LEARNING WHEN TO CALL: PREDICTIVE OPPORTUNITY RANKING FOR EMBODIED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied agents must maintain continuous control while selectively invoking expensive reasoning backends. With a limited call budget, poorly timed requests may miss useful pre-event opportunities. We focus on event-relative opportunity prediction: using causal observations to anticipate pre-onset windows in which to place reasoning requests. The Predictive Reasoning Scheduler (PRS) implements a lightweight scorer from recent state history and current bird's-eye-view context; a separate shared selector enforces budget and cooldown constraints. We evaluate offline exact-call placement on frozen trajectories using Useful-Window Coverage (UWC), the fraction of distinct event windows reached by at least one request. On CARLA Town10HD, a development-selected checkpoint covers 23 of 63 windows with 165 calls, compared with approximately 6 for Random-k. The UWC@$0.5\,\mathrm{s}$ gain is 26.71 percentage points (95% bootstrap interval: 12.64–40.97). On fresh Town10HD scenarios, a model selected using a prespecified development procedure achieves a normalized log-budget UWC-AUC gain of 0.2392 over Random-k. Controlled retraining also finds that current-state inputs and broader supervision can outperform the reference design. These findings support event-relative opportunity prediction as a learning formulation for improving the temporal placement of budgeted reasoning calls.[1]

1 INTRODUCTION

Embodied agents combine continuously running policies with language or multimodal reasoning backends for planning and decision support (Ichter et al., 2023; Liang et al., 2023; Driess et al., 2023; Huang et al., 2023). When these calls are costly, their temporal placement becomes an important part of allocating a limited reasoning budget. Calling at every control step consumes resources on routine states, but waiting for a reactive trigger may miss a useful pre-event opportunity. Calling too early may leave insufficient evidence, while calling too late may leave insufficient time to use the response. The challenge is therefore to anticipate when a limited number of requests should be placed.

Selective tool use and model routing already study how to allocate external computation (Schick et al., 2023; Ong et al., 2025), and embodied orchestration includes sequential invocation and resource decisions (Liu et al., 2026). Against this broader backdrop, we focus on a narrower temporal aspect of this problem: *With call counts and allocation rules held fixed, can causal event-relative predictions improve pre-onset request placement?*

Two properties distinguish this question from ordinary frame classification. First, event association alone does not determine an appropriate request time: an observation may occur too early, inside a pre-onset target window, or after the event boundary. Second, candidate times compete for limited budget and cooldown capacity, so individually promising predictions do not directly determine a good call set. Evaluation must therefore account for both event-relative timing and the distinct events reached by the selected calls.

Our core idea is to *anticipate before allocating*. We introduce the Predictive Reasoning Scheduler (PRS), whose learned component scores candidate request times from recent structured state history and current bird's-eye-view (BEV) context. A shared constrained selector converts these scores into

[1] Supplementary website: `https://predictive-timing-supp.pages.dev/`

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.

