TCMA: TRAINING-FREE CAUSAL ONLINE MEMORY AUGMENTATION FOR BLACK-BOX PEDESTRIAN TRAJECTORY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

A deployed trajectory predictor accumulates information unavailable during training, namely the realized futures of its earlier predictions in the same environment. Existing approaches for exploiting such delayed feedback typically update the base model or train auxiliary memory modules, which may require internal access, gradients, or additional training and can be difficult in black-box or resource-constrained deployment. We therefore study whether completed trajectories can improve the quality of a frozen predictor's candidate set under a fixed output budget. Our key insight is that only trajectories whose futures have fully elapsed are causally admissible as memory. We introduce Training-free Causal Memory Augmentation (TCMA), an output-level adapter that retrieves such trajectories and replaces candidate slots with low historical win rates. To capture cross-scene motion patterns and within-recording recurrence, TCMA combines a scene-invariant egocentric channel with a scene-aware world-frame channel, both tuned online from elapsed labels. We analyze candidate replacement by decomposing its effect into retrieval benefit and displacement harm, and provide finite-sample risk bounds under explicit sampling and selection assumptions. On ETH-UCY, our prospective evaluation across seven predictors reports mean relative reductions of 6.2% in minADE@20 and 8.5% in minFDE@20. These results support output-level adaptation as a promising direction for reusing deployment experience, turning delayed outcomes into causally safe inference-time memory for deployed black-box predictors.

1 INTRODUCTION

Pedestrian trajectory prediction supports motion planning in autonomous vehicles, social robots, and intelligent transportation systems Huang et al. (2022); Mavrogiannis et al. (2023); Kothari et al. (2022); Korbmacher & Tordeux (2022). Recent predictors employ recurrent networks, graph-based interaction models, Transformers, and low-rank representations to generate multiple plausible futures Kothari et al. (2022); Korbmacher & Tordeux (2022).

Once deployed, however, these models are typically fixed. Yet deployment introduces another source of information: trajectories continue to unfold after predictions have been issued. Once their outcomes become observable, earlier predictions yield history-future pairs from the environment in which the system currently operates. Completed trajectories therefore provide direct supervisory signals at no additional annotation cost Park et al. (2024).

Existing approaches exploit completed trajectories primarily through memory-based retrieval or online adaptation. Memory-based methods retrieve stored trajectories using learned components, requiring a dedicated retrieval mechanism Marchetti et al. (2020; 2023); Xu et al. (2022a); Marchetti et al. (2024); Yang et al. (2022). Online adaptation methods update model parameters using delayed labels, requiring access to the predictor and test-time backpropagation Wang et al. (2021); Liang et al. (2024). Predictor-agnostic approaches similarly rely on separately trained adaptation modules Huynh & Alaghband (2023); Thakkar et al. (2024); Ivanovic et al. (2023). Moreover, existing memory-based methods do not explicitly define the temporal conditions under which a completed trajectory can be safely used to answer a query. Consequently, existing methods do not simulta-

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.

neously support training-free operation, black-box compatibility, and the causal use of trajectories collected during deployment.

These limitations raise a question about where adaptation should occur. A deployed predictor may be accessible only through its candidate outputs, and changing its weights may be impractical. Yet its output set can still be revised. A completed trajectory may contain a plausible future that the current candidate set misses. The challenge is to use such evidence without using future information or simply increasing the number of guesses.

To address this gap, we introduce Training-free Causal Memory Augmentation (TCMA), an inference-time, output-level adapter for any predictor that produces K candidate trajectories. TCMA stores completed trajectories, retrieves query-relevant candidates, and replaces low-value candidate slots under a fixed output budget, adding coverage without increasing the number of guesses or displacing high-quality candidates. Retrieval is causal: a trajectory becomes eligible only after its full future horizon elapses, preventing access to its own or another agent's unfinished future. TCMA uses two complementary retrieval channels: an ego-centric channel for cross-scene motion patterns and a world-frame channel for recurring paths within the current recording. The latter uses an online temporal fit-and-check procedure to select retrieval and gating parameters from elapsed deployment labels, eliminating manual tuning, test-time backpropagation, and a held-out validation set.

A key question is when replacing a candidate slot helps. Replacement can add a useful trajectory that the base predictor missed, but it can also remove a candidate that would have been correct. We analyze this tradeoff theoretically by establishing a distribution-free condition under which replacement cannot increase the minimum prediction error, and we bound the probability of violating this condition when win rates are estimated from finite data. We further decompose the error change into retrieval benefit and displacement harm. This decomposition applies to any metric that takes the minimum over candidates, such as minADE and minFDE. Experiments on ETH-UCY confirm that output-level adaptation improves evaluated predictors, reducing minADE@20 by 6.2% and minFDE@20 by 8.5% on average, with sub-millisecond retrieval and injection latency.

Our contributions are:

1. TCMA, a training-free, output-level memory adapter for frozen K-candidate predictors under a fixed output budget, which uses only trajectories whose futures have fully elapsed and requires no learned adapter parameters.
2. Two complementary retrieval protocols and a causal self-tuning procedure that estimates retrieval and gating parameters from elapsed labels. We further analyze candidate replacement by separating retrieval benefit from displacement harm, and provide conditional non-degradation and finite-sample guarantees under explicit assumptions.
3. A prospective evaluation across seven heterogeneous predictors on ETH-UCY, showing average reductions of 6.2% in minADE@20 and 8.5% in minFDE@20 with sub-millisecond per-query augmentation, while exposing the importance of activation coverage, temporal accounting, and adaptation cost.

2 Related Work

Pedestrian trajectory prediction. Early methods relied on hand-crafted kinematic and interaction models Helbing & Molnár (1995); Schöller et al. (2020); Korbmacher & Tordeux (2022). Later learning-based methods model social interactions, multimodal futures, and scene constraints using recurrent, adversarial, graph, Transformer, diffusion, and low-rank formulations Kothari et al. (2022); Gu et al. (2022). Representative methods include Social-LSTM Alahi et al. (2016), Social-GAN Gupta et al. (2018), Trajectron++ Salzmann et al. (2020), Graph-TERN Bae & Jeon (2023), TUTR Shi et al. (2023), EigenTrajectory Bae et al. (2023), and SingularTrajectory Bae et al. (2024b). These methods improve candidate generation, but do not exploit deployment-completed trajectories without retraining, except for recent adaptation methods Thakkar et al. (2024); Ivanovic et al. (2023).

Memory-based prediction. MANTRA Marchetti et al. (2020; 2023) uses learned controllers for associative trajectory memory, while MemoNet Xu et al. (2022a) retrieves trajectories via a learned addresser. Both show that stored trajectories improve candidate generation Marchetti et al. (2024); Yang et al. (2022), but their retrieval components require training and lack an explicit temporal

condition for future availability. TCMA instead uses explicit descriptors and output replacement, requiring neither a memory-aware predictor nor an auxiliary retrieval network.

LLM-based trajectory prediction. Large language models (LLMs) have also been explored for trajectory prediction. TrajEvo Zhao et al. (2026) uses LLM-driven evolution to generate heuristics Romera-Paredes et al. (2024); Ye et al. (2024), while SceneAware Bai & Shim (2025), LG-Traj Chib & Singh (2025), and LMTrajectory Bae et al. (2024a; 2026) use language models for scene constraints or trajectory generation. These require a language model to construct or execute the predictor; TCMA requires none.

Online adaptation. Online adaptation leverages post-deployment observations to improve subsequent predictions. T4P Park et al. (2024) updates representations and memory using delayed ground truth, requiring model access and test-time optimization. OAT-Mem and AOL Huynh & Alaghband (2023; 2020) use separately trained adaptation modules, whereas others rely on learned prompts Thakkar et al. (2024), adaptive layers Ivanovic et al. (2023), meta-learned initializations Yang et al. (2024), or cross-domain transfer Xu et al. (2022b). TrajICL Fujii et al. (2025) performs weight-free in-context adaptation using same-scene examples. TCMA is predictor-agnostic and instead adapts via retrieval, gating, and candidate replacement, without gradient updates or trainable auxiliary components Wang et al. (2021); Liang et al. (2024).

Overall, prior work provides learned trajectory memories, model-based online adaptation, or predictor-agnostic adaptation via trained auxiliary modules Thakkar et al. (2024); Fujii et al. (2025). TCMA uniquely supports black-box compatibility with training-free retrieval and determines the admissibility of each memory entry using an explicit and verifiable causal condition.

