropy of π_i against t_i, so every variant trains the probe. Following a rule imposed on all variants, training labels are corrupted on the fly: in each batch, 10% of labels are flipped uniformly to another type and 10% are replaced by $\varnothing$ (*label-corruption augmentation*). Model selection uses validation ADE with clean labels. We also report every variant without augmentation.

Table 1: Datasets and evaluation splits. Speeds are per model step on the training split; "stationary" is below 2 px (SDD, COSMOS) or 0.1 m (AV2) per step.

dataset	units	classes	train / shift scenes	shift split	class shares (%)	stationary vehicles
COSMOS	px, 2.5 Hz	2	7,617 / 5,748	different day (+4 wk)	veh 37, ped 63	81%
SDD	px, 2.5 Hz	3	2,406 / 699	deathCircle, coupa	ped 77, cyc 13, veh 10	91%
AV2	m	3	11,950 / 8,472	Miami	veh 87, ped 11, cyc 2	56%

Calibration: g_i^{trust} is a posterior probability, but its calibration depends on the learned channel. We fit an isotonic map (Zadrozny & Elkan, 2002) on a random half of the evaluated agents, with labels corrupted at the training rate, and report expected calibration error (Guo et al., 2017) on the other half, before and after.

4 Experimental setup

Datasets: Table 1 summarizes the three datasets (Robicquet et al., 2016; Wilson et al., 2021; Zang et al., 2024). Each provides an in-distribution test split and a *shift* split held out by scene (SDD: deathCircle and coupa), by city (AV2: Miami; AV2 scenes are restricted to those containing at least one pedestrian or cyclist), or by recording date (COSMOS: train and validation from 12 December 2023, test from 9 January 2024, same intersection and camera); all results in the main text are on the shift split. All use 8 observed and 12 predicted steps; types are collapsed to pedestrian, cyclist, and vehicle (COSMOS: vehicle and pedestrian). How well motion identifies the class differs across datasets and is central to the results. On AV2, in metric units, pedestrians (11.1% of agents) occupy a narrow low-speed band (median 0.36, 95th percentile 0.66 m per step), whereas vehicles are bimodal: 56% stationary, the remainder far faster than any pedestrian (95th percentile 4.1 m per step). On SDD, 90.6% of vehicle windows are stationary (parked cars) with a minority of moving golf carts, while pedestrians (median 10.3 px per step) and cyclists (30.0) are well separated. On COSMOS, at 2.5 Hz in pixels, vehicle and pedestrian speeds overlap almost completely (median 0.56 vs. 6.62; 95th percentile 13.8 vs. 13.0 px per step). COSMOS labels are perception outputs (a fine-tuned YOLOv8 detector with BoT-SORT tracking); each track's type is the majority of its per-frame labels, and all 19,980 test tracks were unanimous, so they are treated as clean.

Corruption protocol: Labels of the shift split are corrupted at test time at rates $r \in \{0.1, 0.2, 0.3, 0.5\}$: *uniform* flips relabel a fraction r of agents with a class drawn uniformly from the others; *confusion* flips draw the wrong class from a fixed plausible matrix (pedestrian→cyclist 0.8, cyclist→pedestrian 0.7, vehicle→cyclist 0.9, remainder to the third class), on the three-class datasets only; *missing* replaces the label by $\varnothing$. Corruption is seeded, so every model is evaluated on the same perturbed labels. Trust AUROC separates flipped from unflipped agents by $1 - g_i^{\text{trust}}$ over agents whose label was not dropped.

Baselines and ablations: NONE, HARD, SOFT as above; HARD without augmentation; HARD with label smoothing of the type input at $\varepsilon \in \{0.1, 0.3\}$ (the embedding of $(1-\varepsilon)\,\text{onehot}(t_i) + \varepsilon/K$), testing whether hedging the input alone explains any gain; POSTERIOR without augmentation; and range-augmented HARD$_R$ / POSTERIOR$_R$ with a per-batch corruption rate drawn from $U[0, 0.5]$. A mixture-of-experts decoder over candidate types was dropped as a negative result (Appendix E).

Metrics and statistics: minADE_6 over 6 modes, pooled and per class. Every configuration is trained with five independent runs (three for secondary baselines); tables report mean±std, and POSTERIOR vs. HARD comparisons use paired t-tests over matched runs. Training is not bit-deterministic on our hardware; the number of runs is stated with every result.

Implementation: $d = 128$, 6 modes, AdamW (Loshchilov & Hutter, 2019) with a one-cycle schedule; full hyperparameters in Appendix F. The channel adds 16.6k parameters (3.7% of the model) and 2.8% forward-pass time over HARD (36.0 vs. 37.1 ms per batch of 128 scenes, NVIDIA T1000). Code, evaluation logs, and audit files are provided as supplementary material.

Table 2: Pooled minADE_6 on the held-out (shift) split, mean±std over independent training runs (count in parentheses). Best per column in bold. SOFT and NONE do not read the label and are constant across conditions.

model (runs)	clean	unif. 30%	unif. 50%	miss. 50%
AV2 (m)				
NONE (3)	0.547±0.001	0.547±0.001	0.547±0.001	0.547±0.001
SOFT (3)	**0.541±0.003**	**0.541±0.003**	**0.541±0.003**	**0.541±0.003**
HARD (no aug.) (3)	0.549±0.003	0.608±0.008	0.647±0.011	0.614±0.039
HARD (5)	0.547±0.005	0.552±0.006	0.556±0.006	0.550±0.005
HARD+$\mathrm{LS}_{0.3}$ (3)	0.543±0.003	0.548±0.003	0.552±0.003	0.547±0.003
POSTERIOR (5)	0.545±0.002	0.549±0.003	0.552±0.003	0.548±0.003
SDD (px)				
NONE (3)	20.36±0.25	20.36±0.25	20.36±0.25	20.36±0.25
SOFT (3)	20.16±0.20	20.16±0.20	**20.16±0.20**	20.16±0.20
HARD (no aug.) (3)	20.08±0.14	20.63±0.19	21.02±0.16	22.83±2.00
HARD (5)	20.15±0.12	20.26±0.15	20.34±0.16	20.26±0.17
HARD+$\mathrm{LS}_{0.3}$ (3)	20.03±0.16	20.13±0.14	20.18±0.12	20.14±0.17
POSTERIOR (5)	**20.02±0.10**	**20.10±0.11**	20.17±0.10	**20.09±0.10**
COSMOS (px)				
NONE (3)	4.63±0.06	4.63±0.06	4.63±0.06	4.63±0.06
SOFT (3)	4.62±0.02	4.62±0.02	**4.62±0.02**	4.62±0.02
HARD (no aug.) (3)	**4.46±0.03**	5.36±0.04	5.99±0.06	5.42±0.28
HARD (5)	4.51±0.01	4.63±0.02	4.71±0.02	4.57±0.01
HARD+$\mathrm{LS}_{0.3}$ (3)	4.52±0.04	**4.62±0.03**	4.69±0.03	4.58±0.03
POSTERIOR (5)	4.50±0.05	4.63±0.05	4.72±0.06	**4.56±0.05**