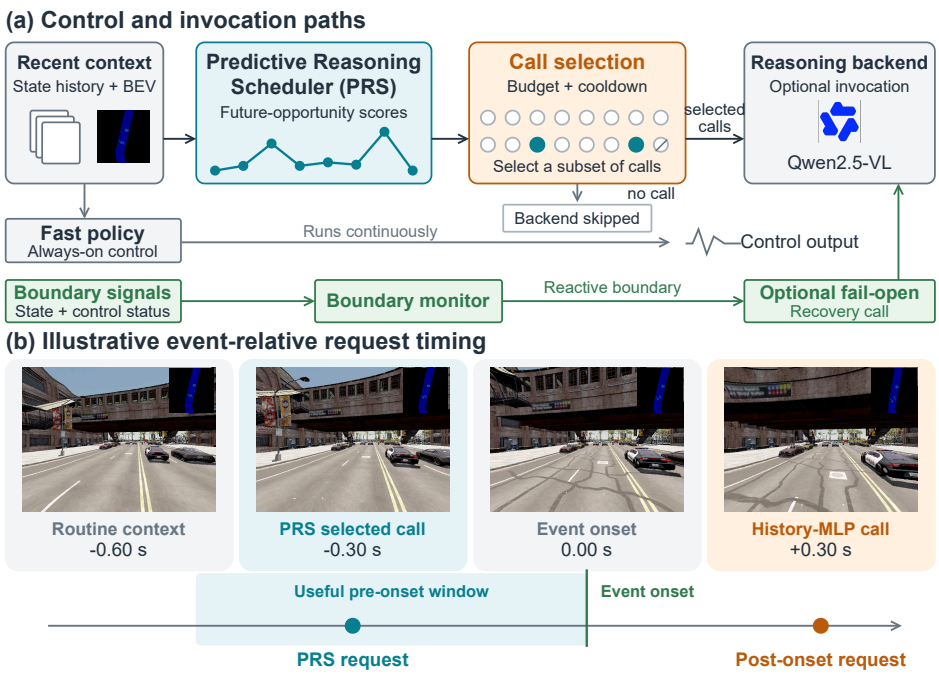

Figure 1: Selective reasoning scheduling. (a) Context feeds an opportunity scorer and constrained selector; the fast policy maintains continuous control, with reactive fail-open recovery handled separately. Qwen2.5-VL denotes an evaluated backend interface. (b) Illustrative Town10HD timing example, with a pre-onset request at $-0.30\,\mathrm{s}$ and a History-MLP reference at $+0.30\,\mathrm{s}$. RGB/BEV frames come from a separate visualization run and illustrate window alignment.

calls (Figure 1). This separates learning the predictive signal from deciding how to allocate a fixed call allowance: PRS supplies opportunity scores, while the selector enforces budget and cooldown.

We evaluate placement on frozen base-policy trajectories with method-independent event windows. To evaluate this placement objective, we introduce Useful-Window Coverage (UWC), which measures the fraction of distinct events reached by a request within the corresponding pre-onset window, independently of backend answer quality. The primary study ranks completed traces using scores computed from causal observations and matches call counts across methods. On CARLA Town10HD, a development-selected PRS checkpoint covers 23 of 63 windows with 165 calls, versus approximately 6 for Random-k; a separately selected model provides complementary evidence on freshly collected scenarios. Controlled learning comparisons examine how inputs, encoders, and supervision affect these gains, and a separate online study examines the gap between offline rankings and sequential calls.

Contributions. (1) We formulate a predictive-placement subproblem of embodied reasoning invocation as event-relative opportunity prediction and evaluate its induced placement at fixed call budgets. (2) We instantiate this predictive-placement formulation with a compact causal scorer and an independent constrained selector, allowing predictive quality to be compared under shared allocation rules. (3) We establish coverage gains for development-selected models and use controlled learning comparisons to characterize how those gains depend on representation and supervision choices.

2 Related Work

Computation allocation, external calls, and routing. Adaptive computation time, SkipNet, and CALM adjust computation during prediction or generation (Graves, 2016; Wang et al., 2018; Schuster et al., 2022); test-time scaling studies reasoning-effort allocation (Snell et al., 2025). External-invocation methods instead choose when to use tools or retrieval (Schick et al., 2023; Asai et al., 2024; Jeong et al., 2024), while model routing and learning to defer choose among models or experts (Ong

et al., 2025; Mozannar & Sontag, 2020; Narasimhan et al., 2022). These decisions address the quality and cost of selective computation. We examine a complementary temporal question: under the same call allowance, which moments along a physical trajectory should receive requests to reach distinct event-relative opportunities?

Embodied reasoning and metareasoning. SayCan, Code as Policies, PaLM-E, and Inner Monologue connect language or multimodal reasoning to action (Ichter et al., 2023; Liang et al., 2023; Driess et al., 2023; Huang et al., 2023). The closely related RARRL learns invocation, role, and budget decisions from observations, execution history, and remaining resources, evaluating orchestration through task performance and latency (Liu et al., 2026). We isolate opportunity prediction by holding trajectories and allocation rules fixed and measuring event-relative request coverage. This question connects to metareasoning about the benefit and cost of computation (Russell & Wefald, 1991) and learned planner-performance profiles for deciding when to execute (Budd et al., 2024). Runtime assurance, including Simplex and SOTER, addresses controller switching and recovery separately (Seto et al., 1998; Desai et al., 2019).

Event anticipation and triggering. Accident anticipation forecasts future events (Suzuki et al., 2018), while predictive triggering forecasts communication needs in control (Trimpe & Baumann, 2019). These approaches are directly relevant sources of predictive signals. Our formulation distinguishes event association over a broad horizon from placing a fixed number of requests inside specified pre-onset windows. Predicting an event earlier need not place a request inside that window; conversely, a broad training target may still produce effective rankings. Applying a shared selector to anticipation scores allows their pre-onset window coverage to be compared under identical call budgets and cooldown constraints.