3 METHODOLOGY

TCMA is an output-level adapter that improves a frozen K-candidate predictor without training or internal model access. It separates two questions: when a memory entry becomes causally available, and whether replacing a candidate with it is beneficial. Given an observed eight-frame history X_i^{obs}, a black-box predictor π produces K candidate futures:

$$P_i = \left(p_i^1, \ldots, p_i^K\right), \tag{1}$$

where K denotes the number of candidate trajectories produced by the predictor. For a candidate trajectory p and ground-truth future x, the average displacement error is

$$\mathrm{ADE}(p, x) = \frac{1}{12} \sum_{\tau=1}^{12} \left\| p_\tau - x_\tau \right\|_2, \tag{2}$$

and the corresponding multimodal error is

$$\mathrm{minADE}\,@K(P, x) = \min_{k \in \{1,\ldots,K\}} \mathrm{ADE}(p^k, x). \tag{3}$$

We define the winning candidate as

$$W(P, x) = \arg \min_{k \in \{1,\ldots,K\}} \mathrm{ADE}(p^k, x), \tag{4}$$

with ties resolved in favor of the lowest index.

TCMA accesses only predicted trajectories and requires no gradients, internal features, architectural modifications, or learned adapters; the base predictor remains frozen. Its memory and configuration processes are separated into three stages: offline initialization builds the memory bank, online memory adaptation appends trajectories as their futures become available, and online configuration adaptation re-selects retrieval and gating settings using elapsed labels. Thus, memory adaptation determines available evidence, while configuration adaptation determines how it is used. The complete pipeline is shown in Fig. 1.

3.1 CAUSAL ONLINE MEMORY AND RETRIEVAL

TCMA admits a trajectory to memory only after its full future horizon has elapsed, preventing queries from accessing their own future or another agent's unfinished future. Unlike prior memory-

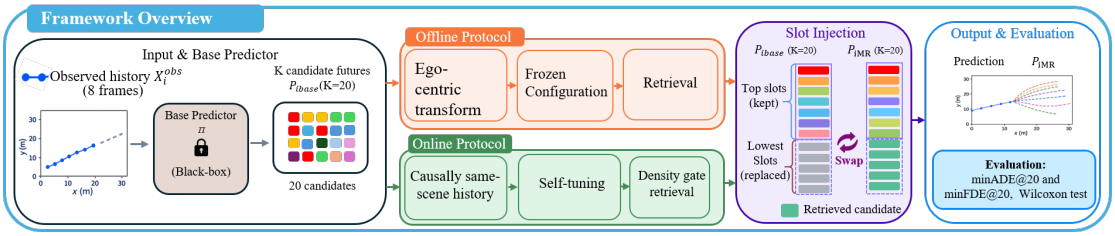

Figure 1: Overview of the TCMA pipeline. A frozen black-box predictor generates K=20 candidate trajectories; offline and online retrieval protocols supply replacement candidates that are injected into the lowest-ranked slots, producing the augmented prediction evaluated via minADE@20, minFDE@20, and the Wilcoxon signed-rank test.

based predictors without explicit temporal constraints, TCMA enforces causal availability by construction. Each memory entry stores a motion descriptor, a realized future, and the completion time:

$$e_j = \left(\phi_j, \widetilde{X}_j^{\text{fut}}, t_j^{\text{end}}\right). \tag{5}$$

At frame t, only entries whose complete future horizons have elapsed are eligible for retrieval:

$$\mathcal{M}_t = \left\{e_j \mid t_j^{\text{end}} \le t\right\}. \tag{6}$$

The memory is updated as trajectories are completed:

$$\mathcal{M}_{t+1} = \mathcal{M}_t \cup \mathcal{E}_t, \qquad \mathcal{E}_t = \left\{e_j \mid t_j^{\text{end}} = t+1\right\}. \tag{7}$$

Equation 6 prevents a query from accessing either its own future or the unfinished future of another agent. An offline memory bank can provide cold-start retrieval before a sufficient number of online entries becomes available.

Ego-centric retrieval. To reduce sensitivity to absolute position and orientation, each trajectory history is transformed into the agent's local coordinate system as illustrated in Fig. 2 (A).

$$T_i(X) = R(-\theta_i)(X - c_i), \tag{8}$$

where c_i denotes the final observed position and θ_i denotes the heading of the final observed displacement. The transformed history is summarized by the motion descriptor

$$\phi_i = \left[v_{\text{last}}, \bar{v}, \sigma_v, a_{\text{last}}, \bar{a}, \kappa\right], \tag{9}$$

which contains the final speed, mean speed, speed variation, final acceleration, mean acceleration, and path curvature. Each component is standardized using statistics computed from the training split. TCMA retrieves the nearest eligible entry in the resulting descriptor space and maps its future into the query coordinate system using T_i^{-1}.

Scene-aware world-frame retrieval. The second retrieval channel exploits absolute spatial information within the current recording. For a query i and memory entry j, their distance is defined as

$$D(i,j) = \frac{\|p_i - p_j\|_2}{\bar{p}} + \gamma\left(1 - \cos\angle(h_i, h_j)\right) + \lambda\frac{|v_i - v_j|}{\bar{v}}, \tag{10}$$

where p, h, and v denote position, heading, and speed, respectively. The normalization scales $\bar{p}$ and $\bar{v}$ are estimated without using the held-out scene. Retrieval is controlled by the density gate

$$\rho(i) = \mathbf{1}\left[\left|\left\{j : s_j = s_i,\ t_j^{\text{end}} < t,\ D(i,j) \le r\right\}\right| \ge \nu\right]. \tag{11}$$

When the gate is active, TCMA uses either the nearest neighbors or their recency-weighted consensus. Otherwise, it retains the base prediction. The online world-frame protocol retrieves only completed trajectories from the same recording and abstains during the cold-start period. Ego-centric and world-frame retrieval are evaluated as separate protocols: the former transfers motion patterns across locations, whereas the latter exploits spatial regularities within a scene.

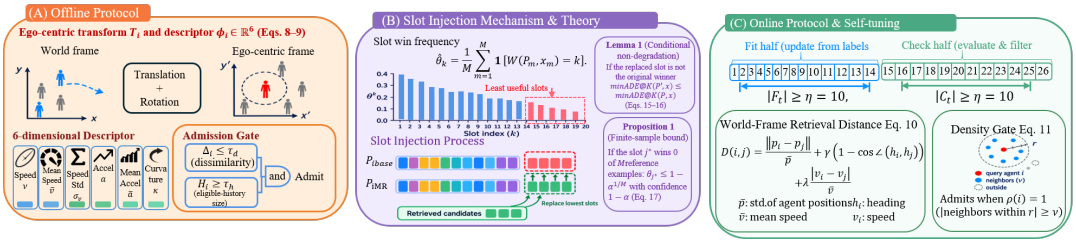

Figure 2: Detailed view of the TCMA components. (A) Offline ego-centric protocol: coordinate transform, 6-D motion descriptor, and admission gate. (B) Slot injection mechanism: per-slot win frequencies identify the least useful candidates, which are replaced under Lemma 1 (non-degradation) and Proposition 1 (finite-sample confidence bound). (C) Online protocol: causal fit/check label split for self-tuning, world-frame retrieval distance, and density-based admission gate.

3.2 Candidate Slot Injection

TCMA injects retrieved trajectories by replacing candidate slots that rarely attain the minimum prediction error (Fig. 2 (B)), following a per-index winning statistic also used in Zhao et al. (2026). Given M reference examples, the empirical winning frequency of slot k is

$$\hat{\theta}_k = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\left[W(P_m, x_m) = k\right]. \tag{12}$$

Offline frequencies are estimated using non-test reference predictions. Online frequencies are computed exclusively from labels that were observed before the selected configuration was activated. TCMA selects the q slots with the lowest empirical winning frequencies:

$$S = \arg \min_{S' \subseteq \{1,\dots,K\} | S'| = q} \sum_{k \in S'} \hat{\theta}_k. \tag{13}$$

Given retrieved candidates $(a_1, \dots, a_q)$, the augmented output is defined as

$$p'_k = \begin{cases} a_{\mathrm{rank}_S(k)}, & k \in S, \\ p_k, & k \notin S, \end{cases} \tag{14}$$

where $\mathrm{rank}_S(k)$ denotes the position of slot k within the ordered set S. All remaining candidates are left unchanged. Because candidate ordering depends on the construction of the predictor's output, slot-winning frequencies are estimated separately for each predictor.

3.3 Online Self-Tuning and Theoretical Analysis

Conditional non-degradation. The following result characterizes when candidate replacement preserves the original minimum prediction error.