5 RESULTS

5.1 POOLED ACCURACY: AUGMENTATION IS WHAT MATTERS

Table 2 reports pooled minADE_6 on the shift split. Without augmentation, HARD is fragile: at 50% uniform corruption it loses 18% on AV2, 5% on SDD, and 34% on COSMOS, and at 50% missing labels 12%, 14%, and 22%. Training with 10% label corruption removes most of this: augmented HARD loses 1.7%, 1.0%, and 4.3% under the same conditions. Once augmented, the conditioning rule has no measurable effect on pooled error: POSTERIOR and HARD are within run-to-run variation on all three datasets in every condition ($|\Delta| \leq 0.8\%$, $p \geq 0.19$; Appendix Table 5), and label smoothing of the type input likewise matches HARD. The pooled value of the label itself is small—relative to NONE, HARD gains 0.1% on AV2, 1.0% on SDD, and 2.6% on COSMOS—and SOFT, which never reads the label, is the best pooled model on AV2 in every condition. A label worth 1–3% pooled is a label whose corruption can cost 1–3%, and that is the scale of every pooled effect in Table 2. This is the main result, and it is negative for the method family this work belongs to: for pooled robustness to label noise, label-corruption augmentation is necessary and, at these noise levels, sufficient.

5.2 MINORITY-CLASS ACCURACY: WHERE THE LABEL AND THE POSTERIOR MATTER

Table 3 breaks out AV2 pedestrians, a class that is rare (11.1% of agents) and kinematically distinct. Relative to NONE, the label is worth 8% with augmented HARD, 15% with non-augmented HARD, and 10% with POSTERIOR; SOFT recovers only 3% from motion alone. The two HARD numbers expose the cost of hardening: augmentation costs pedestrians 8% on clean labels (0.332→0.360), because the model learns to lean less on a label that is sometimes wrong. POSTERIOR pays about a third of that tax (0.352 with augmentation vs. 0.342 without) and keeps its advantage once labels are corrupted. The label helps here because pedestrian motion—slow, heading-change-heavy—is a strong prior the model cannot fully recover from eight observed steps, and pedestrians are too rare for the decoder to have learned it unaided.

Under corruption the methods separate. Augmented HARD loses 5.5% of pedestrian accuracy at 30% flips and 7.6% at 50% missing labels; POSTERIOR loses 4.9% and 2.0%, recovering 14% of the flip-induced loss but 74% of the missing-label loss (Appendix Table 7). The difference is robust:

Table 3: AV2 pedestrians, shift split, minADE_6 (m). The label is worth 8–15% over NONE; POSTERIOR adds 2–7% over augmented HARD, most under missing labels; label smoothing of the input does not reproduce this.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
NONE (3)	0.389±0.003	0.389±0.003	0.389±0.003	0.389±0.003	0.389±0.003
SOFT (3)	0.376±0.005	0.376±0.005	**0.376±0.005**	0.376±0.005	0.376±0.005
HARD (no aug.) (3)	**0.332±0.001**	0.377±0.003	0.405±0.006	0.357±0.002	0.454±0.051
HARD (5)	0.360±0.006	0.380±0.005	0.392±0.005	0.375±0.004	0.387±0.005
HARD+$\mathrm{LS}_{0.3}$ (3)	0.360±0.002	0.376±0.002	0.388±0.003	0.372±0.002	0.383±0.003
HARD_R (3)	0.379±0.003	0.384±0.003	0.387±0.002	0.383±0.003	0.385±0.003
POSTERIOR (no aug.) (3)	0.341±0.003	0.375±0.001	0.395±0.003	**0.357±0.000**	**0.352±0.001**
POSTERIOR (5)	0.352±0.004	**0.369±0.003**	0.380±0.004	0.361±0.004	0.359±0.003
$\mathrm{POSTERIOR}_R$ (3)	0.370±0.006	0.378±0.004	0.384±0.003	0.376±0.004	0.376±0.005

POSTERIOR beats HARD in every condition, in 4/5 or 5/5 runs, with $p = 0.047$ on clean labels, $p \leq 0.006$ under flips, and $p < 0.001$ under missing labels (Appendix Table 5). Label smoothing at $\varepsilon = 0.3$ matches HARD within 0.004 m in every condition and does nothing for missing labels, since smoothing modifies known labels and leaves the $\varnothing$ embedding unchanged; the gain is attributable to the posterior combination, not to hedging the input. The range-augmented $\mathrm{POSTERIOR}_R$ is the best pooled model on AV2 (0.542) but gives back the pedestrian gain (0.370 vs. 0.352): heavier augmentation buys pooled robustness by hedging the minority class. The same pattern appears, weaker, on SDD pedestrians (-1.2 to -1.4%, 4/5 runs, p between 0.06 and 0.11; Appendix Table 11) and not on COSMOS, where pedestrians are the majority and the classes are not separable by motion ($p \geq 0.10$, HARD marginally ahead). We read this as one finding: the label, and the posterior's treatment of it, matter in proportion to how much motion information the class label carries.

5.3 WHERE ROBUSTNESS COMES FROM

Removing augmentation from both models (Appendix Table 6) shows that both are fragile, but not equally. At 50% flips, non-augmented HARD loses 17.8% on AV2, 4.7% on SDD, and 34.4% on COSMOS; non-augmented POSTERIOR loses 6.8%, 3.3%, and 30.2%. Where the probe can identify the minority class from motion (AV2) the Bayesian combination alone recovers most of the robustness; where it cannot (SDD, COSMOS) it recovers little, and its learned reliability $\bar{\rho}$ sits at 0.92–0.96, i.e. the channel trusts essentially every label. Robustness that holds across datasets therefore comes from having seen corruption in training. What the combination confers regardless is the trust signal: non-augmented POSTERIOR still separates flipped from clean labels with AUROC 0.89–0.95 on SDD and AV2 (Table 4), because the probe learns to disagree with labels that contradict motion whether or not noise was injected; augmentation lowers $\bar{\rho}$ to 0.82–0.86 and lets that disagreement change the prediction. An oracle bound (Appendix Table 7) makes the pooled null quantitative: once HARD is augmented, its corruption-induced pooled loss is 0.5–1.7% on AV2 and SDD and at most 4.3% on COSMOS, leaving little to recover; the one cell with a material loss is AV2 pedestrians under missing labels, where POSTERIOR recovers 74%.