3 Event-Relative Call Placement

We formalize event-relative call placement as the connection between two objects: a score predicted from context available at each candidate request time, and a call set formed from these scores under shared constraints. We fix the base control policy and recorded trajectories, so the compared call sets face the same event opportunities.

3.1 Causal Opportunity Prediction

Let $\mathcal{T}_r$ denote the set of eligible request ticks on trajectory r, and let $\mathcal{T} = \bigcup_r \mathcal{T}_r$ denote their union. Trajectory identity is retained separately through the sets $\mathcal{T}_r$ and the quotas K_r below. At tick t, context c_t contains only observations available by that time. Event i has onset e_i on trajectory $r(i)$; events and onsets are defined independently of the compared methods on the frozen trajectories, including lead-vehicle braking, pedestrian crossing, and red-light approaches. For width $\Delta > 0$, define the pre-onset request window

$$W_i(\Delta) = \{t \in \mathcal{T}_{r(i)} : e_i - \Delta \le t < e_i\}. \tag{1}$$

The onset is excluded. We use $\Delta = 0.5\,\mathrm{s}$ as the primary operational target width, with sensitivity to alternative widths evaluated separately. Training uses recorded event assignments: a tick assigned onset e_t^{ann} receives label $y_t^{\mathrm{win}} = \mathbf{1}[0 < e_t^{\mathrm{ann}} - t \le \Delta]$, and off-event examples receive zero. Future onsets supply labels and evaluation windows, never scorer inputs.

We learn a scorer $s_t = g_\theta(c_t)$ to prioritize candidate times using this temporal supervision. The predictive task uses currently available observations to infer whether a candidate request time lies in a pre-onset event window. Section 4 instantiates this scoring function as a probability sum over time-to-event bins.

3.2 Budgeted Temporal Placement

For a call set $A \subseteq \mathcal{T}$, write $A_r = A \cap \mathcal{T}_r$. Each trajectory receives a quota K_r, and cooldown τ_c requires

$$|A_r| \le K_r, \qquad |t - t'| \ge \tau_c \quad \text{for distinct } t, t' \in A_r. \tag{2}$$

Our primary comparisons use the exact-call special case: all methods fill the same feasible per-trajectory quotas. Section 6 specifies this common allowance.

For N eligible events, Useful-Window Coverage evaluates the call set:

$$\mathrm{UWC}\,@\Delta(A) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[A \cap W_i(\Delta) \neq \varnothing]. \tag{3}$$

Each reached event contributes once, and all eligible events remain in the denominator; in-window call precision instead divides by the number of calls. Equal-size call sets can have different UWC even when their invocation rates match. With our default $\Delta = \tau_c = 0.5\,\mathrm{s}$, a feasible call set hits each window at most once. Out-of-window calls and budget/cooldown conflicts can still leave events uncovered. Here "useful window" denotes an operational request target; UWC does not measure the value of a returned answer.

Window supervision supplies a local learning signal, while a shared selector turns the scores into a constrained call set. Evaluation then asks whether the learned priorities reach more distinct windows. This is a predictive surrogate for placement: the local loss does not directly optimize set-level UWC, marginal coverage, or the value of waiting (Appendix D.1).

4 Predictive Reasoning Scheduler

We instantiate event-relative opportunity prediction with the Predictive Reasoning Scheduler (PRS). PRS encodes observed context, learns a temporal opportunity score, and passes the resulting scores to a separate constrained selector. Figure 2 shows a reference implementation of the temporal scorer. The reference design addresses three requirements of this formulation: representing causal context, capturing event-relative timing, and exposing a scalar priority for constrained selection.

4.1 From Causal Context to Opportunity Scores

The first requirement is to map causal temporal and spatial context into an event-relative opportunity score. We use the following lightweight scorer as a reference implementation. The scorer receives 20 structured states $h_t = (x_{t-19}, \ldots, x_t) \in \mathbb{R}^{20\times 17}$ and a current BEV raster $b_t \in \mathbb{R}^{3\times 64\times 64}$. A single-layer unidirectional GRU (Cho et al., 2014) encodes the history into 64 dimensions; a compact CNN encodes BEV into 32 dimensions. Their concatenation gives $f_t \in \mathbb{R}^{96}$. Each history window is encoded independently and ends at t, preserving causal inputs. The 23,447-parameter implementation has GPU batch-1 forward-pass p95 of 1.26 ms, excluding perception, selection, and backend execution. It provides a lightweight reference; Section 6.1 examines encoder dependence.

