Rethinking Training & Inference for Forecasting: Linking Winner-Take-All back to GMMs

Qiyuan Wu[1], Katie Z Luo[2], Bharath Hariharan[1], Wei-Lun Chao[3], and Mark Campbell[1]

[1]Cornell University, USA, [2]Stanford University, USA,[3]Boston University, USA
{qw253,mc288}@cornell.edu; {katieluo}@stanford.edu;
{bharathh}@cs.cornell.edu; {chao209}@bu.edu

Abstract. Trajectory forecasting for autonomous driving has advanced rapidly, yet representative models often produce uninformative posteriors over forecast modes, causing problems for mode pruning. We trace this to a modeling-training mismatch: forecasters are typically modeled as conditional Gaussian mixture models (GMMs) but trained with a winner-take-all (WTA) loss that assigns each sample to its nearest mode. We argue that this K-means-like hard assignment (one-hot), while preventing mode collapse, is the source of uninformative mode probabilities: it over-segments the trajectory space, ignores relatedness among nearby modes, and yields assignment instability under small perturbations. Guided by this lens, we introduce two post-hoc treatments: (1) test-time posterior-weighted merging that aggregates nearby candidate trajectories; and (2) a one-step expectation-maximization (EM) update that replaces hard labels with soft responsibilities, sharing probability mass across neighboring modes. Across several WTA-trained architectures, these lightweight steps produce more informative, faithfully ranked mode posteriors and strengthen final forecasts on popular displacement metrics—without retraining. Our analysis unifies recent design choices through a GMM-vs-K-means perspective and offers principled, practical corrections that better align training objectives with inference.

Keywords: Motion Forecasting · Gaussian Mixture Models · Winner-Take-All Training · Autonomous Vehicles

1 Introduction

Trajectory forecasting in autonomous driving, the task of predicting where other agents (*i.e.*, cars, pedestrians, and cyclists) will go, is critical for planning safe maneuvers for self-driving vehicles in human-populated traffic. There have been numerous industry efforts to establish practical challenges [5, 12, 34], which in turn inspire learning-based technical advances. Fundamental to this is uncertainty estimation, since future motion is inherently multi-modal and ambiguous. For instance, a vehicle pausing at a stop sign can either go forward or be ready to turn, and a downstream self-driving planning stack needs to account for both to ensure a safe action plan. Beyond generating multiple forecasts, it is also

crucial to rank them by likelihood: without a meaningful ordering, an abundance of low-probability hypotheses can trigger overly conservative behaviors (*e.g.*, abrupt braking) in otherwise normal situations—both uncomfortable and potentially unsafe.

This work focuses on one mainstream model family [9, 23, 30–32, 35], which generate multiple candidate trajectory predictions, each with an associated probability. This design sensibly incorporates uncertainty directly into the predicted futures, and offering multiple candidates gives more flexibility for the subsequent downstream motion planning. Prior works show that this class of motion forecasting models is indeed capable of producing future trajectory candidates that cover most of the true trajectories—often capturing even outliers [3]. In practice, the model often predicts an initial set of over-complete trajectories (usually 64) and adopts a post-hoc mode sampling or aggregation process to obtain the final compact set of trajectories for downstream motion planning of the ego-vehicle. Today, this setting has become the de-facto benchmark evaluation, as it balances the flexibility of obtaining a compact and tractable subset for real-time deployment while providing a better coverage of safety-critical possible future motions. The bottleneck, however, is in the assigned probability to each candidate: we find that, while the prediction does model an associated probabilistic mixture weight for each predicted trajectory, the assigned probability is often uninformative, *e.g.*, assuming low likelihood for candidates that are common and correct future behaviors, or fluctuating probabilities assigned to similar candidates. Thus, a naive post-hoc selection methods such as selecting the highest "likelihood" modes may yield trajectories that retain only a small fraction of the total predictive likelihood.

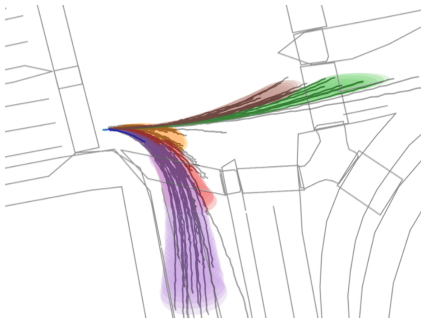


Fig. 1: 64 modes predicted by Wayformer. We visualize a Gaussian mixture of 5 components overlaid from directly predicting the smaller set. Observe that it suffers from over-segmentation, with many predictions per mode, despite many of them being unlikely. The predicted trajectories are colored in black, the ground truth trajectory colored blue, and the map polylines in gray.

This work takes a classical machine learning approach to answer the question: *What is a principled way of obtaining a compact set of trajectories despite uninformative predicted likelihoods*? To approach this, we begin by exploring why the likelihoods predicted under a broad swath of motion forecasting objectives [9, 23, 30–33] are often uninformative. Our investigations show a fundamental mismatch between their training objective and the assumed inference-time probabilistic model. To capture the uncertainty and diversity of future trajectories, these representative algorithms adopt a Gaussian mixture model (GMM) conditioned on the past trajectory. The standard objective to learn GMM is the evidence lower bound (ELBO), which can be approximated by an Expectation-

Maximization (EM) style training—interleaving between assigning each sample *softly* to modes, and updating the candidate predictions and probability assignments. However, such an objective is intractable over a large dataset, and directly optimizing with gradient descent often yields collapsed modes, which is especially undesirable. In practice, these methods adopt a winner-take-all (WTA) assignment, where the soft assignment to modes is replaced with a hard assignment, thus preventing the collapse of all modes towards the single sample update. While this simple change effectively enables the diverse predictions, our key insight is that such a hard assignment changes the objective to a K-Means styled objective [19], thus losing its probabilistic interpretation. This insight sheds light on why the predicted mixture likelihoods are often uninformative, and why models trained under such objective are able to effectively predict trajectories that specialize without mode collapse.