5.4 THE TRUST SCORE

Detection of injected corruption: Table 4 reports AUROC for separating flipped from unflipped agents. On AV2 it is 0.96–0.97 and on SDD 0.90–0.91, stable across corruption types and rates; mean trust is 0.89–0.94 on clean-label agents and 0.13–0.36 on flipped agents. On COSMOS, four of five runs give 0.74–0.80 and one converged to a label-permuted solution (AUROC 0.41). The $\rho \geq \frac{1}{2}$ floor prevents the channel from inverting labels but not the probe from learning the permuted assignment when the two classes are barely separable by motion; we report the run rather than exclude it. The three datasets thus order the trust signal by class separability (Section 4), as a signal derived from label–motion consistency should.

What drives it: The channel recovers label–motion relationships that read as rules: trust in a cyclist label rises with speed and is near zero for a stopped cyclist (the two in the random audit sample of Section 5.5 received $g_i = 0.11$ and 0.12); trust in a vehicle label on SDD falls once the vehicle

Table 4: Trust signal on the shift split: AUROC for detecting flipped labels, mean trust on clean-label / flipped agents, and mean learned reliability $\bar{\rho}$. The signal is present without augmentation; augmentation lowers $\bar{\rho}$ so that it affects the prediction. COSMOS AUROC is 0.74–0.80 on 4/5 runs and 0.41 on one label-permuted run.

dataset	model	AUROC (unif. 30%)	AUROC (conf. 30%)	$\bar{g}$ clean / flipped	$\bar{\rho}$
AV2	POSTERIOR (5)	0.96±0.00	0.97±0.00	0.94 / 0.17	0.86
AV2	POSTERIOR (no aug.) (3)	0.95±0.00	0.96±0.01	0.95 / 0.29	0.95
SDD	POSTERIOR (5)	0.91±0.00	0.91±0.00	0.90 / 0.33	0.84
SDD	POSTERIOR (no aug.) (3)	0.89±0.00	0.90±0.01	0.95 / 0.52	0.96
COSMOS	POSTERIOR (5)	0.70±0.16	–	0.77 / 0.59	0.82
COSMOS	POSTERIOR (no aug.) (3)	0.64±0.02	–	0.92 / 0.89	0.92

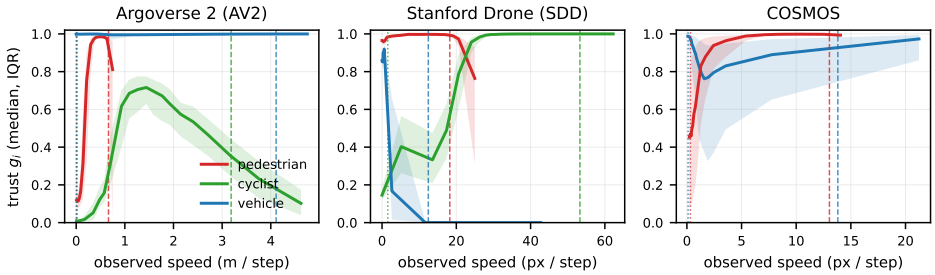


Figure 2: Trust against observed speed by class (median and interquartile range in quantile bins of speed; dashed lines: training 5th/95th percentile speed of each class). Low trust concentrates where an agent's speed is atypical for its labeled class.

moves faster than the parked majority. For every class whose trust varies (standard deviation ≥ 0.28), a gradient-boosted regression on seven kinematic and context features explains $R^2 = 0.90$–0.98 of g_i (one run, clean labels); the other classes have trust ≥ 0.97 with almost no variance. Mean speed dominates on SDD and COSMOS; heading change dominates for AV2 pedestrians (permutation importance 0.88 vs. 0.58). Neighbour count and distances contribute nothing: trust is a function of the agent's own motion (Figure 2).

Calibration: Raw g_i is over-confident (ECE 0.16, 0.07, 0.04 on COSMOS, SDD, AV2; one run, 10% corruption). An isotonic map fitted on a random half of the evaluated agents reduces ECE on the other half to 0.008, 0.009, and 0.002. This requires held-out labels corrupted at a known rate; transfer to other rates was not tested.

5.5 WHAT LOW TRUST MEANS ON REAL LABELS

The corruption above is synthetic. To see what the score does on labels as annotated, we audited 130 windows on SDD's held-out scenes: the 50 lowest-trust agents in the split, all vehicles with $g_i = 0$ (41 distinct tracks); 30 vehicles with $g_i \geq 0.8$ from the same five videos as a same-class control (14 tracks); and 50 random agents (47 tracks). Verdicts follow a fixed rule on the raw annotations, in order: *wrong* if the annotated class does not map to the study class; *unclear* if $\geq 40\%$ of the track is lost or the box area is outside $[0.5, 2]\times$ the class median for that video; *atypical* if the observed speed lies outside the training 5th–95th percentile of the study class; otherwise *ok*. A human rater who judged the first 100 windows before the rule was fixed agreed on 74%, with 25 of 26 disagreements in the direction of the rater over-calling atypicality or uncertainty (Appendix B).

No window in any group was mislabelled. Low-trust vehicles are *atypical* in 38 of the 42 judgeable windows: all are moving cars and golf carts in a class whose training distribution is 90% parked. High-trust vehicles are atypical in 0 of 23; random agents in 2 of 38 (two stopped cyclists, both also low-trust). Low trust on SDD therefore marks motion that contradicts the learned class distribution, not annotation error. The stopped cyclists and the thirteen standing pedestrians in the random sample

make the mechanism precise: a stopped cyclist looks like a pedestrian or a parked vehicle and its trust drops to 0.1; a standing pedestrian is common enough in training that standing is not evidence against the label, and its trust stays at 0.95. Trust falls when the motion is positive evidence for a *different* type, not when it is merely unusual. On the synthetic benchmark, where such agents exist by construction, the score identifies them with AUROC 0.96 in-distribution and 0.91 under shift (Appendix D).