Lemma 1 (Conditional Non-Degradation). Let P' be obtained from P by replacing slot j^* with a candidate a. If

$$W(P, x) \neq j^*, \tag{15}$$

then

$$\mathrm{minADE}\,@K(P', x) \le \mathrm{minADE}\,@K(P, x). \tag{16}$$

Because the original winning candidate remains in P', the minimum error over P' cannot exceed the error attained by that candidate. The condition cannot be verified for an individual query because its future is unknown at prediction time. Instead, its failure probability can be estimated from reference data.

Proposition 1. If slot j^* wins zero times among M independent reference examples, the one-sided Clopper–Pearson upper bound at confidence level $1 - \alpha$ is

$$\theta_{j^*} \le 1 - \alpha^{1/M}. \tag{17}$$

This expression bounds the marginal winning probability under the reference distribution and does not provide an instance-level guarantee Gibbs & Candès (2021); Lindemann et al. (2023). The complete derivation and confidence calculations are presented in Appendix C.

Causal online self-tuning. Let $\mathcal{L}_t$ contain the trajectories completed by frame t, and let t_0 denote their median completion time. The elapsed labels are divided chronologically into a fitting set and a checking set (Fig. 2 (C)):

$$\mathcal{F}_t = \left\{ e_j \in \mathcal{L}_t \mid t_j^{\text{end}} < t_0 \right\}, \qquad \mathcal{C}_t = \left\{ e_j \in \mathcal{L}_t \mid t_j^{\text{end}} \geq t_0 \right\}. \tag{18}$$

Configuration selection begins only when both sets contain at least $\eta = 10$ examples. Each configuration c is evaluated on both sets, and the selected configuration is

$$c^* = \arg\max_c \left[\text{gain}(c, \mathcal{F}_t) + \text{gain}(c, \mathcal{C}_t) \right], \tag{19}$$

subject to

$$\text{gain}(c, \mathcal{F}_t) > 0, \qquad \text{gain}(c, \mathcal{C}_t) > 0. \tag{20}$$

The selected configuration is applied only to subsequent queries:

$$c_\tau = c^*, \qquad \tau \geq t. \tag{21}$$

If either set is too small or no configuration improves performance on both sets, the current configuration remains active. If no configuration has yet been accepted, TCMA abstains and retains the base prediction. Self-tuning selects among fixed retrieval, gating, and injection settings; it does not update any learned weights Park et al. (2024); Wang et al. (2021).

3.4 Complete Pipeline and Complexity

For each query, the base predictor produces

$$P_i^{\text{base}} = \pi \left(X_i^{\text{obs}} \right). \tag{22}$$

TCMA searches the eligible memory and applies Eq. 14 when the retrieval gate is active:

$$P_i^{\text{MR}} = \begin{cases} I \left(P_i^{\text{base}}, \mathcal{N}_i \right), & g_i = 1, \\ P_i^{\text{base}}, & g_i = 0, \end{cases} \tag{23}$$

where $\mathcal{N}_i$ denotes the retrieved candidate set and I denotes the slot-injection operation. Memory and configuration updates affect only subsequent queries. For a descriptor dimension of $d = 6$, brute-force ego-centric retrieval has a computational complexity of $O(|\mathcal{M}_t|d)$, whereas same-scene world-frame retrieval requires $O(|\mathcal{M}_t^{\text{scene}}|)$. Memory construction is linear in the number of stored trajectories. Configuration search is performed only at scheduled tuning boundaries; Section 5 reports the measured runtime.

4 Experiments

4.1 Experimental Setup

We evaluate TCMA on the standard ETH-UCY benchmark Pellegrini et al. (2009); Lerner et al. (2007), comprising five scenes (ETH, HOTEL, UNIV, ZARA1, ZARA2) with 1,536 trajectories under diverse pedestrian flows. Following the leave-one-scene-out protocol, we train and build offline memory on four scenes and evaluate on the fifth. Predictions use 8 observed frames (3.2 s) to forecast 12 future frames (4.8 s). All adaptation is causal: trajectories enter memory only after their ground-truth futures have fully elapsed, retrieval configurations are selected solely from pre-boundary labels, and world-frame normalization uses only the four non-held-out scenes.

For the online protocol, we fix the evaluation boundary at $t_{\text{eval}} = t_{\min} + 0.4(t_{\max} - t_{\min})$ for every scene and predictor. Trajectories whose futures elapse strictly before t_{eval} may support memory construction and configuration selection but are excluded from scoring. Agents observed at or after t_{eval} form the prospective evaluation phase and are scored under identical base and augmented conditions. This ensures a strict separation between the warm-up period used for memory and configuration selection and the prospective evaluation period.

Table 1: Per-scene minADE@20 / minFDE@20 gains (%) under the online self-tuning protocol. "–" denotes abstention: no configuration was accepted for the scene.

Predictor	ETH	HOTEL	UNIV	ZARA1	ZARA2
CVM-S Schöller et al. (2020)	–	+40.5/+47.6	+17.3/+24.4	+17.6/+27.7	+13.2/+19.8
TrajEvo heuristic Zhao et al. (2026)	–	+8.6/+12.8	+11.6/+16.5	+14.8/+22.5	+17.3/+24.7
Ours (standalone)	–	+6.1/+7.8	+4.1/+6.3	+9.0/+14.1	+7.3/+10.7
Graph-TERN Bae & Jeon (2023)	–	+12.3/+15.5	+1.5/+2.0	–	+8.3/+9.6
MemoNet Xu et al. (2022a)	–	+3.5/+6.5	+2.4/+3.9	-0.8/-0.5	+4.4/+5.2
TUTR Shi et al. (2023)	–	+3.2/+3.1	+2.2/+2.0	–	+4.0/+3.6
EigenTrajectory Bae et al. (2023)	–	+3.7/+3.6	+1.6/+2.6	–	+3.5/+4.7

We evaluate seven predictors spanning diverse paradigms: CVM-S Schöller et al. (2020), TrajEvo Zhao et al. (2026), our standalone heuristic predictor, Graph-TERN Bae & Jeon (2023), MemoNet Xu et al. (2022a), TUTR Shi et al. (2023), and EigenTrajectory Bae et al. (2023). These cover heuristic, control-point, memory-based, Transformer-based, and low-rank approaches. TCMA neither modifies nor retrains any predictor, using only its candidate outputs for training-free, post-hoc augmentation. We follow the standard evaluation protocol, which uses K=20 candidate trajectories per query. This setting is used by prior work for fair comparison and reflects a practical balance between multimodal coverage and downstream planning constraints.

We report minimum average and final displacement errors over 20 candidates (minADE@20 and minFDE@20), standard best-of-K metrics that measure fixed-size candidate-set quality. Scene-level results average over agents, followed by equal weighting across scenes. Since TUTR Shi et al. (2023) and MemoNet Xu et al. (2022a) use different ETH populations from the shared loader, direct predictor comparisons use population-matched four-scene averages. TCMA abstains on ETH for all predictors, so this difference does not affect the headline averages. Complete population accounting is provided in Appendix H.3, and absolute baseline values are provided in Appendix I.

Stochastic predictors are evaluated with three independent seeds and averaged across seeds, while TCMA is deterministic given the exported candidate predictions. Statistical significance is assessed using paired one-sided Wilcoxon signed-rank tests on per-agent minADE@20. We also report the fraction of agents for which augmentation strictly reduces error. Exact statistics and Clopper–Pearson confidence intervals are provided in Appendix M.

4.2 Main Results under the Causal Online Protocol

We first evaluate the primary online self-tuning protocol, in which TCMA uses only labels already available in the held-out scene. Configuration selection is prospective, and previous predictions are never modified retroactively. The headline gain for each predictor is the equally weighted mean across all five scenes. Abstained scenes contribute zero gain because TCMA retains the base prediction, which prevents degradation when causal evidence is insufficient.

Fig. 3 summarizes the results across all seven predictors. TCMA improves mean minADE@20 by 6.20% and minFDE@20 by 8.47%. Positive gains extend across heuristic, control-point, memory-based, Transformer-based, and low-rank predictors, demonstrating compatibility with diverse architectures. Gains are generally larger for weaker predictors, suggesting that retrieval contributes more when the original candidate set provides limited coverage; however, this relationship is descriptive rather than formally monotonic.

To determine whether the aggregate gains are concentrated in a small subset of scenes, Table 1 reports the scene-level results. Among the 25 predictor–scene combinations in which TCMA activates, 24 show significant improvements in minADE@20 ($p < 0.001$). The only negative case is MemoNet Xu et al. (2022a) on ZARA1, where minADE@20 and minFDE@20 decrease by approximately 0.8% and 0.5%, respectively; neither change is statistically significant. TCMA abstains when its activation requirements are not satisfied, including for all predictors on ETH and several on ZARA1, thereby preserving the original predictor when sufficient causal evidence is unavailable.