Four time-to-event (TTE) logits $\ell_t^{\mathrm{tte}} = W_{\mathrm{tte}} f_t + a_{\mathrm{tte}}$ yield probabilities $p_{t,j} = \mathrm{softmax}(\ell_t^{\mathrm{tte}})_j$. The first two categories jointly represent $0 < e_t^{\mathrm{ann}} - t \leq 0.5\,\mathrm{s}$, and their sum defines the opportunity score:

$$s_t = \sum_{j\in\mathcal{B}_{0.5}} p_{t,j}, \qquad \mathcal{B}_{0.5} = \{0, 1\}. \tag{4}$$

Higher mass in the target interval gives higher priority. This is the only head output passed to the selector, which receives neither event labels nor the other predictions. The resulting reference scorer has 23,447 parameters and a measured GPU batch-1 forward-pass p95 of 1.26 ms, excluding perception, selection, and backend execution.

4.2 Learning the Opportunity Score

The second requirement is to align the learned score with event-relative timing while preserving direct supervision of the target window. Accordingly, the TTE head receives categorical supervision of event-relative regions and binary supervision of target-window membership. For a binary loss on the probability sum in Equation 4, its equivalent logit is

$$u_t = \log \sum_{j\in\mathcal{B}_{0.5}} e^{\ell_{t,j}^{\mathrm{tte}}} - \log \sum_{j\notin\mathcal{B}_{0.5}} e^{\ell_{t,j}^{\mathrm{tte}}}, \qquad \sigma(u_t) = s_t. \tag{5}$$

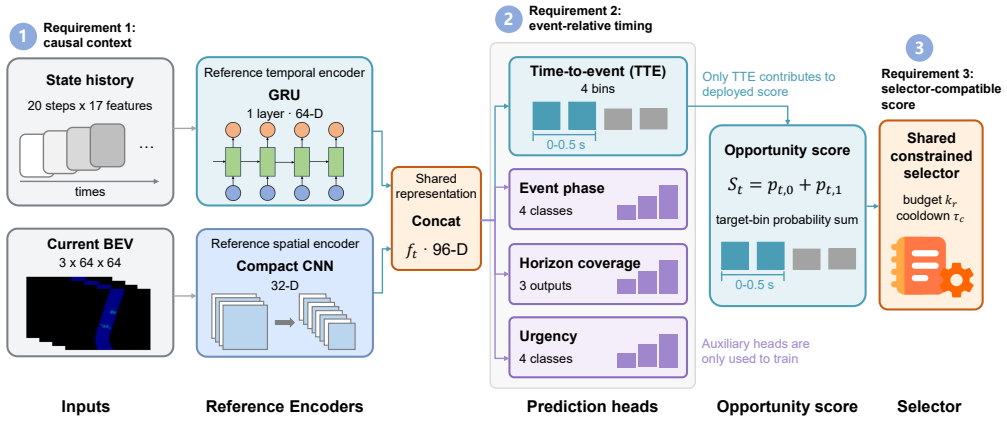

Figure 2: Reference opportunity scorer. The reference design addresses three requirements: (1) causal context representation, using recent state history and current BEV with temporal and spatial encoders; (2) event-relative timing prediction, using a TTE head with auxiliary phase, horizon, and urgency supervision; and (3) a selector-compatible opportunity score, obtained from the target TTE bins and passed to a shared budget-and-cooldown selector. Only the TTE head contributes to the deployed score; the auxiliary heads are used for training only.

As auxiliary reference choices, three additional heads on f_t predict event phase, horizon coverage, and urgency without entering the selection score. The complete reference loss is

$$\begin{aligned}\mathcal{L}_t = \; & \mathrm{CE}_{w_T}(\ell_t^{\mathrm{tte}}, y_t^{\mathrm{tte}}) + 0.5\,\mathrm{CE}_{w_P}(\ell_t^{\mathrm{phase}}, y_t^{\mathrm{phase}}) \\ & + 0.5\,\mathrm{BCE}_{+2}(u_t, y_t^{\mathrm{win}}) + 0.1\,\mathrm{BCE}(\ell_t^{\mathrm{hor}}, y_t^{\mathrm{hor}}) + 0.1\,\mathrm{CE}(\ell_t^{\mathrm{urg}}, y_t^{\mathrm{urg}}).\end{aligned} \tag{6}$$

Batches balance useful, too-early, boundary/post-onset, and off-event examples. CE denotes categorical cross-entropy with the indicated class weights; BCE uses logits, and $+2$ weights positive binary examples by 2. The TTE and window terms supervise the same head at different temporal granularities. Weighted sampling and losses change the effective training distribution, so we use the score for ranking without assuming deployment calibration. Appendix A.2 gives the class weights and optimizer settings; the learning comparisons test this full recipe against simpler supervision.