Indeed, we argue that the winner-take-all objective has many merits, as it inherits the strengths of K-Means predictions, and is capable of capturing the diverse behaviors necessary in motion forecasting methods. This understanding also sheds light on why these models tend to over-segment —*i.e.*, generate many predictions over— a single intention mode, such as "move-forward". In section 4, we analyze the objective to demonstrate the perspective that the WTA objective maps closely to a K-Means clustering objective. With this insight, we propose to use an aggregation and single-step expectation maximization in section 5 to mitigate the over-segmentation characteristic and to replace hard labels with soft responsibilities for more informative, faithfully ranked mode posteriors. Our key contribution are providing theoretical groundings and interpretations for this class of methods: why we need them, and how they work. Finally, in section 6 we demonstrate that our simple treatments improve the distance error prediction performance over naive greedy selection over the predicted trajectory likelihoods without the need for retraining. Our contributions are as follows:

- We offer a unified GMM-*vs*-K-means perspective on the winner-take-all trajectory forecasting objective.
- We offer a principled, practical correction to align the training objective with inference-time probabilistic modeling assumptions.
- We empirically verify our findings with three representative winner-take-all models on post-hoc aggregation strategies.

2 Related Works

Motion Forecasting for Self-Driving. Motion forecasting has been a longstanding problem in self-driving research. The learning-based methods primarily formulate it as a multi-agent sequence prediction problem, where given some history of the agent's (*i.e.* another vehicle or a pedestrian) location, the goal is to predict the future locations [2, 20]. Early approaches predict one or more waypoints trajectories of the actors of interest [1, 15, 24]. Other works explored how to model uncertainty either as independent Gaussian distributions over future

Model	training	post-hoc
Wayformer [23]	WTA	Merging
Multipath++ [33]	WTA	Merging
MTR-e2e [30]	WTA	Non-Maximum Suppression
EMP [25]	WTA	None

Table 1: Training objective and post-hoc method of motion prediction models.

timesteps [7, 26], or as non-parametric occupancy predictions [16, 17, 21]. Another branch of works aimed at identifying intentions [14, 18, 27, 36], either as the end goal or as a co-objective towards multimodal motion forecasting. These works often obtain very diverse, context-dependent behaviors, but are slow to run in real-time. To handle the necessity of fast runtime forward passes, methods opted for implicit likelihoods to model intention as sampling [8, 10, 15, 29]. However, the lack of an explicit likelihood makes these models hard to estimate uncertainty that is needed for downstream ego-vehicle planning. A few works instead leverage an explicit multimodal Gaussian mixture model (GMM) representation to predict implicit intentions as mixtures, and directly regress timestep-independent Gaussian future predictions [8,9,11,30,32,33]. These yield interpretable uncertainties, but are often uncalibrated in practice [6]. In the multimodal, GMM-based approaches, popularized by [9], many works optimize their Gaussian Mixture models using a diversity-inducing loss, where the likelihoods are trained with a winner-take-all objective [23, 30, 32, 33]. This method of optimization yields diverse, multimodal predictions and stable convergences.

Post-hoc Trajectory Selection Many motion predictors, such as MTR [30], Multipath++ [33], and Wayformer [23], standardly employ 64 initial modes ($K = 64$) for a better coverage of possible trajectories, and adopt a mode sampling or aggregation process to obtain the final M modes for benchmark evaluation ($M \ll K$, usually 5 or 6). This configuration is argued to be computationally tractable for the predictor while providing a sufficiently over-complete set to enable the stable learning of diverse motion patterns. Table 1 summarizes the training objective and post-hoc mode selection method of some models. These post-hoc methods allow practitioners to modify or down-select trajectory modes without retraining, providing greater test-time flexibility in the output representation. However, despite their practical importance, there is currently no unified framework or systematic analysis that connects these post-hoc trajectory selection/aggregation procedures with the underlying training objectives.

3 Preliminaries

3.1 Motion Forecasting

Motion forecasting for autonomous driving requires predicting multiple plausible future trajectories of surrounding agents. Key desiderata include: (1) multimodality to capture diverse behavioral intentions, (2) accurate likelihood esti-

mates for uncertainty-aware planning, and (3) computational efficiency for real-time deployment.

Problem Formulation. Given history states of all agents and scene map context, predict the target agent's future trajectory in the form of M candidates $\hat{\mathbf{y}}_m^{(t)}, t \in \{1, ..., T\}, m \in \{1, ..., M\}$ over a prediction horizon T. Each timestep is predicted independently, conditioned on the inputs and the selected mode, so the model factorizes the joint prediction across time. In this paper, we assume all predictions are conditioned on inputs (which are dropped in the equations for simplicity), and drop the datapoint index.

GMM Representation. Gaussian Mixture Models (GMMs) naturally address many of these requirements and have become a popular choice for modeling future trajectories in motion forecasting [23, 30, 32]. GMMs provide multimodality through mixture components representing distinct behavioral modes, principled likelihood estimates for uncertainty-aware planning, and a compact parametric form that is efficient for training and inference. These advantages have led many state-of-the-art motion forecasting methods to adopt GMM-based output representations, making them a de facto standard in the field. Formally, given K mixture components, the GMM negative log-likelihood objective is:

$$\mathcal{L}_{\text{GMM}} = -\log \sum_{k=1}^{K} \hat{p}_k \cdot \prod_{t=1}^{T} \mathcal{N}(\mathbf{y}^{*(t)} | \hat{\mathbf{y}}_k^{(t)}, \Sigma_k^{(t)}) \tag{1}$$

where T is the prediction horizon, $\mathbf{y}^{*(t)}$ is the ground truth at time t, $\hat{\mathbf{y}}_k^{(t)}$ denotes the k-th predicted trajectory at time t, Σ_k the covariance (often simplified to a fixed isotropic covariance), and $\hat{p}_k$ its mixture weight.