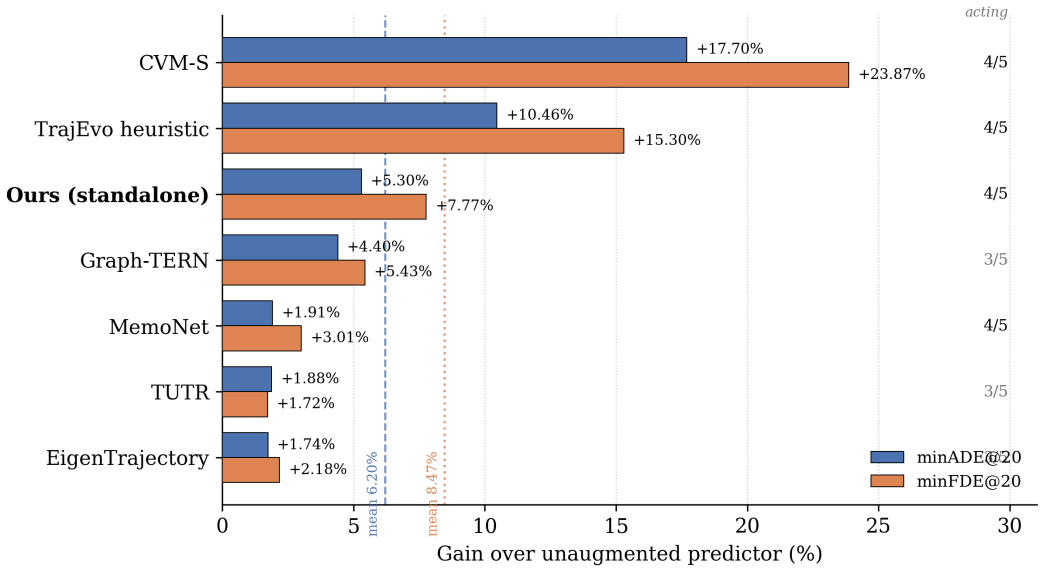

Figure 3: Online self-tuning gains across seven heterogeneous trajectory predictors. Gains are relative to the corresponding unaugmented predictor. "Acting" denotes the number of scenes in which the online protocol activates.

Table 2: Key TCMA ablations. The memory and descriptor blocks use the standalone predictor and report average minADE@20. The gate block uses TrajEvo and reports average relative gain (%). The additional static-versus-causal comparison on CVM-S is discussed in the text. Values are comparable only within each block.

Component	Variant	Result
Standalone Memory	No memory	0.277
Standalone Memory	Static memory	0.256
Standalone Memory	Static + causal memory	**0.248**
Gate	Percentile gate	+0.1%
Gate	Absolute distance/history gate	**+3.7%**
Descriptor	6D	0.2404
Descriptor	9D	0.2402

4.3 Ablations and Additional Analysis

We next examine the principal design choices underlying TCMA. Table 2 summarizes three key ablation groups: memory construction, retrieval gating, and descriptor dimensionality. Because these experiments use different predictors and evaluation protocols, comparisons should be made within each block rather than across blocks.

Causal memory. For the standalone predictor, average minADE@20 decreases from 0.277 without memory to 0.256 with a static bank and 0.248 with causally completed deployment trajectories. For CVM-S Schöller et al. (2020) under the offline protocol, static and causal memory improve the baseline by 24.9% and 28.3%, respectively, showing that causal memory adds useful evidence without accessing unobserved futures.

Retrieval gate. On TrajEvo, the percentile gate gives a 0.1% average gain versus 3.7% for the absolute distance/history gate. We therefore use the latter because its thresholds remain interpretable as the memory grows.

Descriptor design. The six-dimensional descriptor uses recent and mean speed, speed variation, recent and mean acceleration, and path curvature. Its average validation minADE differs from the

Table 3: Measured computational overhead of TCMA on the largest evaluated scene. Retrieval and injection are per-query costs, whereas configuration search is amortized over a configuration-selection boundary.

Operation	Cost
Ego-centric retrieval	717 μs/query
World-frame retrieval	891 μs/query
Slot injection	1.2 μs/query
Peak additional memory	5.3 MB
Configuration search	23.6 min/boundary

nine-dimensional alternative by less than 0.1 percentage points: it wins on two scenes, ties on one, and loses on two. We retain the smaller descriptor for compactness. Further retrieval comparisons, definitions, and ablations are provided in Appendices A, B, and K.

Offline versus online adaptation. We also evaluate an offline protocol that selects configurations on validation data before testing. Its minADE@20 gains range from 0.3% to 28.3% across seven predictors, with the largest gain on CVM-S Schöller et al. (2020). These results are not directly comparable to the primary online protocol, which selects configurations from elapsed labels in the held-out scene and may abstain when evidence is insufficient. Full results and frozen configurations are provided in Appendix J.

Computational Overhead. TCMA requires neither gradient computation nor modification of the underlying predictor. As shown in Table 3, retrieval and injection add less than 1,ms per query and 5.3,MB peak memory. Configuration search is performed once per scene at the first tuning boundary with at least 10 fitting and 10 checking examples and is excluded from per-query latency. The complete search grid is provided in Appendix D.

5 Discussion & Limitations

TCMA improves all seven predictors on average, with minADE@20 gains ranging from 1.7% for EigenTrajectory to 17.7% for CVM-S Schöller et al. (2020), while adding less than 1 ms per query. The larger gains for weaker baselines suggest that retrieval is most beneficial when the original candidate set leaves useful motion patterns uncovered.

TCMA conservatively retains the base output when there are insufficient completed trajectories. Among 25 active predictor-scene combinations, only MemoNet Xu et al. (2022a) on ZARA1 shows a small, non-significant decrease. Lemma 1 and finite-sample analysis identify when candidate replacement preserves performance and bound the marginal risk of replacing a winning slot.

Our evaluation focuses on ETH-UCY, a standard pedestrian trajectory benchmark. Extending TCMA to other domains, such as vehicle trajectory prediction and robot navigation, remains an important direction. The current descriptor and retrieval protocols are designed for pedestrian motion; whether they transfer directly to other agent types is an open question.

6 Conclusion

We introduced TCMA, a training-free framework that augments black-box trajectory predictors with deployment trajectories whose futures have fully elapsed. TCMA retrieves relevant trajectories, replaces candidate slots with low historical winning frequencies, and leaves the base output unchanged when causal evidence is insufficient. Across seven predictors on ETH-UCY, TCMA reduces average minADE@20 by 6.2% and minFDE@20 by 8.5%, with less than 1 ms of per-query retrieval and injection overhead. These results show that completed deployment trajectories can improve candidate coverage without retraining the predictor. More broadly, TCMA turns delayed outcomes into causally safe inference-time memory, offering a practical route to continual improvement of deployed black-box predictors. Future work will extend this framework to other prediction domains.

AI USE STATEMENT

We used generative AI tools solely to aid and polish the writing of this paper (improving clarity, grammar, and readability). No generative AI tools were used for research ideation, methodology design, experiments, data analysis, mathematical claims, proofs, or any other substantive research contributions. We have reviewed all AI-assisted text and take full responsibility for the final content of this work.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, 2016. doi: 10.1109/CVPR.2016.110.

Inhwan Bae and Hae-Gon Jeon. A set of control points conditioned pedestrian trajectory prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i5.25759.

Inhwan Bae, Jean Oh, and Hae-Gon Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9983–9995, 2023. doi: 10.1109/ICCV51070.2023.00919.

Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Can language beat numerical regression? language-based multimodal trajectory prediction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 753–766, 2024a. doi: 10.1109/CVPR52733.2024.00078.

Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. Singulartrajectory: Universal trajectory predictor using diffusion model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17890–17901, 2024b. doi: 10.1109/CVPR52733.2024.01694.

Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Social reasoning-aware trajectory prediction via multimodal language model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6): 6035–6052, 2026. doi: 10.1109/TPAMI.2025.3582000.

Juho Bai and Inwook Shim. SceneAware: Scene-constrained pedestrian trajectory prediction with LLM-guided walkability. June 2025.

Pranav Singh Chib and Pravendra Singh. Lg-traj: Llm guided pedestrian trajectory prediction. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 6861–6871, 2025. doi: 10.1109/ICCVW69036.2025.00709.

Ryo Fujii, Hideo Saito, and Ryo Hachiuma. Towards predicting any human trajectory in context. June 2025.

Isaac Gibbs and Emmanuel J. Candès. Adaptive conformal inference under distribution shift. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17092–17101, 2022.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018. doi: 10.1109/CVPR.2018.00240.

Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51: 4282–4286, May 1995. doi: 10.1103/PhysRevE.51.4282.