4.3 Converting Scores into Calls

The final requirement is to convert local opportunity scores into a feasible call set under shared resource constraints. For the primary offline evaluation, frozen scores from completed trajectories and quotas $K = (K_r)_r$ enter a shared selector:

$$A^K_{\mathrm{PRS}} = \mathrm{TopK}_{\mathcal{C}}(\{s_t : t \in \mathcal{T}\}, K). \tag{7}$$

Within each trajectory, it scans candidates in descending score order and accepts cooldown-compatible ticks until the quota is reached or candidates are exhausted; ties are broken by frame index. It uses no event labels, and quota shortfalls are reported separately. Sharing this rule across score-based methods holds allocation fixed, so their call-set differences reflect the rankings supplied (Appendix A.3).

Each score uses observations available by its candidate time, but ranking the completed trace compares it with later scores. The primary comparison therefore measures offline placement. The scorer does not condition on remaining quota, previous calls, or backend responses; Section 6.2 examines online commitment separately.

5 Experimental Setup

Data and checkpoint selection. We collect CARLA (Dosovitskiy et al., 2017) trajectories with a fixed TransFuser policy (Prakash et al., 2021). Town01 provides training, Town03 supports checkpoint selection, and Town05/Town06 supply development resource evaluations. The primary Town10HD

cohort uses a Town03-selected checkpoint on 63 eligible trajectories from 80 planned slots, with 63 events and 8,253 candidate ticks. For the separate fresh-scenario evaluation, we train ten seeds on Town01, select checkpoints on Town03, and choose the final model by a prespecified Town07-only rule. We then evaluate it on 72 fresh Town10HD episodes, comprising 18 route clusters and 12,822 ticks (Appendix A). Town07 also supports learning and allocation diagnostics and is a development domain for this final-model selection.

Baselines and shared constraints. We compare against a set of heuristic, risk-based, history-based, and oracle baselines under shared call constraints. Random-k averages feasible random selections; Periodic-k averages periodic selections across phases. Reactive padded-k starts from cooldown-compatible reactive-edge calls and uses current-risk scores to truncate or supplement them to the quota. Current-risk/TTC ranks present risk, and History-MLP scores historical observations. Exact-DP knows the event windows and supplies an upper bound under the same candidates, quotas, and cooldown. All methods share candidates, event windows, and a $0.5\,\mathrm{s}$ cooldown at nominal budgets $\rho \in \{0.5, 1, 2, 5, 10\}\%$. The exact-call allowance is $K_\rho = \lfloor \rho |\mathcal{T}| \rfloor$, with $\sum_r K_r = K_\rho$; the common per-trajectory quotas are fixed before scores are examined, and calls cannot move between trajectories (Appendix A.3).

Metrics and uncertainty. We report UWC@$0.5\,\mathrm{s}$ and normalized UWC-AUC, computed by trapezoidal integration over log budget and divided by the axis span. Paired 95% intervals use 10,000 bootstrap draws over 35 primary groups defined by event family, map anchor, and weather, or 18 fresh route clusters. Learning comparisons use the paired-seed and spatial resampling units specified in Appendix A.4.

Learning comparisons. Frozen-checkpoint input interventions and matched retraining answer different questions. We first alter the history presented to one selected model, then compare current-only, shuffled-history, and ordered-history training with three matched seeds. Separate encoder and target comparisons use five paired seeds; the target comparison holds the encoder and 1,000 updates per run fixed. A matched development search compares five methods and five seeds with four configurations per method and seed, each trained for 1,000 updates. These controls separate input sensitivity from the choice of representation, encoder, supervision, and development configuration.

6 RESULTS

We first test fixed-budget placement, then examine the learning choices behind the scores, and finally analyze call allocation.