5.6 Trust is not confidence

A natural use of a per-agent score is selective prediction. It does not work, and the reason is instructive. For pedestrians and cyclists, abstaining on the 30% least-trusted agents *raises* the error of the rest on all three datasets (shift split, 5-run means: AV2 pedestrians 0.35→0.38, AV2 cyclists 0.92→1.13, SDD cyclists 39.1→43.8, COSMOS pedestrians 5.29→6.18), and the rank correlation of g_i with error is positive ($+0.29$ to $+0.51$); vehicles are the exception, for the reason in Section 5.5. Ensemble disagreement, already inside the model, behaves as a confidence score should ($\rho = -0.45$ to -0.78). Controlling for speed, trust is uncorrelated with error in 13 of 14 class×split cells (partial Spearman $|\rho| \leq 0.18$); the exception, SDD-shift vehicles ($\rho = -0.68$), is again the parked/moving split. Trust is high for fast, typical-for-their-class agents, which have large absolute and small relative error; it ranks agents by label consistency, and error ranks them by how far they move. We therefore present g_i as a label-audit signal, not a confidence score (Appendix C).

6 Discussion and limitations

What the study establishes: The value of a type label to a trajectory predictor is the amount of motion information the class carries that the model cannot already read from the observed trajectory. On the datasets and horizon we study, that is large for one class on one dataset and small elsewhere. Any conditioning rule trained under label corruption is robust on pooled error; the posterior earns its cost on the minority class, most of all under missing labels, where it recovers most of what naive conditioning loses, and it is the only label-reading rule that retains some robustness without augmentation where the type is identifiable from motion. Its trust score is an interpretable by-product that says, per agent, whether the motion supports the label; it is not a statement about the accuracy of the prediction.

Label errors are synthetic: The trust signal was validated on injected flips and on synthetic behavioral anomalies, because none of the three datasets contains natural label errors we could find: the audit found 0/130 on SDD, and COSMOS's detector-derived labels are unanimous within every track. COSMOS is the deployment setting the work is motivated by—labels from a perception pipeline—but without human ground truth we cannot measure its label error rate, and a unanimous track can still be uniformly wrong. The score's behavior on labels with known real errors is therefore unmeasured. This is the main limitation.

Scope: The backbone is a compact reference model, chosen so that conditioning is the only variable; whether the class-dependence of label value transfers to large predictors is future work. The pedestrian result is one class on one dataset, supported by a weaker replication on SDD. The confusion-flip pattern is designed by us; uniform flips are the primary condition. Recalibration requires held-out labels corrupted at a known rate, and we did not test transfer across rates. Training is nondeterministic on our hardware, and the COSMOS trust signal collapsed to a label-permuted solution on one of five runs.

7 Conclusion

We asked when a heterogeneous trajectory predictor should trust its type labels. The answer we can support is: train under label corruption regardless of conditioning rule; expect the label to help only where the class carries motion information the observed trajectory does not reveal; if labels can be wrong or missing and a minority class matters, treat the label as evidence and combine it with motion; and read the resulting trust score as a statement about the label, not about the prediction.

Reproducibility statement

Every table in this paper is generated by a script from the per-run evaluation logs, and every audit verdict is regenerated from the raw annotations by the stated rule with a fixed seed (Appendix F). Supplementary material contains the code (training, evaluation, corruption, the three dataset converters, table generation, and the audit and analysis scripts), the evaluation logs of every run, the audit files, and the derived result files. The datasets are public: the Stanford Drone Dataset (SDD), Argoverse 2 (AV2) motion forecasting, and the COSMOS trajectory dataset (CC-BY-4.0); download links and the exact converter commands, including the split definitions, are given in the supplementary README. Hyperparameters and software versions are listed in Appendix F.

AI use statement

Generative AI tools were used solely for proofreading and language refinement. They were not used to generate scientific content, develop the methodology, conduct the analysis, interpret the results, or formulate the conclusions. All scientific content and final revisions were reviewed and approved by the authors.

References

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.

Alexander Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C*, 28(1):20–28, 1979.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11550–11559. IEEE, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. TrafficPredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019.

Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9554–9567, 2022.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *NeurIPS*, 2019.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Ben A. Rainbow, Qianhui Men, and Hubert P. H. Shum. Semantics-STGCNN: A semantics-guided spatial-temporal graph convolutional network for multi-class trajectory prediction. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021.

Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, 2016.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.

Chengbo Zang, Mehmet Kerem Turkcan, Gil Zussman, Javad Ghaderi, and Zoran Kostic. Data-driven traffic simulation for an intersection in a metropolis. Presented at the CVPR Workshop POETS, June 2024. URL `https://arxiv.org/abs/2408.00943`.

Sibo Zhang, Yuexin Ma, and Ruigang Yang. CVPR 2019 WAD challenge on trajectory prediction and 3D perception. *arXiv preprint arXiv:2004.05966*, 2020.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. HiVT: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022.

A FULL RESULTS

Tables referenced from Section 5, followed by all variants on both splits and per-class results.

Table 5: Paired t-tests, POSTERIOR minus HARD (both augmented), matched over 5 runs; negative Δ favours POSTERIOR. *: $p < 0.05$. Confusion flips are not evaluated on the two-class COSMOS.

	AV2 pedestrians			AV2 pooled			SDD pooled			COSMOS pooled		
condition	Δ	wins	p	Δ	wins	p	Δ	wins	p	Δ	wins	p
clean	-2.2%	4/5	0.047*	-0.3%	3/5	0.562	-0.6%	3/5	0.272	-0.1%	3/5	0.819
unif. 30%	-2.8%	5/5	0.005*	-0.5%	3/5	0.457	-0.8%	3/5	0.232	+0.1%	3/5	0.780
unif. 50%	-3.0%	5/5	0.006*	-0.7%	3/5	0.362	-0.8%	4/5	0.217	+0.3%	2/5	0.543
conf. 30%	-3.5%	5/5	0.002*	-0.3%	3/5	0.583	-0.8%	3/5	0.244	–	–	–
miss. 50%	-7.2%	5/5	0.000*	-0.4%	3/5	0.512	-0.8%	3/5	0.199	-0.2%	2/5	0.715

Table 6: Augmentation ablation, pooled minADE_6, shift split. Without label-corruption augmentation both models are fragile; the posterior combination alone confers robustness only where the type is identifiable from motion (AV2).