Yanjun Huang, Jiatong Du, Ziru Yang, Zewei Zhou, Lin Zhang, and Hong Chen. A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 7(3):652–674, 2022. doi: 10.1109/TIV.2022.3167103.

Manh Huynh and Gita Alaghband. AOL: Adaptive online learning for human trajectory prediction in dynamic video scenes. February 2020.

Manh Huynh and Gita Alaghband. Online adaptive temporal memory with certainty estimation for human trajectory prediction. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 940–949, 2023. doi: 10.1109/WACV56688.2023.00100.

Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7786–7793, 2023. doi: 10.1109/ICRA48891.2023.10161155.

Raphael Korbmacher and Antoine Tordeux. Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24126–24144, 2022. doi: 10.1109/TITS.2022.3205676.

Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *Trans. Intell. Transport. Sys.*, 23(7):7386–7400, July 2022. ISSN 1524-9050. doi: 10.1109/TITS.2021.3069362.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007. doi: https://doi.org/10.1111/j.1467-8659.2007.01089.x.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *Int. J. Comput. Vision*, 133(1):31–64, July 2024. ISSN 0920-5691. doi: 10.1007/s11263-024-02181-w.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8): 5116–5123, 2023. doi: 10.1109/LRA.2023.3292071.

Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7141–7150, 2020. doi: 10.1109/CVPR42600.2020.00717.

Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Multiple Trajectory Prediction of Moving Agents With Memory Augmented Networks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(06):6688–6702, June 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2020.3008558.

Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Smemo: Social memory for trajectory forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4410–4425, 2024. doi: 10.1109/TPAMI.2024.3356755.

Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. *J. Hum.-Robot Interact.*, 12(3), April 2023. doi: 10.1145/3583741.

Daehee Park, Jaeseok Jeong, Sung-Hoon Yoon, Jaewoo Jeong, and Kuk-Jin Yoon. T4p: Test-time training of trajectory prediction via masked autoencoder and actor-specific token memory. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15065–15076, 2024. doi: 10.1109/CVPR52733.2024.01427.

S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, 2009. doi: 10.1109/ICCV.2009.5459260.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco J R Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, January 2024.

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

Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020. doi: 10.1109/LRA.2020.2969925.

Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9641–9650, 2023. doi: 10.1109/ICCV51070.2023.00887.

Neerja Thakkar, Karttikeya Mangalam, Andrea Bajcsy, and Jitendra Malik. Adaptive human trajectory prediction vianbsp;latent corridors. pp. 297–314, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72919-5. doi: 10.1007/978-3-031-72920-1_17.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6478–6487, 2022a. doi: 10.1109/CVPR52688.2022.00638.

Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable gnn. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6510–6521, 2022b. doi: 10.1109/CVPR52688.2022.00641.

Biao Yang, Fucheng Fan, Rongrong Ni, Jie Li, Loochu Kiong, and Xiaofeng Liu. Continual learning-based trajectory prediction with memory augmented networks. *Know.-Based Syst.*, 258 (C), December 2022. ISSN 0950-7051. doi: 10.1016/j.knosys.2022.110022.

Biao Yang, Yanan Lu, Rui Wan, Hongyu Hu, Changchun Yang, and Rongrong Ni. Meta-irlsot++: A meta-inverse reinforcement learning method for fast adaptation of trajectory prediction networks. *Expert Syst. Appl.*, 240(C), April 2024. ISSN 0957-4174. doi: 10.1016/j.eswa.2023.122499.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. ReEvo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems 37*, pp. 43571–43608, San Diego, California, USA, 2024. Neural Information Processing Systems Foundation, Inc. (NeurIPS).

Zhikai Zhao, Chuanbo Hua, Federico Berto, Kanghoon Lee, Zihan Ma, Jiachen Li, and Jinkyoo Park. TrajEvo: Trajectory prediction heuristics design via LLM-driven evolution. *Proc. Conf. AAAI Artif. Intell.*, 40(21):18072–18080, March 2026.

A Causal Memory and Retrieval

This section provides additional mathematical details for the TCMA framework. The definitions below complement the formulation presented in the main paper.

We use the same prediction formulation and evaluation metrics defined in the main paper. For candidate replacement analysis, we additionally define the winning candidate index as follows

$$W(P, x) = \arg \min_{k \in \{1,\dots,K\}} \mathrm{ADE}(p^k, x), \tag{24}$$

where ties are resolved using the lowest candidate index.

This index is the object every guarantee in the theoretical analysis is stated about: the conditional non-degradation lemma and the finite-sample confidence bound both depend only on whether injection touches $\tilde{j} = W(P, x)$, not on any property of the injected trajectory. Because candidate ordering differs between predictors, slot-winning statistics are computed independently for each predictor.

A.1 Causal Memory Eligibility

This section defines the causal eligibility rule that governs when a memory entry can be used.

Each stored trajectory is represented as a memory entry:

$$e_j = \left(\phi_j, \widetilde{X}_j^{\mathrm{fut}}, t_j^{\mathrm{end}}\right), \tag{25}$$

where ϕ_j denotes the trajectory descriptor, $\widetilde{X}_j^{\mathrm{fut}}$ denotes the stored future trajectory, and t_j^{end} denotes the frame at which the future becomes completely observable.

At deployment frame t, TCMA admits only trajectories whose futures have fully elapsed:

$$\mathcal{M}_t = \left\{e_j \mid t_j^{\mathrm{end}} \le t\right\}. \tag{26}$$

This eligibility rule ensures that a query cannot access its own future or the unfinished future of another trajectory, an entry is admitted only once $t \ge t_j^{\mathrm{end}}$.

The memory update operation is:

$$\mathcal{M}_{t+1} = \mathcal{M}_t \cup \mathcal{E}_t, \tag{27}$$

where

$$\mathcal{E}_t = \left\{e_j \mid t_j^{\mathrm{end}} = t + 1\right\}. \tag{28}$$

Eqs. 27–28 only append newly-eligible entries at each frame, no entry is ever removed or revised, so $|\mathcal{M}_t|$ is non-decreasing over the course of deployment.

A.2 Ego-Centric Retrieval

To reduce sensitivity to absolute position and orientation, TCMA represents trajectories in an agent-centric coordinate system.

The transformation is:

$$T_i(X) = R(-\theta_i)(X - c_i), \tag{29}$$

where c_i is the final observed position and θ_i is the heading direction of the final observed displacement.

The selected descriptor is:

$$\phi_i = [v_{\mathrm{last}}, \bar{v}, \sigma_v, a_{\mathrm{last}}, \bar{a}, \kappa], \tag{30}$$

which contains:

- final speed,
- average speed,
- speed variation,
- final acceleration,
- average acceleration,

- path curvature.

Because ϕ_i is computed after the transform T_i, it is invariant to the agent's absolute position and heading, so two agents performing the same maneuver at different locations map to the same region of descriptor space; this is what allows retrieval to transfer across scenes rather than being confined to a single recording. Descriptor normalization uses fixed dataset statistics following the benchmark protocol, and TCMA optimizes no additional parameters during deployment.

For a retrieved trajectory stored in the local coordinate frame, the inverse transformation maps it back to the query coordinate system:

$$X_{j\to i}^{\mathrm{fut}} = T_i^{-1}(\widetilde{X}_j^{\mathrm{fut}}). \tag{31}$$

The inverse transform T_i^{-1} in Eq. 31 maps the retrieved future out of the memory entry's local frame and into the query's own frame, turning a match found anywhere in descriptor space into a candidate that is physically valid for the agent's actual position and heading.

A.3 World-Frame Retrieval

For the online world-frame protocol, TCMA retrieves completed trajectories from the current recording.

For query trajectory i and memory trajectory j, the retrieval distance is:

$$D(i,j) = \frac{\|p_i - p_j\|_2}{\bar{p}} + \gamma(1 - \cos\angle(h_i, h_j)) + \lambda\frac{|v_i - v_j|}{\bar{v}}, \tag{32}$$

where:

- p denotes position,
- h denotes heading,
- v denotes speed,
- $\bar{p}$ and $\bar{v}$ are normalization constants.

The distance $D(i,j)$ in Eq.32 combines spatial proximity, heading misalignment, and speed mismatch into a single score, so two agents are treated as neighbors only when they are close in position, direction, and pace simultaneously. Retrieval is controlled using the density gate:

$$\rho(i) = \mathbf{1}\left[\left|\left\{j : s_j = s_i, t_j^{\mathrm{end}} < t, D(i,j) \le r\right\}\right| \ge \nu\right]. \tag{33}$$

The gate $\rho(i)$ in Eq. 33 requires at least ν same-scene, already-completed neighbors within radius r before world-frame retrieval is trusted, when local evidence is too sparse, TCMA leaves the base predictor output unchanged rather than retrieving from an unreliable neighborhood.