Post-hoc Trajectory Selection In real-world applications, there exists a tradeoff between more modes and the computation constraints: adding modes yields lower minimum displacement errors (recall) but at the cost of computation latency, a sensitivity of real-time applications. Previous works from Luo et al., and Roh et al. [22, 28] emphasize that incorporating multi-modal predictions from multiple agents substantially increases planning complexity, often requiring mode pruning or reduction to meet real-time constraints. Additionally, training the predictor itself requires generating a large number of initial trajectory candidates. Both [30] and [23] employ 64 initial trajectories, arguing that this number enables stable learning of diverse motion patterns while remaining computationally tractable, with downstream tasks subsequently selecting or pruning the top M trajectories based on their specific constraints. Therefore, we consider a setting in which the predictor produces an over-complete set of K candidate trajectories, while only a limited subset of M can be feasibly utilized for final evaluation.

Evaluation Metrics. The motion forecasting community has developed several metrics to evaluate these desiderata [4,7,9,12,15]. The most common displacement-

based metrics are minimum Average Displacement Error (minADE) and minimum Final Displacement Error (minFDE), which compute the L2 distance between the ground truth and the closest prediction from a set of M trajectories:

$$\text{minADE} = \min_{m \in \{1,\dots,M\}} \frac{1}{T} \sum_{t=1}^{T} \|\hat{\mathbf{y}}_m^{(t)} - \mathbf{y}^{*(t)}\|_2 \tag{2}$$

$$\text{minFDE} = \min_{m \in \{1,\dots,M\}} \|\hat{\mathbf{y}}_m^{(T)} - \mathbf{y}^{*(T)}\|_2 \tag{3}$$

where $\hat{\mathbf{y}}_m^{(t)}$ denotes the m-th predicted trajectory at time t, $\mathbf{y}^{*(t)}$ is the ground truth at time t, and T is the prediction horizon. The Waymo Motion Prediction Challenge [12] introduced Miss Rate, which computes the ratio of scenes where all predictions exceed a scenario-based final displacement error.

3.2 Winner-Take-All Objective

Motion forecasting models ideally aim to learn a distribution over future trajectories, commonly approximated as a Gaussian Mixture Model (GMM). However, directly optimizing the GMM objective often leads to mode collapse, where the model learns to predict nearly identical trajectories across all mixture components to minimize the loss for any given ground truth. This degeneracy defeats the purpose of multimodal prediction, as the model fails to capture the diversity of plausible future behaviors. Moreover, GMM objective requires iterative EM training, which is computationally heavy.

To address these issues, practitioners have adopted the winner-take-all (WTA) training paradigm. Introduced in [9], instead of updating all predictions based on their likelihood under the ground truth, WTA selects only the best prediction and updates its parameters:

$$\mathcal{L}_{\text{WTA}} = -\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{\{k = \arg\min_j \|\hat{\mathbf{y}}_j - \mathbf{y}^*\|_F\}} \Big[\log \hat{p}_k + \sum_{t=1}^{T} \log \mathcal{N}(\mathbf{y}^{*(t)} | \hat{\mathbf{y}}_k^{(t)}, \Sigma_k^{(t)}) \Big] \tag{4}$$

This objective encourages diverse predictions by ensuring that only one component needs to explain each training sample, preventing the collapse to a single mode, and is thus popular in various methods in motion forecasting that output GMM representations [23, 30, 32, 35]. While this approach successfully maintains prediction diversity, we observe that it fundamentally changes the nature of what the model learns — a mismatch we analyze in detail in section 4.

4 What is the WTA Objective Doing?

In this work, we argue that WTA is actually a hybrid of GMM and K-Means clustering. We aim to study the winner-take-all objective that is prevalent in training many GMM outputs of prediction models [9, 23, 30–33].