model (runs)	clean	unif. 50%	miss. 50%
AV2 (m)			
HARD (no aug.) (3)	0.549±0.003	0.647±0.011	0.614±0.039
HARD (5)	0.547±0.005	0.556±0.006	0.550±0.005
POSTERIOR (no aug.) (3)	0.548±0.002	0.586±0.004	0.554±0.003
POSTERIOR (5)	**0.545±0.002**	**0.552±0.003**	**0.548±0.003**
SDD (px)			
HARD (no aug.) (3)	20.08±0.14	21.02±0.16	22.83±2.00
HARD (5)	20.15±0.12	20.34±0.16	20.26±0.17
POSTERIOR (no aug.) (3)	20.17±0.26	20.84±0.35	20.28±0.25
POSTERIOR (5)	**20.02±0.10**	**20.17±0.10**	**20.09±0.10**
COSMOS (px)			
HARD (no aug.) (3)	**4.46±0.03**	5.99±0.06	5.42±0.28
HARD (5)	4.51±0.01	**4.71±0.02**	4.57±0.01
POSTERIOR (no aug.) (3)	4.46±0.04	5.81±0.15	4.63±0.05
POSTERIOR (5)	4.50±0.05	4.72±0.06	**4.56±0.05**

Table 7: Oracle recovery: each model's clean-label performance is its own upper bound under corruption. *loss* is the relative increase in minADE_6 from clean to corrupted labels; *recovered* $= 1 - \mathrm{loss}_{\mathrm{POSTERIOR}}/\mathrm{loss}_{\mathrm{HARD}}$. Pooled losses of augmented HARD are 0.5–4.3%, leaving little to recover; on AV2 pedestrians under missing labels the loss is 7.6% and POSTERIOR recovers 74% of it.

dataset / metric	condition	HARD loss	POSTERIOR loss	recovered
AV2 pooled	unif. 30%	1.0%	0.8%	16%
	unif. 50%	1.7%	1.4%	18%
	conf. 30%	0.5%	0.5%	1%
	miss. 50%	0.6%	0.5%	13%
AV2 pedestrians	unif. 30%	5.5%	4.9%	14%
	unif. 50%	9.0%	8.0%	13%
	conf. 30%	4.0%	2.6%	38%
	miss. 50%	7.6%	2.0%	74%
SDD pooled	unif. 30%	0.6%	0.4%	30%
	unif. 50%	1.0%	0.8%	21%
	conf. 30%	0.6%	0.5%	23%
	miss. 50%	0.6%	0.4%	41%
COSMOS pooled	unif. 30%	2.6%	2.9%	-9%
	unif. 50%	4.3%	4.8%	-10%
	miss. 50%	1.3%	1.3%	4%

Table 8: All variants, shift split, pooled minADE_6, mean±std over independent runs (count in parentheses).

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
AV2 (m)					
NONE (3)	0.547±0.001	0.547±0.001	0.547±0.001	0.547±0.001	0.547±0.001
SOFT (3)	**0.541±0.003**	**0.541±0.003**	**0.541±0.003**	**0.541±0.003**	**0.541±0.003**
HARD (no aug.) (3)	0.549±0.003	0.608±0.008	0.647±0.011	0.591±0.005	0.614±0.039
HARD (5)	0.547±0.005	0.552±0.006	0.556±0.006	0.549±0.005	0.550±0.005
HARD+$\mathrm{LS}_{0.1}$ (3)	0.545±0.003	0.551±0.004	0.554±0.004	0.548±0.004	0.548±0.004
HARD+$\mathrm{LS}_{0.3}$ (3)	0.543±0.003	0.548±0.003	0.552±0.003	0.546±0.003	0.547±0.003
HARD_R (3)	0.547±0.002	0.548±0.002	0.549±0.002	0.548±0.002	0.548±0.002
POSTERIOR (no aug.) (3)	0.548±0.002	0.570±0.003	0.586±0.004	0.575±0.003	0.554±0.003
POSTERIOR (5)	0.545±0.002	0.549±0.003	0.552±0.003	0.548±0.003	0.548±0.003
$\mathrm{POSTERIOR}_R$ (3)	0.542±0.004	0.544±0.005	0.545±0.005	0.543±0.005	0.543±0.005
SDD (px)					
NONE (3)	20.36±0.25	20.36±0.25	20.36±0.25	20.36±0.25	20.36±0.25
SOFT (3)	20.16±0.20	20.16±0.20	20.16±0.20	20.16±0.20	20.16±0.20
HARD (no aug.) (3)	20.08±0.14	20.63±0.19	21.02±0.16	20.68±0.17	22.83±2.00
HARD (5)	20.15±0.12	20.26±0.15	20.34±0.16	20.27±0.16	20.26±0.17
HARD+$\mathrm{LS}_{0.1}$ (3)	20.17±0.20	20.27±0.19	20.34±0.19	20.28±0.21	20.27±0.17
HARD+$\mathrm{LS}_{0.3}$ (3)	20.03±0.16	20.13±0.14	20.18±0.12	20.14±0.14	20.14±0.17
HARD_R (3)	**19.96±0.24**	**20.00±0.24**	**20.04±0.22**	**20.01±0.23**	**20.02±0.24**
POSTERIOR (no aug.) (3)	20.17±0.26	20.58±0.28	20.84±0.35	20.66±0.30	20.28±0.25
POSTERIOR (5)	20.02±0.10	20.10±0.11	20.17±0.10	20.12±0.10	20.09±0.10
$\mathrm{POSTERIOR}_R$ (3)	20.22±0.02	20.25±0.03	20.29±0.03	20.27±0.02	20.26±0.02
COSMOS (px)					
NONE (3)	4.63±0.06	4.63±0.06	4.63±0.06	–	4.63±0.06
SOFT (3)	4.62±0.02	4.62±0.02	4.62±0.02	–	4.62±0.02
HARD (no aug.) (3)	**4.46±0.03**	5.36±0.04	5.99±0.06	–	5.42±0.28
HARD (5)	4.51±0.01	4.63±0.02	4.71±0.02	–	4.57±0.01
HARD+$\mathrm{LS}_{0.1}$ (3)	4.53±0.04	4.63±0.03	4.70±0.03	–	4.59±0.04
HARD+$\mathrm{LS}_{0.3}$ (3)	4.52±0.04	4.62±0.03	4.69±0.03	–	4.58±0.03
HARD_R (3)	4.58±0.03	**4.60±0.03**	**4.61±0.03**	–	4.60±0.03
POSTERIOR (no aug.) (3)	4.46±0.04	5.26±0.11	5.81±0.15	–	4.63±0.05
POSTERIOR (5)	4.50±0.05	4.63±0.05	4.72±0.06	–	**4.56±0.05**
$\mathrm{POSTERIOR}_R$ (3)	4.59±0.04	4.61±0.03	4.63±0.03	–	4.61±0.03