B Candidate Slot Injection

TCMA replaces candidate slots that historically contribute least to the minimum prediction error.

Given M reference examples, the empirical winning frequency of candidate slot k is:

$$\hat{\theta}_k = \frac{1}{M}\sum_{m=1}^{M}\mathbf{1}[W(P_m, x_m) = k]. \tag{34}$$

$\hat{\theta}_k$ in Eq. 34 estimates, from reference data alone, how often slot k has ever been the arg-min-error candidate. A low $\hat{\theta}_k$ indicates that slot has historically contributed little to the predictor's reported minADE.

The selected replacement set is:

$$S = \arg \min_{\substack{S' \subseteq \{1,\dots,K\} \\ |S'|=q}} \sum_{k \in S'} \hat{\theta}_k. \tag{35}$$

Eq. 35 selects the q slots with the smallest cumulative win rate, i.e. the candidates least likely to have determined the predictor's error and therefore the safest to sacrifice under the condition established in Lemma 1. Given retrieved trajectories $(a_1, \dots, a_q)$, the augmented candidate set becomes:

$$p'_k = \begin{cases} a_{\mathrm{rank}_S(k)}, & k \in S, \\ p_k, & k \notin S. \end{cases} \tag{36}$$

Only selected low-winning-frequency slots are modified. All remaining candidates are preserved.

C Theoretical Analysis

C.1 Conditional Non-Degradation Lemma

Lemma 1 (Conditional Non-Degradation).

Let P' be obtained from P by replacing slot j^* with a retrieved candidate a. If:

$$W(P, x) \neq j^*, \tag{37}$$

then:

$$\mathrm{minADE}\,@K(P', x) \leq \mathrm{minADE}\,@K(P, x). \tag{38}$$

Proof. Let:

$$\tilde{j} = W(P, x) \tag{39}$$

The assumption gives:

$$\tilde{j} \neq j^* \tag{40}$$

Therefore, the original winning candidate remains unchanged after replacement:

$$p'_{\tilde{j}} = p_{\tilde{j}} \tag{41}$$

Since the minimum over the modified candidate set cannot exceed the error of any remaining candidate:

$$\begin{aligned} \mathrm{minADE}\,@K(P', x) &\leq \mathrm{ADE}(p'_{\tilde{j}}, x) && (42) \\ &= \mathrm{ADE}(p_{\tilde{j}}, x) && (43) \\ &= \mathrm{minADE}\,@K(P, x) && (44) \end{aligned}$$

This guarantee is conditional on the assumption $W(P, x) \neq j^*$ stated above: it says nothing about the case $W(P, x) = j^*$, i.e. when the replaced slot was the true winner. The following subsection makes this tradeoff explicit by decomposing the change in error into a harm term and a benefit term, recovering the present lemma as the special case in which the harm term is zero; Appendix C.3 then bounds how often the harmful case can occur.

C.2 BENEFIT–HARM DECOMPOSITION

Lemma 1 gives a binary safety condition but says nothing about the *magnitude* of the change in error, nor about what happens when the condition fails. The following result decomposes the change in $\mathrm{minADE@}K$ into two non-negative quantities: the *harm* of evicting slot j^*, defined as how much the achievable minimum would rise if that slot were simply removed, and the *benefit* of inserting a, defined as how much a improves on the best of the remaining candidates. Harm depends only on j^* and is zero exactly when j^* was not the original winner; benefit depends only on the retrieved candidate a and is zero whenever a fails to beat the remaining candidates. The net change in error is their difference, so Lemma 1 is recovered as the special case in which harm is zero.

Lemma 2 (Benefit–Harm Decomposition). *Let P' be obtained from P by replacing slot j^* with a candidate a, and let*

$$\mathrm{minADE}_{-j^*}\,@K(P,x) = \min_{k\neq j^*} \mathrm{ADE}(p^k,x)$$

denote the best achievable error using P's candidates other than slot j^*. Then

$$\mathrm{minADE}\,@K(P',x) - \mathrm{minADE}\,@K(P,x) = \mathrm{harm}(j^*,x) - \mathrm{benefit}(a,j^*,x). \tag{45}$$

where

$$\mathrm{harm}(j^*,x) = \mathrm{minADE}_{-j^*}\,@K(P,x) - \mathrm{minADE}\,@K(P,x) \geq 0, \tag{46}$$

$$\mathrm{benefit}(a,j^*,x) = \max\left(0,\, \mathrm{minADE}_{-j^*}\,@K(P,x) - \mathrm{ADE}(a,x)\right) \geq 0. \tag{47}$$

Proof. Since P' agrees with P on every slot except j^*, where it holds a instead of p^{j^*},

$$\mathrm{minADE}\,@K(P',x) = \min\left(\mathrm{minADE}_{-j^*}\,@K(P,x), \mathrm{ADE}(a,x)\right). \tag{48}$$

Using the identity

$$\min(u,v) = u - \max(0, u-v),$$

with

$$u = \mathrm{minADE}_{-j^*}\,@K(P,x), \qquad v = \mathrm{ADE}(a,x),$$

we obtain

$$\mathrm{minADE}\,@K(P',x) = \mathrm{minADE}_{-j^*}\,@K(P,x) - \mathrm{benefit}(a,j^*,x). \tag{49}$$

Substituting

$$\mathrm{minADE}_{-j^*}\,@K(P,x) = \mathrm{minADE}\,@K(P,x) + \mathrm{harm}(j^*,x)$$

gives

$$\begin{aligned}\mathrm{minADE}\,@K(P',x) = {}& \mathrm{minADE}\,@K(P,x) \\ & + \mathrm{harm}(j^*,x) - \mathrm{benefit}(a,j^*,x).\end{aligned} \tag{50}$$

Rearranging yields the stated result.

The non-negativity of $\mathrm{harm}(j^*,x)$ follows from

$$\begin{aligned}\mathrm{minADE}\,@K(P,x) &= \min\left(\mathrm{minADE}_{-j^*}\,@K(P,x), \mathrm{ADE}(p^{j^*},x)\right) \\ &\leq \mathrm{minADE}_{-j^*}\,@K(P,x).\end{aligned} \tag{51}$$

The non-negativity of $\mathrm{benefit}(a,j^*,x)$ follows directly from the $\max(0,\cdot)$ in its definition.

If $j^* \neq W(P,x)$, removing a non-minimal candidate does not change the achievable minimum. Thus,

$$\mathrm{minADE}_{-j^*}\,@K(P,x) = \mathrm{minADE}\,@K(P,x), \tag{52}$$

which implies

$$\mathrm{harm}(j^*,x) = 0. \tag{53}$$

Consequently, the decomposition reduces to Lemma 1:

$$\mathrm{minADE}\,@K(P',x) - \mathrm{minADE}\,@K(P,x) = -\,\mathrm{benefit}(a,j^*,x) \leq 0. \tag{54}$$

Conversely, if $j^* = W(P,x)$ and the minimizer is unique, then $\mathrm{harm}(j^*,x)$ is strictly positive, since removing the unique minimizer strictly increases the achievable minimum.

The identical argument applies verbatim to $\mathrm{minFDE}\,@K$ by replacing ADE with FDE throughout.

C.3 Finite-Sample Confidence Bound

Let θ_{j^*} denote the probability that slot j^* is the winning slot under the reference distribution.

If the slot is observed to win zero times over M independent examples:

$$X \sim \text{Binomial}(M, \theta_{j^*}), \tag{55}$$

with:

$$X = 0, \tag{56}$$

then:

$$P(X = 0) = (1 - \theta_{j^*})^M \tag{57}$$

The one-sided Clopper–Pearson upper bound at confidence level $1 - \alpha$ satisfies:

$$(1 - \theta_{j^*})^M = \alpha \tag{58}$$

Solving gives:

$$\theta_{j^*} \leq 1 - \alpha^{1/M} \tag{59}$$

This bound describes the marginal probability of a slot winning under the reference distribution, i.e. how often the premise of Lemma 1 ($W(P, x) \neq j^*$) holds across the reference population, rather than the probability for the query currently being served. It justifies which slots are safe to sacrifice on average; no individual replacement carries a per-query guarantee.

D Causal Online Self-Tuning

The online self-tuning procedure used in the primary evaluation selects retrieval and injection parameters only from trajectories whose futures have already elapsed.

D.1 Temporal Fit-and-Check Split

Let $\mathcal{L}_t$ denote the set of trajectories whose futures have become available by deployment frame t.