At the primary 2% budget, every method receives 165 calls. The Town03-selected PRS checkpoint covers 23 of 63 windows (UWC 0.3651), versus approximately six on average for Random-k (0.0980) and UWC 0.0954 for Periodic-k. The paired advantage over Random-k is $+0.2671$ (95% interval $[0.1264, 0.4097]$). Across five budgets, selected-model UWC rises from 0.0794 to 0.8571, and it exceeds Random-k, Periodic-k, and History-MLP at every budget. In contrast, Current-risk/TTC and Reactive padded-k remain at zero through 5% and reach 0.3016 only at 10% (Table 3). All methods meet their assigned quotas and cooldown constraints.

Fresh-scenario evaluation. The model chosen by the separately frozen Town07-only selection procedure is evaluated on newly collected Town10HD scenarios. At 2%, 256 calls yield UWC 0.3194, versus 0.1007 for Random-k and 0.1250 for both History-MLP and Reactive padded-k. The selected model exceeds Random-k at all five budgets, with normalized log-budget AUC gain $+0.2392$ (95% interval $[0.1159, 0.3690]$). Figure 3c retains the secondary paired comparisons with History-MLP and Reactive padded-k. AUC differences against Random-k are nonnegative in 14 of 18 route clusters.

Event association and window alignment. At the primary 2% point, Current-risk and Reactive padded-k each select 63 event-associated calls classified as too early and 102 unmatched calls, yielding zero UWC. These selections illustrate why event association alone does not ensure target-window coverage.

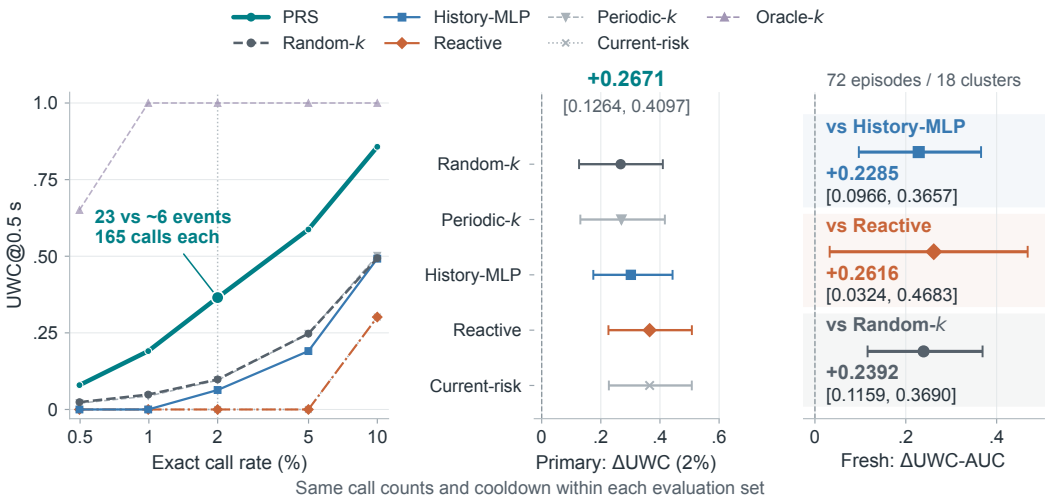

Figure 3: Same call budget, more events covered. (a) Primary coverage frontier. (b) UWC differences at 2%, with 95% paired bootstrap intervals over 35 protocol-defined groups. (c) Fresh-scenario normalized log-budget UWC-AUC differences for the Town07-selected model, with 95% paired intervals over 18 route clusters (10,000 draws). Call counts and cooldown are matched within each evaluation set. Full comparisons appear in Appendix B.

Development invocation-rate comparison. A separate Town06 development evaluation provides a complementary unequal-rate operating point. PRS achieves 32.53% UWC at a 7.32% call rate, whereas reactive triggering reaches 21.69% UWC at 26.39%. Thus, the favorable placement observed under matched budgets is also visible at this development operating point with substantially fewer requests. Invocation rate measures request frequency.

6.1 What Shapes the Learned Opportunity Ranking?

History interventions and retraining. We examine input history, encoders, and supervision through frozen-checkpoint interventions and controlled retraining (Table 1). For the history interventions, we hold the checkpoint, candidates, quotas, and cooldown fixed, alter the history input, and recompute scores and selections. Reversal and permutation preserve past observations while changing their order; current-repeat removes historical variation. Ordered inputs yield higher equal-domain, equal-budget aggregate UWC than current-repeat, reversed-past, and permuted-past inputs (Figure 4). On Town07, ordered-minus-current-repeat is negative at 0.5% and 1%, and positive at 2%, 5%, and 10%. In the separate retraining comparison with three matched seeds, current-only inputs yield higher aggregate coverage than ordered history, while the ordered–shuffled interval includes zero (Appendix C.1).