Table 9: All variants, in-distribution split, pooled minADE_6.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
AV2 (m)					
NONE (3)	0.559±0.004	0.559±0.004	0.559±0.004	0.559±0.004	0.559±0.004
SOFT (3)	0.550±0.003	**0.550±0.003**	**0.550±0.003**	**0.550±0.003**	**0.550±0.003**
HARD (no aug.) (3)	0.552±0.004	0.613±0.005	0.660±0.005	0.594±0.005	0.611±0.032
HARD (5)	0.553±0.004	0.560±0.005	0.565±0.005	0.557±0.005	0.558±0.005
HARD+$\mathrm{LS}_{0.1}$ (3)	0.552±0.004	0.559±0.005	0.564±0.005	0.556±0.005	0.556±0.005
HARD+$\mathrm{LS}_{0.3}$ (3)	0.552±0.005	0.558±0.005	0.563±0.005	0.555±0.005	0.556±0.005
HARD_R (3)	0.554±0.003	0.555±0.003	0.556±0.004	0.555±0.003	0.555±0.003
POSTERIOR (no aug.) (3)	0.553±0.002	0.576±0.003	0.592±0.004	0.580±0.002	0.559±0.002
POSTERIOR (5)	0.550±0.003	0.555±0.002	0.559±0.002	0.553±0.002	0.553±0.002
$\mathrm{POSTERIOR}_R$ (3)	**0.549±0.003**	0.552±0.004	0.554±0.005	0.551±0.004	0.551±0.004
SDD (px)					
NONE (3)	10.62±0.05	**10.62±0.05**	**10.62±0.05**	**10.62±0.05**	10.62±0.05
SOFT (3)	10.73±0.04	10.73±0.04	10.73±0.04	10.73±0.04	10.73±0.04
HARD (no aug.) (3)	10.67±0.05	11.13±0.03	11.44±0.07	11.25±0.06	12.44±0.98
HARD (5)	10.63±0.11	10.71±0.10	10.75±0.10	10.73±0.11	10.67±0.12
HARD+$\mathrm{LS}_{0.1}$ (3)	**10.57±0.13**	10.65±0.15	10.70±0.12	10.68±0.14	10.61±0.18
HARD+$\mathrm{LS}_{0.3}$ (3)	10.64±0.06	10.69±0.02	10.73±0.01	10.71±0.02	10.68±0.02
HARD_R (3)	10.60±0.07	10.63±0.09	10.64±0.09	10.64±0.09	10.62±0.09
POSTERIOR (no aug.) (3)	10.61±0.13	10.86±0.13	11.01±0.13	10.98±0.12	**10.59±0.12**
POSTERIOR (5)	10.64±0.12	10.69±0.12	10.71±0.12	10.70±0.13	10.65±0.12
$\mathrm{POSTERIOR}_R$ (3)	10.66±0.18	10.68±0.18	10.68±0.18	10.68±0.18	10.66±0.18

B AUDIT PROTOCOL AND RATER AGREEMENT

Windows were sampled from the shift split of a POSTERIOR run evaluated at 10% uniform corruption, restricted to agents whose label was neither flipped nor dropped. Low-trust: the 50 agents with the lowest g_i (all vehicles, $g_i = 0$). Control: 30 vehicles with $g_i \geq 0.8$ sampled with seed 0 from the

Table 10: Per-class minADE_6 on the shift split, AV2.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
AV2 (m) – pedestrian					
NONE (3)	0.389±0.003	0.389±0.003	0.389±0.003	0.389±0.003	0.389±0.003
SOFT (3)	0.376±0.005	0.376±0.005	**0.376±0.005**	0.376±0.005	0.376±0.005
HARD (no aug.) (3)	**0.332±0.001**	0.377±0.003	0.405±0.006	**0.357±0.002**	0.454±0.051
HARD (5)	0.360±0.006	0.380±0.005	0.392±0.005	0.375±0.004	0.387±0.005
HARD+$\mathrm{LS}_{0.3}$ (3)	0.360±0.002	0.376±0.002	0.388±0.003	0.372±0.002	0.383±0.003
POSTERIOR (5)	0.352±0.004	**0.369±0.003**	0.380±0.004	0.361±0.004	**0.359±0.003**
AV2 (m) – cyclist					
NONE (3)	0.923±0.019	0.923±0.019	0.923±0.019	0.923±0.019	0.923±0.019
SOFT (3)	0.917±0.003	0.917±0.003	0.917±0.003	0.917±0.003	0.917±0.003
HARD (no aug.) (3)	0.987±0.011	1.005±0.019	1.014±0.023	1.015±0.024	1.036±0.041
HARD (5)	0.917±0.007	0.919±0.007	0.921±0.006	0.919±0.007	0.920±0.009
HARD+$\mathrm{LS}_{0.3}$ (3)	**0.912±0.006**	**0.914±0.006**	**0.916±0.006**	**0.915±0.006**	**0.915±0.006**
POSTERIOR (5)	0.919±0.012	0.925±0.013	0.929±0.014	0.924±0.013	0.926±0.013
AV2 (m) – vehicle					
NONE (3)	0.555±0.001	0.555±0.001	0.555±0.001	0.555±0.001	0.555±0.001
SOFT (3)	**0.549±0.004**	**0.549±0.004**	**0.549±0.004**	**0.549±0.004**	**0.549±0.004**
HARD (no aug.) (3)	0.561±0.004	0.623±0.008	0.664±0.011	0.605±0.005	0.620±0.046
HARD (5)	0.557±0.006	0.561±0.007	0.564±0.007	0.559±0.006	0.558±0.006
HARD+$\mathrm{LS}_{0.3}$ (3)	0.554±0.004	0.558±0.004	0.560±0.004	0.555±0.004	0.555±0.004
POSTERIOR (5)	0.556±0.003	0.559±0.003	0.561±0.003	0.558±0.003	0.558±0.003