Let t_0 be the median completion time of the available trajectories:

$$t_0 = \text{median}\{t_j^{\text{end}} : e_j \in \mathcal{L}_t\} \tag{60}$$

The elapsed trajectories are divided chronologically into two subsets:

$$\mathcal{F}_t = \{e_j \in \mathcal{L}_t | t_j^{\text{end}} < t_0\}, \tag{61}$$

and

$$\mathcal{C}_t = \{e_j \in \mathcal{L}_t | t_j^{\text{end}} \geq t_0\} \tag{62}$$

The first subset is used for fitting candidate configurations, while the second, strictly later-arriving subset provides an independent temporal check, splitting by completion time rather than at random ensures that the outcomes in $\mathcal{C}_t$ had not yet occurred at the time $\mathcal{F}_t$ was used for fitting.

Configuration selection starts only when both subsets contain at least: $\eta = 10$

completed trajectories.

D.2 Configuration Evaluation

For a configuration c, the gain on a trajectory set U is computed as:

$$\mathrm{gain}(c, U) = 100 \times \frac{E_{\mathrm{base}}(U) - E_{\mathrm{TCMA}}(c, U)}{E_{\mathrm{base}}(U)} \tag{63}$$

A configuration is accepted only when it improves both temporal subsets independently, i.e. $\mathrm{gain}(c, \mathcal{F}_t) > 0$ and $\mathrm{gain}(c, \mathcal{C}_t) > 0$ are required separately rather than only their sum, which prevents a configuration that overfits one half from being selected on the strength of the other:

$$\mathrm{gain}(c, \mathcal{F}_t) > 0, \tag{64}$$

and

$$\mathrm{gain}(c, \mathcal{C}_t) > 0 \tag{65}$$

Among valid configurations, TCMA selects:

$$c^* = \arg\max_c \left[\mathrm{gain}(c, \mathcal{F}_t) + \mathrm{gain}(c, \mathcal{C}_t)\right] \tag{66}$$

If no configuration satisfies these conditions, TCMA retains the current configuration and abstains from augmentation.

D.3 Prospective Application Rule

The selected configuration is applied only to future queries:

$$c_\tau = c^*, \qquad \tau \geq t \tag{67}$$

Predictions generated before configuration selection are never modified retroactively.

This temporal separation ensures that self-tuning uses only information available at deployment time, extending the causal admissibility rule of Eq. 26 from memory retrieval to configuration selection itself: no part of TCMA conditions on unobserved future outcomes.

D.4 Precise Activation Condition

The activation indicator g_i used in Eq. 23/69 is determined jointly by two conditions operating at different levels, and both must hold for $g_i = 1$.

Scene-level condition. A configuration c^* must have been accepted for the query's scene under the temporal fit-and-check rule of Section D.1–D.2: both $\mathcal{F}_t$ and $\mathcal{C}_t$ must contain at least $\eta = 10$ elapsed examples, and c^* must satisfy $\mathrm{gain}(c^*, \mathcal{F}_t) > 0$ and $\mathrm{gain}(c^*, \mathcal{C}_t) > 0$ simultaneously Eq. 20. Until this holds, no configuration is active for the scene, and every query in it retains the base prediction for the remainder of the evaluation phase. This is the condition responsible for the per-scene abstention entries marked "–" in Table 1: on those scenes, no configuration satisfies this rule throughout the entire evaluation phase.

Query-level condition. Given an accepted configuration, an individual query i is additionally subject to the density gate $\rho(i)$ of Eq. 11 (world-frame retrieval) or its ego-centric equivalent (Section 3.1): a query lacking sufficient eligible same-scene neighbours retains the base prediction even within an otherwise-active scene.

Formally,

$$g_i = 1 \iff \big(\text{a configuration } c^* \text{ is accepted for query } i\text{'s scene}\big) \ \wedge \ \rho(i) = 1.$$

Otherwise $g_i = 0$ and TCMA returns P_i^{base} unchanged (Eq. 69).

E Complete TCMA Pipeline

Algorithmically, TCMA first queries the frozen predictor, then applies causal retrieval, gating, and slot injection when an accepted configuration is available. The base predictor output is:

$$P_i^{\text{base}} = \pi(X_i^{\text{obs}}) \tag{68}$$

The augmented (final) output is:

$$P_i^{\text{MR}} = \begin{cases} I(P_i^{\text{base}}, \mathcal{N}_i), & g_i = 1, \\ P_i^{\text{base}}, & g_i = 0, \end{cases} \tag{69}$$

where $\mathcal{N}_i$ is the retrieved trajectory set and $I(\cdot)$ denotes candidate-slot injection. The case $g_i = 0$ corresponds exactly to the abstention entries reported as "–" in the per-scene results tables, when the online gate has not accepted a configuration, TCMA returns the frozen predictor's output unchanged.

F Computational Complexity

For ego-centric retrieval, the descriptor dimension is $d = 6$.

A brute-force search over the eligible memory bank has complexity:

$$O(|\mathcal{M}_t|d) \tag{70}$$

For world-frame retrieval, the search complexity is:

$$O(|\mathcal{M}_t^{\text{scene}}|) \tag{71}$$

Memory construction is linear in the number of stored trajectories.

Configuration selection is performed only at scheduled tuning boundaries and does not contribute to per-query inference latency.

G Black-Box and Training-Free Properties

TCMA operates strictly at the prediction-output level. It requires no gradients, model parameters, internal features, architectural modifications, or learned adapters. The base predictor remains frozen and only exported candidate trajectories are used.

- no gradients,
- no access to model parameters,
- no internal feature representations,
- no architectural modification,
- no additional learned adapter.

H Experimental Protocol and Reproduction Details

H.1 Dataset and Evaluation Setup

All experiments are conducted on the ETH–UCY pedestrian trajectory benchmark consisting of five scenes:

- ETH
- HOTEL
- UNIV

- ZARA1
- ZARA2

Each trajectory contains an observed history of eight frames followed by a twelve-frame prediction horizon.

The evaluation follows a leave-one-scene-out protocol. For each experiment, four scenes are used for training statistics, validation, and reference memory construction, while the remaining scene is held out for evaluation.

All predictors are evaluated using their exported candidate trajectories.

The reported metrics are: minADE@20 and minFDE@20

H.2 Online Evaluation Boundary

The primary evaluation follows the causal online protocol.

For a recording spanning frames from $t_{\min}$ to $t_{\max}$, the evaluation boundary is:

$$t_{\text{eval}} = t_{\min} + 0.4(t_{\max} - t_{\min}) \tag{72}$$

The coefficient 0.4 was fixed by convention rather than tuned via a sensitivity sweep, a systematic study of how the choice of warm-up fraction trades off memory maturity against evaluation-set size is left to future work. Trajectories whose futures finish before this boundary may be used for:

- causal memory construction,
- self-tuning,
- reference statistics.

However, they are not included in the final prospective evaluation.

All evaluated predictions occur after the boundary.

H.3 Population Accounting

The ETH scene contains different test populations depending on the predictor evaluation pipeline.

Five predictors use the shared evaluation loader and contain 181 ETH test agents.

TUTR and MemoNet are evaluated using their original published pipelines, resulting in 364 ETH test agents.

HOTEL, UNIV, ZARA1, and ZARA2 use identical agent populations across all predictors.

Therefore, comparisons involving ETH averages should consider this population difference.

H.4 Predictor Reproduction

TCMA treats all base predictors as frozen black boxes.

The evaluated predictors include:

- CVM-S
- TrajEvo heuristic
- standalone predictor
- Graph-TERN
- MemoNet
- TUTR
- EigenTrajectory

Table 4: Per-scene minADE@20 (base/augmented/gain, in metres) under the online world-frame self-tuning protocol. N is the number of evaluation-phase agents (after $t_{\text{eval}} = t_{\min} + 0.4(t_{\max} - t_{\min})$), excluding warm-up agents used only for configuration selection.

Predictor	ETH	HOTEL	UNIV	ZARA1	ZARA2
CVM-S	N=124 –	N=780 0.254/0.151/+40.5%	N=12224 0.367/0.304/+17.3%	N=1369 0.285/0.235/+17.6%	N=4618 0.202/0.175/+13.2%
TrajEvo heuristic	N=124 –	N=780 0.130/0.119/+8.6%	N=12224 0.343/0.303/+11.6%	N=1369 0.288/0.246/+14.8%	N=4618 0.193/0.160/+17.3%
Ours (standalone)	N=124 –	N=780 0.131/0.123/+6.1%	N=12224 0.284/0.273/+4.1%	N=1369 0.253/0.230/+9.0%	N=4618 0.164/0.152/+7.3%
Graph-TERN	N=124 –	N=780 0.144/0.126/+12.3%	N=12224 0.228/0.225/+1.5%	N=1369 –	N=4618 0.144/0.132/+8.3%
MemoNet	N=303 –	N=732 0.122/0.118/+3.5%	N=12224 0.240/0.234/+2.4%	N=1438 0.180/0.181/-0.8%	N=4615 0.129/0.123/+4.4%
TUTR	N=303 –	N=732 0.134/0.130/+3.2%	N=12224 0.229/0.224/+2.2%	N=1438 –	N=4615 0.127/0.122/+4.0%
EigenTrajectory	N=124 –	N=780 0.119/0.115/+3.7%	N=12224 0.233/0.229/+1.6%	N=1369 –	N=4618 0.124/0.120/+3.5%

Table 5: Per-scene minFDE@20 (base/augmented/gain) under the online world-frame self-tuning protocol. Conventions as in Table 4: N is the evaluation-phase sample size, "–" denotes abstention.