Encoder and supervision choices. In a separate matched-objective encoder comparison with five paired seeds, causal TCN reaches a mean Town10HD AUC of 0.4616, a gain of $+0.2754$ over Random-k, with four positive seeds and positive mean differences at all five budgets. On Town07, its mean AUC difference from Random-k is -0.0063; on Town10HD, the interval against current-state training includes zero (Appendix C.2). The matched supervision comparison further examines the training target: under the same half-second UWC metric, broad two-second event supervision yields higher coverage than half-second supervision and the full reference objective (Appendix C.3).

Matched development search. A matched development search compares five methods and five seeds under equal search opportunities, comprising 100 training runs and 25 development-selected models (Appendix C.4). Alternative encoders or supervision targets yield higher mean equal-domain AUC than the canonical GRU configuration. Under the study's conditional bootstrap analysis, the

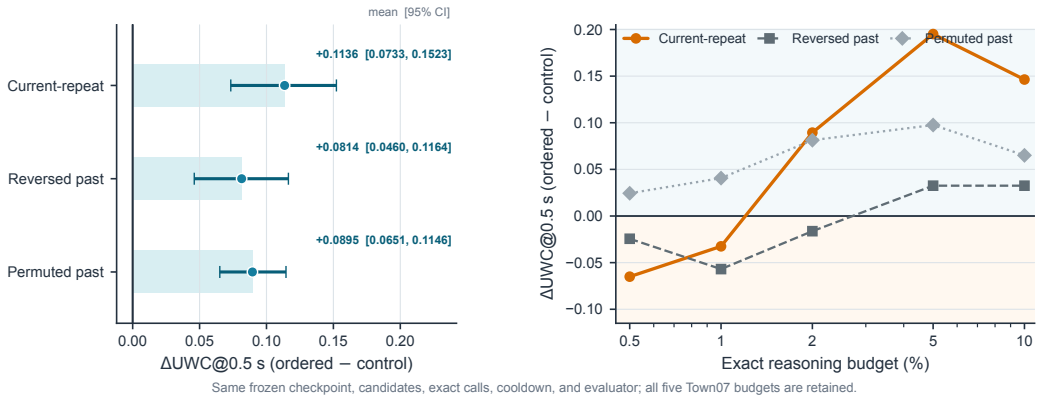

Figure 4: Temporal context shapes the selected scorer's coverage. (a) Equal-domain, equal-budget mean UWC differences with 95% paired bootstrap intervals. (b) Town07 ordered-minus-control differences across all five budgets. The checkpoint and call quotas are fixed; scores and selected times are recomputed for each intervention.

TCN and broader-horizon GRU contrasts pass the multiplicity-adjusted 0.05 threshold; the GRU/PRS contrasts with Current/PRS and Transformer/PRS each have adjusted $p = 0.116588$.

Table 1: Learning comparisons and their scope. Controlled retraining complements the frozen-history interventions in Figure 4. All comparisons evaluate half-second window coverage; estimates and tests follow each study's paired-seed and spatial resampling design.

Comparison	Reported finding	Interpretation
Input retraining 3 matched seeds	Current-only exceeds ordered history in aggregate; the ordered–shuffled interval includes zero.	Current-state inputs support opportunity learning.
Encoder comparison 5 paired seeds	TCN Town10HD AUC is 0.4616 ($+0.2754$ over Random-k); the gain is not reproduced on Town07.	Encoder effects depend on the domain.
Target comparison 5 paired seeds	Broad 2 s supervision exceeds narrow 0.5 s and full-objective training on UWC@0.5 s.	Broader supervision improves narrow-window placement.
Matched search 5 methods, 5 seeds	TCN and broader-horizon GRU improve mean equal-domain AUC over the canonical GRU; adjusted contrasts pass 0.05.	Matched search favors TCN and broader supervision.