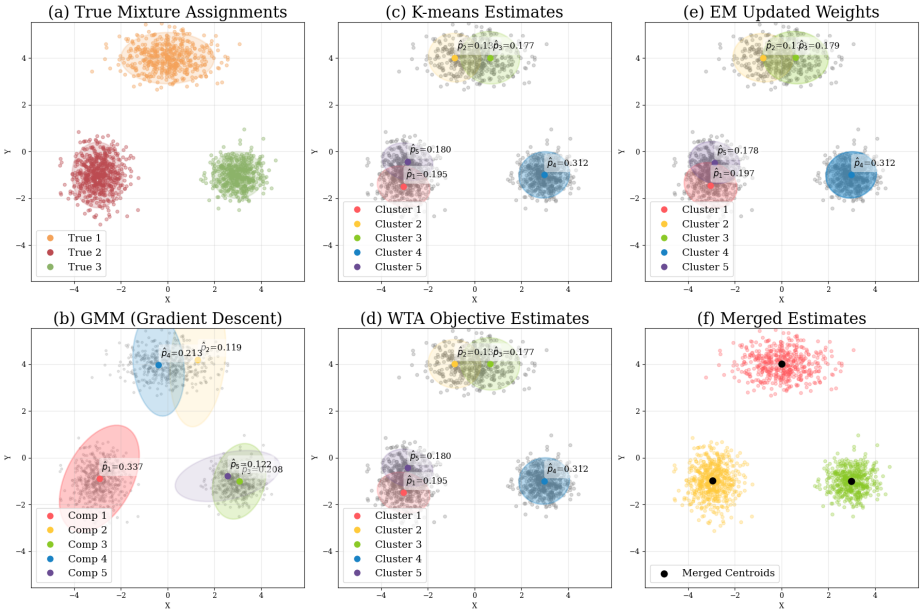


Fig. 2: Minimal 2D Example. To demonstrate how the winner-takes-all objective influences the final predicted modes and probability, we construct a 3-mode Gaussian mixture (a), and fit a GMM with gradient descent on the parameters (b) and with the WTA objective (d). Observe that the parameter estimates exhibit mode collapse; while the WTA objective covers diverse modes but over-segments the space, which is identical to the K-means result (c). We propose to update the mixture weights with a single EM step (e) or use weighted cluster merging (f), which yields more valid assignments. Notice in (d) $\hat{p}_1$, $\hat{p}_4$, and $\hat{p}_5$ are the top 3, causing a miss on the top true Gaussian; In (e), the top 3 clusters are now $\hat{p}_1$, $\hat{p}_3$, and $\hat{p}_4$, covering all 3 true Gaussians.

4.1 Revisiting K-Means Clustering

Recall that the K-means clustering objective is to learn a partition over the observations into clusters in which each observation belongs to the cluster with the nearest mean, or cluster exemplar [19]. The objective is to minimize the total variance within the cluster. Formally, given a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$, the objective is to find K sets $\mathbf{S} = \{S_1, \cdots, S_K\}$:

$$\arg\min_{\mathbf{S}} \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathbf{S_k}} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2, \tag{5}$$

where each centroid $\boldsymbol{\mu}_k$ is the mean of the cluster k, *i.e.* $\boldsymbol{\mu}_k = \frac{1}{|S_k|} \sum_{\mathbf{x} \in \mathbf{S_k}} \mathbf{x}$.

Observe that we can rewrite this objective as an iteration over the data points instead,

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{\{k = \arg\min_j \|\mathbf{x} - \boldsymbol{\mu}_k\|\}} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2, \tag{6}$$

where the indicator $\mathbb{1}_{\{k=\arg\min_j \|\mathbf{x}-\boldsymbol{\mu}_k\|\}}$ assigns the points to a cluster if it is closest to its centroid. This can be viewed as optimizing the minimum distances across all datapoints within a cluster. Typically, this objective is maximized using an iterative refinement technique [19], separating the cluster assignment step from the centroid update.

4.2 Mapping WTA to K-Means Objective

We hypothesize that the winner-take-all (WTA) objective's hard minimum assignment exhibits similar behaviors to K-means objective. Observe that we can decompose the WTA objective in Equation 4 into the mixture weight likelihood term and the Gaussian component regression term:

$$\mathcal{L}_{WTA} = -\sum_{i=1}^{n} \Big(\underbrace{\sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} \log \hat{p}_k}_{\text{Mode cross-entropy } \mathcal{L}_{class}} + \underbrace{\sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} \sum_{t=1}^{T} \log \mathcal{N}(\mathbf{y}^{*(t)} | \hat{\mathbf{y}}_k^{(t)}, \Sigma_k^{(t)})}_{\text{Mode Gaussian parameter regression } \mathcal{L}_{reg}} \Big) \tag{7}$$

Looking only at the regression term for a particular data point i, observe that

$$\begin{aligned}
\hat{\mathbf{y}}^* &= \arg\min_{\hat{\mathbf{y}}} \sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} \sum_{t=1}^{T} -\log \mathcal{N}(\mathbf{y}^{*(t)} | \hat{\mathbf{y}}_k^{(t)}, \Sigma_k^{(t)}) \\
&= \arg\min_{\hat{\mathbf{y}}} \sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} \sum_{t=1}^{T} \|(\Sigma_k^{(t)})^{-1/2} (\mathbf{y}^{*(t)} - \hat{\mathbf{y}}_k^{(t)})\|^2 \\
&= \arg\min_{\hat{\mathbf{y}}} \sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} \|(\Sigma_k)^{-1/2} (\mathbf{y}^* - \hat{\mathbf{y}}_k)\|_F^2 \\
&= \arg\min_{\hat{\mathbf{y}}} \sum_{k=1}^{K} \mathbb{1}_{\{\cdot\}} D_M(\mathbf{y}^*, \hat{\mathbf{y}}_k; \Sigma_k)^2,
\end{aligned}$$

where $\mathbf{y}^*$ and $\hat{\mathbf{y}}_k$—with a slight abuse of notation—are the timestep-stacked ground-truth and prediction trajectory, and Σ_k is the block diagonal stacked covariances at sample i. The condition of the indicator is from Equation 4 and dropped for brevity; it is the indicator for the closest prediction to the data point according to the Frobenius norm. D_M is the Mahalanobis distance under the Gaussian distribution of each mode k.

Setting the gradient of WTA objective function with respect to $\hat{\mathbf{y}}_k$ to zero yields

$$\frac{\partial \mathcal{L}_{WTA}}{\partial \hat{\mathbf{y}}_k} = -2\, \Sigma_k^{-1} \sum_{i \in \mathcal{C}_k} \big(\mathbf{y}_i^* - \hat{\mathbf{y}}_k\big) = \mathbf{0} \tag{8}$$

The optimal $\hat{\mathbf{y}}_k$ is exactly the mean of the cluster: $\hat{\mathbf{y}}_k^* = \frac{1}{N_k} \sum_{i \in \mathcal{C}_k} \mathbf{y}_i^*$. Here $\mathcal{C}_k = \{i : \arg\min_j \|\mathbf{y}_i^* - \hat{\mathbf{y}}_j\|_F = k\}$ with $N_k = |\mathcal{C}_k|$, $\Sigma_k \succ 0$. *The identical optimal prediction trajectory suggests an equivalence between WTA and K-means.*

Such a re-mapping to the K-means objective yields interesting insights into why the WTA objective exhibits behaviors that are mode covering, but often oversegments the space.