Table 11: Per-class minADE_6 on the shift split, SDD; the pedestrian block is referenced in Section 5.2.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
SDD (px) – pedestrian					
NONE (3)	10.62±0.19	10.62±0.19	10.62±0.19	10.62±0.19	10.62±0.19
SOFT (3)	10.65±0.23	10.65±0.23	10.65±0.23	10.65±0.23	10.65±0.23
HARD (no aug.) (3)	**10.52±0.07**	10.81±0.07	11.02±0.03	10.86±0.08	11.23±0.47
HARD (5)	10.66±0.14	10.71±0.14	10.76±0.14	10.74±0.12	10.68±0.12
HARD+$\mathrm{LS}_{0.3}$ (3)	10.62±0.19	10.66±0.22	10.70±0.26	10.68±0.22	10.67±0.21
POSTERIOR (5)	10.53±0.05	**10.57±0.04**	**10.60±0.04**	**10.59±0.04**	**10.55±0.03**
SDD (px) – cyclist					
NONE (3)	40.03±0.63	40.03±0.63	40.03±0.63	40.03±0.63	40.03±0.63
SOFT (3)	39.35±0.83	39.35±0.83	**39.35±0.83**	39.35±0.83	39.35±0.83
HARD (no aug.) (3)	39.21±0.39	40.42±0.50	41.18±0.46	40.47±0.43	46.26±5.08
HARD (5)	39.32±0.63	39.56±0.70	39.72±0.72	39.56±0.70	39.64±0.74
HARD+$\mathrm{LS}_{0.3}$ (3)	**39.03±0.76**	**39.25±0.76**	39.36±0.78	**39.25±0.76**	39.31±0.82
POSTERIOR (5)	39.13±0.30	39.31±0.32	39.47±0.30	39.32±0.31	**39.30±0.30**
SDD (px) – vehicle					
NONE (3)	**11.51±0.20**	**11.51±0.20**	**11.51±0.20**	**11.51±0.20**	11.51±0.20
SOFT (3)	11.65±0.08	11.65±0.08	11.65±0.08	11.65±0.08	11.65±0.08
HARD (no aug.) (3)	12.07±0.19	11.97±0.08	12.10±0.08	12.04±0.04	12.37±0.71
HARD (5)	11.57±0.27	11.58±0.24	11.59±0.23	11.56±0.27	**11.50±0.24**
HARD+$\mathrm{LS}_{0.3}$ (3)	11.54±0.33	11.53±0.33	11.57±0.31	11.52±0.34	11.52±0.38
POSTERIOR (5)	11.70±0.37	11.69±0.39	11.72±0.38	11.71±0.37	11.70±0.37

videos containing the low-trust set. Random: 50 agents sampled with seed 0. Box-area ratios use the median box area of the annotated class in the same video, visible frames only. Speed percentiles are computed on the training scenes with the same windowing as the model (stride 12 raw frames, 8 steps, visible frames only): vehicle $[0.00, 12.47]$, pedestrian $[0.00, 19.23]$, cyclist $[1.57, 63.20]$ px/step. The vehicle range is wider than that of cars alone ($[0.00, 3.25]$) because the study class pools cars with buses and carts; with the car range every moving vehicle window is atypical.

Table 12: Per-class minADE_6 on the shift split, COSMOS.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
COSMOS (px) – pedestrian					
NONE (3)	5.35±0.02	5.35±0.02	5.35±0.02	–	5.35±0.02
SOFT (3)	5.34±0.02	5.34±0.02	**5.34±0.02**	–	5.34±0.02
HARD (no aug.) (3)	5.27±0.04	6.54±0.07	7.51±0.11	–	6.10±0.34
HARD (5)	5.26±0.01	5.34±0.01	5.41±0.02	–	**5.29±0.01**
HARD+$\mathrm{LS}_{0.3}$ (3)	**5.25±0.01**	**5.33±0.02**	5.40±0.02	–	5.30±0.01
POSTERIOR (5)	5.29±0.03	5.37±0.03	5.44±0.04	–	5.31±0.03
COSMOS (px) – vehicle					
NONE (3)	4.05±0.09	4.05±0.09	4.05±0.09	–	4.05±0.09
SOFT (3)	4.04±0.02	4.04±0.02	**4.04±0.02**	–	4.04±0.02
HARD (no aug.) (3)	**3.80±0.02**	4.41±0.06	4.76±0.06	–	4.88±0.39
HARD (5)	3.90±0.03	4.05±0.03	4.13±0.04	–	3.99±0.03
HARD+$\mathrm{LS}_{0.3}$ (3)	3.93±0.06	4.05±0.05	4.13±0.05	–	4.00±0.05
POSTERIOR (5)	3.87±0.08	**4.03±0.07**	4.13±0.08	–	**3.96±0.07**

Table 13: Audit verdicts by the rule (windows). A human rater who judged the 100 low-trust and random windows before the rule was fixed agreed on 74/100. Of the 26 disagreements, 25 are the rater over-calling: 12 agents inside the class speed range called atypical (10 stationary pedestrians, one slow cyclist, one pedestrian at 14.5 px/step), 6 at the ≥ 0.40 lost-track boundary called atypical, 4 slow carts inside the pooled-vehicle range called atypical, and 3 moving carts called unclear that pass the track and box checks; the remaining one is a parked car with box ratio 2.1 that the rule marks unclear and the rater judged ok.

group	windows (tracks)	atypical	ok	unclear	wrong
low-trust vehicles ($g = 0$)	50 (41)	38	4	8	0
high-trust vehicles ($g \geq 0.8$)	30 (14)	0	23	7	0
random, all classes	50 (47)	2	36	12	0

C COVERAGE ANALYSIS

For each class we rank agents by a score, retain the top fraction, and report minADE_6 on the retained set (5-run mean, POSTERIOR runs, shift split, clean labels). Scores: trust g_i; negative ensemble disagreement; oracle (true error, i.e. the best achievable ranking).