Predictor	ETH	HOTEL	UNIV	ZARA1	ZARA2
CVM-S	N=124 –	N=780 0.480/0.252/+47.6%	N=12224 0.773/0.585/+24.4%	N=1369 0.555/0.401/+27.7%	N=4618 0.424/0.340/+19.8%
TrajEvo heuristic	N=124 –	N=780 0.215/0.187/+12.8%	N=12224 0.686/0.573/+16.5%	N=1369 0.595/0.461/+22.5%	N=4618 0.385/0.289/+24.7%
Ours (standalone)	N=124 –	N=780 0.205/0.189/+7.8%	N=12224 0.558/0.523/+6.3%	N=1369 0.483/0.415/+14.1%	N=4618 0.318/0.284/+10.7%
Graph-TERN	N=124 –	N=780 0.230/0.194/+15.5%	N=12224 0.368/0.360/+2.0%	N=1369 –	N=4618 0.232/0.210/+9.6%
MemoNet	N=303 –	N=732 0.187/0.175/+6.5%	N=12224 0.431/0.414/+3.9%	N=1438 0.312/0.313/-0.5%	N=4615 0.221/0.210/+5.2%
TUTR	N=303 –	N=732 0.210/0.203/+3.1%	N=12224 0.416/0.407/+2.0%	N=1438 –	N=4615 0.220/0.212/+3.6%
EigenTrajectory	N=124 –	N=780 0.183/0.176/+3.6%	N=12224 0.413/0.402/+2.6%	N=1369 –	N=4618 0.212/0.202/+4.7%

For learned predictors, released checkpoints are used without retraining or fine-tuning.

Before applying TCMA augmentation, baseline reproduction was verified.

The reproduced results are within:

- 0.015 ADE of the published TUTR results,
- 0.01 ADE of the published MemoNet results.

Graph-TERN is evaluated using the authors' released checkpoints.

I Complete Online Results

I.1 Per-Scene Online Improvement

Table 4 and 5 reports the complete online self-tuning results.

Each entry shows percentage improvement in: $\min ADE@20 / \min FDE@20$.

The framework abstains when the causal history is insufficient.

Abstention leaves the original predictor output unchanged.

I.2 Aggregate Online Results

Across predictors, the average online improvement is 6.20% for $\min ADE@20$, and 8.47% for $\min FDE@20$.

The improvement range across predictors is $1.74\% - 17.70\%$ for $\min ADE@20$.

The largest gain is obtained by CVM-S, while smaller gains occur for predictors that already contain stronger learned trajectory priors.

J Offline Validation-Tuned Results

The offline protocol provides a secondary analysis where configurations are selected using validation data before evaluation.

Table 6: Offline validation-tuned protocol results.

Predictor	minADE base	minADE aug	Gain	minFDE base	minFDE aug	Gain
CVM-S	0.4037	0.2893	28.3%	0.8525	0.5439	36.2%
TrajEvo heuristic	0.2817	0.2712	3.7%	0.5311	0.5021	5.5%
Ours	0.2560	0.2482	3.1%	0.4698	0.4496	4.3%
Graph-TERN	0.2236	0.2157	3.5%	0.3357	0.3236	3.6%
TUTR	0.2202	0.2162	1.8%	0.3633	0.3590	1.2%
MemoNet	0.2209	0.2193	0.7%	0.3671	0.3651	0.5%
EigenTrajectory	0.2092	0.2085	0.3%	0.3426	0.3422	0.1%

K Additional Ablation Studies

K.1 Memory Construction Ablation

To evaluate the contribution of causal memory, we compare three memory configurations inside the standalone predictor:

1. no memory,
2. static memory only,
3. static memory combined with causal online memory.

The results are reported in Table 7.

Table 7: Memory construction ablation inside the standalone predictor. Values are per-agent $\min ADE@20$ in metres.

Configuration	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
No memory	0.450	0.139	0.321	0.278	0.196	0.277
Static memory only	0.420	0.132	0.298	0.252	0.179	0.256
Static + causal memory	**0.418**	**0.127**	**0.283**	**0.239**	**0.174**	**0.248**

The causal memory configuration improves over static memory on every evaluated scene.

K.2 Static versus Causal Retrieval Source

The effect of causal memory is further evaluated using CVM-S.

Table 8: Comparison between static and causal memory sources for CVM-S.

Source	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG Gain
Static bank only	+37.4%	+31.4%	+16.2%	+4.0%	+14.0%	+24.9%
Causal online memory	+39.7%	+35.7%	+19.8%	+11.2%	+15.6%	+28.3%

Although we evaluate ego-centric and world-frame retrieval as separate protocols, the results indicate that they capture different information. Table 8 compares a static ego-centric memory bank

with the causal online world-frame memory on CVM-S. The static bank already provides substantial gains, but the online world-frame channel improves further in every scene, with the largest additional gain on ZARA1 (+4.0% vs. +11.2%). This suggests that ego-centric retrieval transfers general motion patterns, while world-frame retrieval exploits scene-specific recurrences that are not captured by the offline bank. The two channels are therefore complementary in what they retrieve.

K.3 Gate Design Ablation

The admission gate controls when retrieved trajectories are injected.

We compare:

- v1: scene-relative percentile gate,
- v2: absolute distance plus minimum-history gate.

Table 9: Gate design ablation on the TrajEvo heuristic predictor.

Gate Design	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
v1: scene-relative percentile	-12.1%	+3.7%	+7.0%	+3.5%	+5.5%	+0.1%
v2: absolute distance + history	-0.3%	-0.3%	+11.6%	-0.6%	+6.5%	+3.7%

The final framework uses the second gate because it avoids relying on scene-relative statistics and uses only information available during deployment.

K.4 Descriptor Dimension Ablation

The ego-centric retrieval descriptor was evaluated using six and nine dimensions.

The six-dimensional descriptor contains:

- last-frame speed
- mean speed
- speed variation
- last-frame acceleration
- mean acceleration
- path curvature

The nine-dimensional alternative additionally includes:

- total displacement
- stillness fraction
- straightness ratio

Table 10: Validation $\min ADE@20$ after augmentation using different descriptor dimensions.

Descriptor	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
6-dimensional	0.2388	0.2494	0.2081	0.2427	0.2632	0.2404
9-dimensional	0.2388	0.2496	0.2071	0.2430	0.2624	0.2402

The two descriptors produce nearly identical validation performance. The six-dimensional descriptor is retained because it provides the same performance with fewer components.

L COMPUTATIONAL ANALYSIS

L.1 RUNTIME MEASUREMENT

The computational overhead is measured on UNIV, the largest evaluated scene containing 24,334 test agents.

All retrieval and gating operations are executed on CPU.

Table 11: Measured computational cost on UNIV.

Stage	Measured Cost
Offline memory construction	54.5 ms
Ego-centric retrieval	717 μs/query
World-frame retrieval	891 μs/query
Slot injection	1.2 μs/query
Self-tuning pass	1413 s ($\approx$23.6 min)
Peak memory bank size	5.3 MB

The self-tuning stage is executed once per scene and predictor configuration.

It is not part of the per-query inference latency.

The additional latency introduced during prediction is $717\mu s$ for ego-centric retrieval and $891\mu s$ for world-frame retrieval.

M STATISTICAL REPORTING

Statistical comparisons are performed using paired one-sided Wilcoxon signed-rank tests on per-agent $\min ADE@20$ values.

The primary online evaluation contains 25 predictor-scene combinations where TCMA acts.

Among these:

- 24 combinations improve $\min ADE@20$ with $p < 0.001$
- one combination shows a negative but non-significant change

The non-improving case corresponds to MemoNet on ZARA1 -0.8% change in $\min ADE@20$ and -0.5% change in $\min FDE@20$.

Exact individual p-values and confidence interval endpoints are retained in the experimental logs and are not reproduced here.