To demonstrate intuitively what the WTA objective optimizes as compared to the GMM model, we turn to a minimal example in Fig. 2, where we look at a simple 2-dimensional case. Observe that while the WTA objective (Fig. 2d) does not exhibit mode collapse, like in the GMM representation (Fig. 2b), it often breaks up modes and oversegments the space, which is characteristic of K-Means clustering (Fig. 2c). We analyze the consequences of this behavior in the following section. In section 5, we introduce post-hoc treatments to mitigate both failure modes — replacing hard assignments with soft responsibilities (Fig. 2e) and merging oversegmented predictions into coherent modes (Fig. 2f).

4.3 Consequences of Hard Assignment

The equivalence between WTA and K-means clustering has two concrete negative consequences for trajectory forecasting:

Oversegmentation. When the number of learned modes K exceeds the number of true behavioral modes, hard assignment encourages the model to partition a single true mode into multiple learned modes, each claiming exclusive ownership of a subset of training samples. This is the characteristic oversegmentation behavior of K-means: rather than concentrating multiple mode anchors on a single high-density region, it spreads them across an artificially fragmented partition of that region (Fig. 2c). The result is an over-complete set of predicted trajectories that fragments the plausible trajectory space beyond what is semantically meaningful. This characteristic oversegmentation is visualized in Fig. 1, where any single intention-mode (colored by Gaussian mixtures) is fragmented across many predicted modes.

Uninformative Mode Probabilities. Oversegmentation directly undermines the quality of predicted mode probabilities. A single high-probability true mode — say, "turn right" — may be represented by many learned modes, each capturing only a fraction of the associated training samples (Fig. 1). Consequently, each individual learned mode accumulates only a small share of the probability mass, even though their union corresponds to a highly likely future behavior. When the top-K predictions are selected at inference time, these fragmented modes are each individually ranked as low-probability, and may be suppressed or ignored — despite collectively representing a dominant intention. This mismatch between individual mode probability and true behavioral likelihood is the source of the uninformative posteriors we observe in practice.

5 Proposals for Prediction Selection

5.1 Test-Time Merging

When the model's number of modes exceeds the number of components of the true distribution, the WTA objective tends to break up modes and over-segment the space (Fig. 2d). Merging the broken-up clusters can recover the true distribution components, if clusters are grouped correctly (Fig. 2f).

Therefore, we suggest a test-time merging method based on weighted K-means clustering to be used for models trained with WTA objective, which is simpler compared to the existing similar implementations in Wayformer [23] and MultiPath++ [33]. In detail, it adopts an iterative process like K-means clustering. Given initially predicted trajectories $\hat{\mathbf{y}}_k, k \in \{1, ..., K\}$ and we want M cluster centers $\bar{\mathbf{y}}_m, m \in \{1, ..., M\}, M < K$ as our final output. In each iteration, 1) Assign initially predicted trajectories $\hat{\mathbf{y}}_k$ to the closest cluster (Euclidean distance of the final waypoints $\|\hat{\mathbf{y}}_k^{(T)} - \bar{\mathbf{y}}_m^{(T)}\|$); 2) Recompute cluster centers using the weighted mean formula: $\bar{\mathbf{y}}_m = \frac{\sum_{k \in C_m} \hat{p_k} \hat{\mathbf{y}}_k}{\sum_{k \in C_m} \hat{p_k}}$; 3) Repeat until cluster centers stabilize. The stabilized $\bar{\mathbf{y}}_m$ is the final output.

In the previous section, we have shown that this merging method theoretically aligns with the problem and targets exactly at the issues of WTA. We also demonstrate the effectiveness of this merging method through experiments.

5.2 One-Step Expectation-Maximization

Since the trajectories predicted under the WTA objective exhibit diverse, but yield uninformative mode prediction likelihoods, we can instead apply a post-processing step to convert the final modeled outputs into a true mixture model with a Bayesian update on the predicted likelihoods. We note that we can now turn to one step of expectation-maximization (EM) to run another iteration of training, replacing the hard label assignments during training with soft responsibilities. Specifically, the EM steps follow from the standard GMM updates, with the E-step:

$$w_k := p(k|\mathbf{y}^*) = \frac{p(\mathbf{y}^*|\hat{\mathbf{y}}_k, \Sigma_k) \cdot \hat{p}_k}{\sum_{z=1}^{K} p(\mathbf{y}^*|\hat{\mathbf{y}}_z, \Sigma_z) \cdot \hat{p}_z}, \tag{9}$$

where the likelihoods are obtained from our trained motion forecasting model, parameterized as a Gaussian.

We select to compute the M-step update over the training dataset maximizing the data distribution (minimizing the negative log likelihood $\mathcal{L}$):

$$\mathcal{L} = -\sum_{i=1}^{n} \log p(\mathbf{y}^*) = \sum_{i=1}^{n} \sum_{k=1}^{K} -w_k \log \hat{p}_k, \tag{10}$$

which can be interpreted as the cross-entropy over the *soft* assignments.