Table 14: minADE_6 at coverage 100/90/80/70% and Spearman correlation of the score with error, shift split, clean labels, 5-run means.

dataset / class	score	100%	90%	80%	70%	ρ(score, err)
AV2 pedestrian	trust	0.35	0.34	0.36	0.38	+0.29
	ens. disagreement	0.35	0.31	0.30	0.29	−0.56
	oracle	0.35	0.25	0.20	0.17	
AV2 cyclist	trust	0.92	0.96	1.04	1.13	+0.51
	ens. disagreement	0.92	0.81	0.74	0.68	−0.66
	oracle	0.92	0.63	0.50	0.41	
AV2 vehicle	trust	0.56	0.52	0.48	0.48	−0.03
	ens. disagreement	0.56	0.44	0.35	0.26	−0.77
	oracle	0.56	0.32	0.22	0.14	
SDD pedestrian	trust	10.53	9.70	10.33	11.19	+0.39
	ens. disagreement	10.53	8.33	7.31	6.35	−0.78
	oracle	10.53	6.56	5.00	3.83	
SDD cyclist	trust	39.13	42.32	43.69	43.75	+0.40
	ens. disagreement	39.13	36.24	35.03	33.54	−0.45
	oracle	39.13	29.76	25.47	22.02	
SDD vehicle	trust	11.70	7.43	2.00	1.08	−0.49
	ens. disagreement	11.70	5.84	2.01	0.98	−0.65
	oracle	11.70	3.48	1.18	0.79	
COSMOS vehicle	trust	3.87	3.79	3.83	3.87	−0.30
	ens. disagreement	3.87	2.01	1.51	1.36	−0.72
	oracle	3.87	1.20	0.69	0.53	
COSMOS pedestrian	trust	5.29	5.51	5.83	6.18	+0.40
	ens. disagreement	5.29	4.63	4.19	3.81	−0.65
	oracle	5.29	3.66	2.93	2.39	

The same pattern holds on the in-distribution splits and under 30% label corruption (not shown). Partial Spearman correlation of trust with log error after residualising both on log speed (5-run mean, clean labels): AV2 id $-0.12/+0.09/+0.01$ and shift $-0.11/+0.07/+0.01$ (pedestrian/cyclist/vehicle); SDD id $-0.18/+0.11/-0.17$ and shift $+0.07/+0.02/-0.68$; COSMOS shift $+0.05/+0.14$ (vehicle/pedestrian).

D SYNTHETIC BENCHMARK

Three agent types are simulated with *overlapping* speed ranges (pedestrian 0.8–2, cyclist 3–6, vehicle 3–10 m/s) but different future behavior: vehicles brake and turn, cyclists drift in heading, and 30% of pedestrians stop. The label therefore carries information about the future that the observed past does not, which is the regime in which conditioning on type is useful. A fraction of agents are *behaviorally anomalous*: they retain their correct label but move according to another type's dynamics. This is the situation the real-data audit (Section 5.5) finds on SDD, here with ground truth. Scenes contain 2–6 agents at 0.4 s per step with 8 observed and 12 predicted steps; 12,000 training scenes, 2,000 validation, 3,000 test. The shift split slows vehicles to 2.5–6 m/s (more overlap with cyclists), reduces turning, and raises the anomaly rate from 10% to 25%. All variants are trained as on the real data. POSTERIOR is ahead of HARD on clean labels in-distribution (-2.7% pooled, 3/3 runs, $p = 0.16$) and within run-to-run variation everywhere else (shift: -1.6%, $p = 0.45$; under corruption $|\Delta| \leq 1.4\%$). The trust score detects injected label flips with AUROC 0.90 (in-distribution) and 0.77 (shift), and detects the behaviorally anomalous agents—whose labels are correct—with AUROC 0.96 and 0.91. The second number is the one that matters for the interpretation in Section 5.5: where a label is correct but the motion is another type's, trust drops, by design and against ground truth.

Table 15: Synthetic benchmark, pooled minADE_6, both splits.

model (runs)	clean	unif. 30%	unif. 50%	conf. 30%	miss. 50%
id split					
NONE (3)	0.461±0.008	0.461±0.008	0.461±0.008	0.461±0.008	0.461±0.008
SOFT (3)	0.449±0.018	**0.449±0.018**	**0.449±0.018**	**0.449±0.018**	**0.449±0.018**
CONST (3)	0.446±0.019	0.484±0.038	0.509±0.050	0.494±0.042	0.468±0.030
GATE-MIX (3)	**0.434±0.004**	0.480±0.029	0.510±0.054	0.496±0.047	0.458±0.013
HARD (3)	0.450±0.004	0.495±0.023	0.527±0.040	0.509±0.039	0.470±0.011
POSTERIOR (3)	0.438±0.008	0.488±0.031	0.521±0.050	0.506±0.044	0.469±0.023
shift split					
NONE (3)	0.483±0.004	0.483±0.004	0.483±0.004	0.483±0.004	0.483±0.004
SOFT (3)	**0.483±0.017**	**0.483±0.017**	**0.483±0.017**	**0.483±0.017**	**0.483±0.017**
CONST (3)	0.492±0.032	0.528±0.048	0.553±0.061	0.539±0.053	0.509±0.043
GATE-MIX (3)	0.484±0.013	0.530±0.042	0.561±0.063	0.546±0.058	0.498±0.020
HARD (3)	0.496±0.013	0.537±0.031	0.565±0.045	0.551±0.044	0.511±0.019
POSTERIOR (3)	0.488±0.026	0.534±0.045	0.567±0.061	0.551±0.057	0.512±0.035

E NEGATIVE ABLATION: MIXTURE OF EXPERTS OVER TYPES

A variant that runs the decoder once per candidate type and draws the K_m output modes from the joint mixture $q_i(k)\, p(m \mid k)$ was unstable on two-class data (a mirror solution with permuted type branches) and gave no improvement over POSTERIOR on the three-class datasets in preliminary runs; it was dropped before the multi-run protocol and is not reported.

F HYPERPARAMETERS AND REPRODUCIBILITY

GRU hidden $d = 128$; 4 attention heads; type embedding $d_e = 32$; decoder MLP $2d \to 2d \to K_m T_p \cdot 2$ with $K_m = 6$, $T_p = 12$; probe MLP $d \to 64 \to K$; reliability MLP $2d + 1 \to 64 \to 1$ with bias initialized so that $\rho = 0.9$; ensemble of 5 heads $d \to 64 \to 2$ with bootstrap rate 0.7; loss weights $\lambda_{\mathrm{ch}} = \lambda_{\mathrm{ens}} = 0.5$; batch 128 (64 on SDD); AdamW, learning rate 10^{-3}, weight decay 10^{-4}, one-cycle schedule; gradient clipping 1.0; 40 epochs (COSMOS, AV2) or 120 (SDD); best epoch by validation ADE on clean labels. Training corruption: 10% uniform flips and 10% missing per batch. Range-augmented variants draw the flip rate per batch from $U[0, 0.5]$. Software: PyTorch

2.4.1, CUDA 12.1; NVIDIA T1000. Training is not bit-deterministic on this setup. Table generation: `collect_results.py` and `make_tables.py` over `runs/*/*/eval.jsonl`; audit: `sdd_audit.py`, `enrich_audit.py`, `audit_rules.py` with seed 0; secondary analyses (calibration, attribution, coverage, dataset statistics): `analysis.py`.