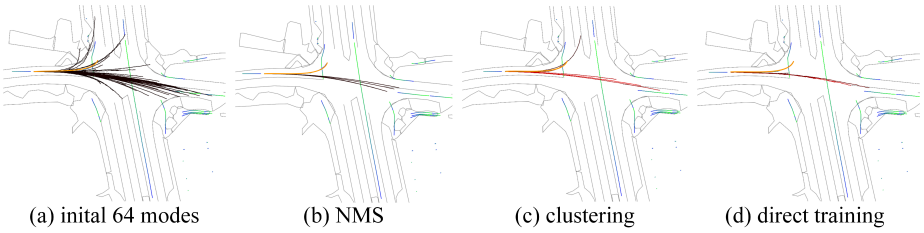


Fig. 3: Qualitative results of Wayformer. The predicted trajectory modes are colored in red(high probability) to black(low probability), logged trajectories are colored from green to blue, progressing across time, the map polylines are in gray, and the ground truth future trajectory of the target vehicle is in orange.

6 Experiments

We compare one-step expectation maximization with the original WTA method over multiple variants of mode selection methods, showing the improvement of predicted mode probabilities.

Datasets and evaluation metrics We evaluate accuracy metrics, informative metrics, and qualitative results on NuScenes Prediction [4] and Waymo Open Motion (WOMD) [12] datasets. The standard prediction metrics (minADE, minFDE, miss rate, and brier-minFDE) are used in this work. We evaluate with the number of the final output modes $M = 5$.

Mode selection variants We have 3 major types of post-hoc mode selection:

- Greedy selection: choose the top M modes with the highest predicted scores.
- Non-maximum suppression (NMS): choose the modes with the highest predicted scores while suppressing nearby modes whose predicted trajectories are too similar (i.e., within a certain distance threshold) to avoid redundant predictions and form the final M modes, which is adopted by MTR [32].
- Clustering-based merging: merge the initial predictions into M modes (Section 5.1). The initial predicted modes are K-means clustered through an iterative process, and the M centroids are output as the final M modes after converging. A similar method is used in Wayformer [23].

Additionally, we include a direct training approach that fixes the decoder to only predict M modes. This method skips the post-hoc mode selection, but suffers from the inflexibility when a different number of modes is needed at test time.

Implementation details The analysis is implemented with three existing models: Wayformer [23], MTR-e2e [32], and EMP [25]. We apply variants of post-hoc mode selection to all models using a unified framework provided by [13]. The model decoder predicts $K = 64$ initial modes, and compresses to $M = 5$ modes. A two-layer perceptron head is added to the mode probability head for one-step EM, which is finetuned over training data. Unless otherwise specified, the distance threshold of NMS is 1.2m displacement of the final waypoint.

	Method	brier_FDE5 ↓	minADE5 ↓	minFDE5 ↓	miss rate5 ↓
MTR-e2e	greedy	6.016	2.115	5.369	0.623
	NMS	5.089	1.793	4.440	**0.521**
	merging	**3.432**	**1.268**	**2.848**	0.535
	direct training	3.450	1.268	2.938	0.508
EMP	greedy	5.508	1.987	4.863	0.596
	NMS	5.048	1.857	4.401	0.525
	merging	**3.086**	**1.187**	**2.533**	**0.507**
	direct training	3.038	1.156	2.445	0.475
Wayformer	greedy	8.935	3.302	8.287	0.712
	NMS	7.679	2.871	7.024	0.637
	merging	**3.646**	**1.439**	**3.050**	**0.627**
	direct training	3.584	1.399	2.976	0.584

Table 2: Accuracy of different post-hoc selection on NuScenes. Clustering is better than greedy selection, which shows the mode probability is uninformative and the modes are over-segmented.

	Method	brier_FDE5 ↓	minADE5 ↓	minFDE5 ↓	miss rate5 ↓
MTR-e2e	greedy	6.982	2.421	6.335	0.634
	NMS	6.498	2.273	5.858	**0.591**
	merging	**4.110**	**1.530**	**3.546**	0.602
	direct training	3.930	1.477	3.387	0.556
EMP	greedy	9.906	2.887	9.260	0.665
	NMS	7.300	2.562	6.660	**0.631**
	merging	**4.442**	**1.677**	**3.855**	0.658
	direct training	4.057	1.482	3.460	0.577
Wayformer	greedy	11.645	4.143	10.994	0.695
	NMS	11.423	4.080	10.775	**0.684**
	merging	**5.529**	**2.025**	**4.968**	0.757
	direct training	4.288	1.562	3.710	0.624

Table 3: Accuracy of different post-hoc selection on WOMD.

6.1 WTA-Learned Mode Probability

We begin by evaluating different post-hoc mode selection methods on the NuScenes dataset, and compare the results of directly using the predicted likelihoods for sampling-based post-hoc methods (greedy and NMS selection) with merging to obtain our final compact set. Table 2 and Table 3 present the prediction accuracy with the final compact set, and their visualization is in Fig. 3. Observe that the clustering-based merging is then better than NMS or greedy selection, which naively relies on the predicted likelihoods. Note that the error lowerbound of directly training is only slightly better than merging in the case of Wayformer

	Method	brier_FDE5 ↓	minADE5 ↓	minFDE5 ↓	miss rate5 ↓
Wayformer	Greedy	8.935	3.302	8.287	0.712
	with FT	6.353 (-29%)	2.419 (-27%)	5.702 (-31%)	0.651 (-8.6%)
	NMS	7.679	2.871	7.024	0.637
	with FT	5.748 (-25%)	2.223 (-23%)	5.093 (-27%)	0.605 (-5.0%)
MTR-e2e	Greedy	6.016	2.115	5.369	0.623
	with FT	4.793 (-20%)	1.648 (-22%)	4.144 (-23%)	0.551 (-12%)
	NMS	5.089	1.793	4.440	0.521
	with FT	4.400 (-14%)	1.526 (-15%)	3.752 (-15%)	0.500 (-4.0%)

Table 4: Accuracy with one-step EM finetuning on NuScenes. Finetuning improves the accuracy of greedy selection significantly, reflecting a more informative mode probability. EMP is excluded from this analysis since it does not output waypoint Gaussians.

and EMP, and merging for MTR-e2e even slightly outperforms directly training. The results reveal two important observations: First, methods selecting a final model only relying on one single initial mode (greedy selection and NMS) do not work well, revealing that the predicted mode probabilities are uninformative about the overall pattern of all 64 modes or unaware of relatedness among nearby modes. Second, the clustering method assigning cluster centroids to be the final modes performs relatively well, reflecting multiple elements (elements in the same cluster) segment the same single component of the true mixture distribution. In other words, if each element learns a single mixture component, the cluster centroids would be placed between components.

	Method	brier_FDE5 ↓	minADE5 ↓	minFDE5 ↓	miss rate5 ↓
Wayformer	Greedy	11.645	4.143	10.994	0.695
	with FT	11.084 (-4.8%)	3.804 (-8.2%)	10.370 (-5.7%)	0.757
	NMS	11.423	4.080	10.775	0.684
	with FT	10.968 (-4.0%)	3.771 (-7.6%)	10.255 (-4.8%)	0.753
MTR-e2e	Greedy	6.982	2.421	6.335	0.634
	with FT	6.387 (-8.5%)	2.209 (-8.8%)	5.719 (-9.7%)	0.625
	NMS	6.498	2.273	5.858	0.591
	with FT	6.137 (-5.6%)	2.137 (-6.0%)	5.475 (-6.5%)	0.603

Table 5: Accuracy with one-step EM finetuning on WOMD.

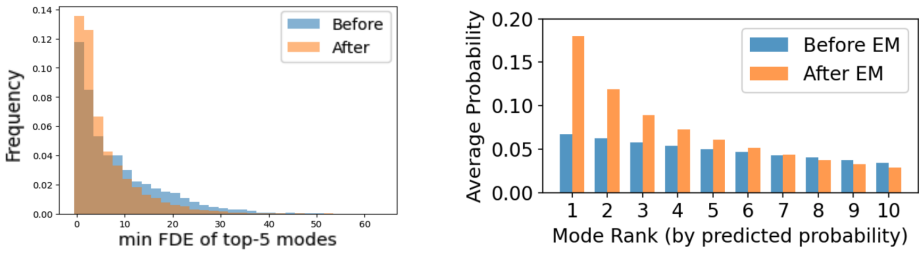
6.2 Evaluation of One-Step Expectation-Maximization

We also validate our post-hoc likelihood correction via the one-step EM method, and compare it to the original predicted likelihoods. Accuracy of one-step EM

	top-5 before ↓	top-5 after ↓	top-1 before ↓	top-1 after ↓
Wayformer	4404.5	1796.4 (-59%)	7711.0	5089.5 (-34%)
MTR-e2e	518.8	354.1 (-32%)	792.9	777.0 (-2.0%)

Table 6: Average NLL before and after one-step EM finetuning on NuScenes. Top-5 NLL is the neg-log-likelihood of the GMM composed by top-5 modes. Top-1 NLL is that of the top-1 Gaussian.

is compared against the original model with sampling-based post-hoc selection (greedy selection and NMS) in Table 4 and Table 5. Beyond displacement metrics, negative-log-likelihood of the ground truth is reported in Table 6. Note that the *mode trajectories predicted are kept consistent, and only the mode likelihoods are changed*, leading to improved selection of the final compact trajectory set.



(a) (b)

Fig. 4: (a) minFDE distribution before and after 1-step EM finetuning. Although the locations of the 64 trajectories have not been changed, 1-step EM finetunes the score for trajectory selection, leading to a better final set that improves minFDE. **(b) Probability concentration of top-10 predictions.** The likelihoods are more concentrated in candidates with correct future behavior after finetuning. Both results are obtained on NuScenes.

We observe that correction via the finetuning step enables the greedy and NMS selection to improve significantly, in terms of both improved displacement accuracy and distribution quality. To better understand this effect, we visualize the minFDE distribution before and after one-step EM finetuning in Fig. 4a. As compared to before finetuning, the likelihoods shift towards more accurate predictions, enabling better final trajectory selection. Further, Fig. 4b exhibits that the likelihoods are more concentrated in candidates with correct future behavior after finetuning. These results indicate that the 1-step EM brings spatial awareness to the likelihoods, highlighting spatially representative trajectories to mitigate the uninformativeness of likelihoods caused by over-segmentation.

7 Discussion

Why do we use the WTA objective? The WTA objective avoids severe mode collapse caused by full likelihood training, allowing different modes to specialize in different future behaviors, which aligns naturally with benchmark metrics like minADE/minFDE. Besides, the WTA objective also avoids computationally heavy EM processes. However, training with WTA objective sacrifices at predicting uninformative mode probabilities.

Implications for Practitioners. If the desired number of modes is fixed and training is feasible, training a model to directly produce that number of modes is the most effective strategy, as it provides the cleanest supervision signal, avoids test-time selection, and leads to higher accuracy. If flexibility in the number of modes is needed at test time and accuracy is prioritized over inference speed, we suggest using the clustering-based merging method described in Section 5.1, though note that clustering and merging will slow down inference. Finally, if superior inference speed is desired, applying one-step EM finetuning described in Section 5.2 on the training data, followed by non-maximum suppression or greedy selection, would help maintain accuracy.

8 Conclusion

In this work, we analyze the widely adopted winner-take-all (WTA) loss and provide a perspective linking Gaussian mixture models (GMM) to K-means clustering objective for trajectory forecasting of autonomous driving. We observe that WTA loss leads to uninformative mode probabilities and the modes over-segment the true multi-modal distribution instead of collapsing. We further offer principled and practical post-hoc treatments to align the training objective with inference-time probabilistic modeling assumptions and mitigate probability uninformativeness: test-time merging and one-step EM finetuning. Our findings and proposed treatments are empirically verified with three representative winner-take-all models on NuScenes and WOMD.

Acknowledgments

This research is supported in part by grants from the National Science Foundation (IIS-2107077, IIS-2107161, CNS-2211599, IIS-2305532). We thank the self-driving collaboration group for insightful discussions culminating in this work.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016)

2. Alahi, A., Ramanathan, V., Fei-Fei, L.: Socially-aware large-scale crowd forecasting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2203–2210 (2014)
3. Baniodeh, M., Goel, K., Ettinger, S., Fuertes, C., Seff, A., Shen, T., Gulino, C., Yang, C., Jerfel, G., Choe, D., et al.: Scaling laws of motion forecasting and planning–a technical report. arXiv preprint arXiv:2506.08228 (2025)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
6. Cao, C., Chen, X., Wang, J., Song, Q., Tan, R., Li, Y.H.: Cctr: calibrating trajectory prediction for uncertainty-aware motion planning in autonomous driving. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 20949–20957 (2024)
7. Casas, S., Gulino, C., Liao, R., Urtasun, R.: Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 9491–9497. IEEE (2020)
8. Casas, S., Gulino, C., Suo, S., Luo, K., Liao, R., Urtasun, R.: Implicit latent variable model for scene-consistent motion forecasting. In: European Conference on Computer Vision. pp. 624–641. Springer (2020)
9. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint arXiv:1910.05449 (2019)
10. Cui, A., Casas, S., Sadat, A., Liao, R., Urtasun, R.: Lookout: Diverse multi-future prediction and planning for self-driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16107–16116 (2021)
11. Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 1468–1476 (2018)
12. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., et al.: Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9710–9719 (2021)
13. Feng, L., Bahari, M., Amor, K.M.B., Zablocki, É., Cord, M., Alahi, A.: Unitraj: A unified framework for scalable vehicle trajectory prediction. In: European Conference on Computer Vision. pp. 106–123. Springer (2024)
14. Gao, K., Li, X., Chen, B., Hu, L., Liu, J., Du, R., Li, Y.: Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment. IEEE Transactions on Intelligent Transportation Systems **24**(6), 6203–6216 (2023)
15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2255–2264 (2018)
16. Hu, Y., Zhan, W., Tomizuka, M.: Probabilistic prediction of vehicle semantic intention and motion. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 307–313. IEEE (2018)

17. Jain, A., Casas, S., Liao, R., Xiong, Y., Feng, S., Segal, S., Urtasun, R.: Discrete residual flow for probabilistic pedestrian behavior prediction. In: Conference on Robot Learning. pp. 407–419. PMLR (2020)
18. Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., Fidler, S.: Situation recognition with graph neural networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4173–4182 (2017)
19. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489
20. Luber, M., Stork, J.A., Tipaldi, G.D., Arras, K.O.: People tracking with human motion predictions from social forces. 2010 IEEE International Conference on Robotics and Automation pp. 464–469 (2010), https://api.semanticscholar.org/CorpusID:1046089
21. Luo, K., Casas, S., Liao, R., Yan, X., Xiong, Y., Zeng, W., Urtasun, R.: Safety-oriented pedestrian occupancy forecasting. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1015–1022. IEEE (2021)
22. Luo, W., Park, C., Cornman, A., Sapp, B., Anguelov, D.: Jfp: Joint future prediction with interactive multi-agent modeling for autonomous driving. In: Conference on Robot Learning. pp. 1457–1467. PMLR (2023)
23. Nayakanti, N., Al-Rfou, R., Zhou, A., Goel, K., Refaat, K.S., Sapp, B.: Wayformer: Motion forecasting via simple & efficient attention networks. arXiv preprint arXiv:2207.05844 (2022)
24. Park, S.H., Kim, B., Kang, C.M., Chung, C.C., Choi, J.W.: Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In: 2018 IEEE intelligent vehicles symposium (IV). pp. 1672–1678. IEEE (2018)
25. Prutsch, A., Bischof, H., Possegger, H.: Efficient motion prediction: A lightweight & accurate trajectory prediction model with fast training and inference speed. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 9411–9417. IEEE (2024)
26. Rhinehart, N., Kitani, K.M., Vernaza, P.: R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 772–788 (2018)
27. Rhinehart, N., McAllister, R., Kitani, K., Levine, S.: Precog: Prediction conditioned on goals in visual multi-agent settings. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2821–2830 (2019)
28. Roh, J., Mavrogiannis, C., Madan, R., Fox, D., Srinivasa, S.: Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections. In: Conference on Robot Learning. pp. 2216–2227. PMLR (2021)
29. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1349–1358 (2019)
30. Shi, S., Jiang, L., Dai, D., Schiele, B.: Motion transformer with global intention localization and local movement refinement. Advances in Neural Information Processing Systems (2022)
31. Shi, S., Jiang, L., Dai, D., Schiele, B.: Mtr-a: 1st place solution for 2022 waymo open dataset challenge–motion prediction. arXiv preprint arXiv:2209.10033 (2022)
32. Shi, S., Jiang, L., Dai, D., Schiele, B.: Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. arXiv preprint arXiv:2306.17770 (2023)

33. Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K.S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C.P., Anguelov, D., et al.: Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 7814–7821. IEEE (2022)
34. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023)
35. Zhou, Z., Wang, J., Li, Y.H., Huang, Y.K.: Query-centric trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
36. Zyner, A., Worrall, S., Ward, J., Nebot, E.: Long short term memory for driver intent prediction. In: 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 1484–1489. IEEE (2017)