6.2 Allocation Diagnostics

An online allocator commits using present and past information and its remaining budget. We compare offline exact-k placement, causal realized-k calls, and offline matched-k placement at the realized call count (Appendix D.2). At a nominal 2% budget, the evaluated causal allocator retains about 60–65% of offline UWC in four development Towns and 0–16% in Town10HD and Town07. We also evaluate an offline reservation probe that uses known future opportunities while keeping the opportunity scores fixed. On the development Towns, 116 of 351 frozen-score calls at 2% fall outside useful windows. Future-aware reservation increases mean UWC-AUC from 0.5931 to 0.7858 across five budgets (Appendix D.3).

7 Discussion and Conclusion

This study examines event-relative opportunity prediction for the temporal placement of reasoning calls under fixed budgets. On frozen trajectories, development-selected models cover more distinct

pre-event windows than the evaluated baselines at identical invocation counts. The offline comparison separates ranking quality from the additional decisions required to commit limited calls online.

The learning comparisons distinguish the problem formulation from its reference implementation. Current-state training, alternative encoders, and broader supervision can produce strong rankings. Matched comparisons favor current-state inputs over ordered histories in aggregate coverage and show that encoder performance varies across evaluation domains. With evaluation fixed to half-second windows, broad two-second supervision yields higher mean UWC-AUC than half-second supervision and the full reference objective. These findings identify representation and supervision as consequential choices for opportunity prediction.

Deployment also requires requests with relevant context to return in time. A backend with latency ℓ_t^R returns at $t + \ell_t^R$, which must precede the event onset for pre-onset availability. In the four-event diagnostic, longer lead times enabled timely returns but did not improve the joint consistency-and-timeliness endpoint (Appendix E). A natural extension is to study the closed-loop setting by jointly incorporating opportunity scores, online budget allocation, backend occupancy, and response latency, together with comparisons to adapted anticipation and invocation systems.

A call budget determines how many requests are available, but event-relative placement determines which opportunities those requests can reach. With a shared constrained selector, predictive opportunity scores can place the same number of reasoning calls inside more distinct pre-onset event windows.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Matthew Budd, Bruno Lacerda, and Nick Hawes. Stop! Planner time: metareasoning for probabilistic planning using learned performance profiles. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 20053–20060, 2024.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

Ankush Desai, Shromona Ghosh, Sanjit A. Seshia, Natarajan Shankar, and Ashish Tiwari. SOTER: A runtime assurance framework for programming safe robotics systems. In *Proc. of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proc. of the Conference on Robot Learning (CoRL)*, 2017.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proc. of the Conference on Robot Learning (CoRL)*, volume 205, pp. 1769–1782. PMLR, 2023.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T. Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as I can, not as I say: Grounding language in robotic affordances. In *Proc. of the Conference on Robot Learning (CoRL)*, volume 205, pp. 287–318. PMLR, 2023.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 7036–7050, 2024.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023.

Jun Liu, Pu Zhao, Zhenglun Kong, Xuan Shen, Peiyan Dong, Fan Yang, Lin Cui, Hao Tang, Geng Yuan, Wei Niu, Wenbin Zhang, Xue Lin, Gaowen Liu, Yanzhi Wang, and Dong Huang. When should a robot think? resource-aware reasoning via reinforcement learning for embodied robotic decision-making. *arXiv preprint arXiv:2603.16673*, 2026.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 119, pp. 7076–7087. PMLR, 2020.

Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Post-hoc estimators for learning to defer to an expert. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2025.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, 2021.

Stuart J. Russell and Eric H. Wefald. *Do the Right Thing: Studies in Limited Rationality*. MIT Press, 1991. ISBN 9780262181440.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. doi: 10.52202/075280-2997.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Danbing Seto, Bruce H. Krogh, Lui Sha, and Alongkrit Chutinan. The Simplex architecture for safe on-line control system upgrades. In *Proc. of the American Control Conference (ACC)*, volume 6, pp. 3504–3508, 1998.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2025.

Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident DB. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3521–3529. IEEE, 2018.

Sebastian Trimpe and Dominik Baumann. Resource-aware IoT control: Saving communication through predictive triggering. *IEEE Internet of Things Journal*, 6(3):5013–5028, 2019.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. SkipNet: Learning dynamic routing in convolutional networks. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

A Data, Training, and Evaluation Protocols

This appendix presents the data, training settings, evaluation protocols, and supplementary results for the main paper. Unless stated otherwise, coverage uses a $0.5\,\mathrm{s}$ pre-onset window, a $0.5\,\mathrm{s}$ cooldown, and budgets $\{0.5, 1, 2, 5, 10\}\%$.

A.1 Data Roles, Cohorts, and Event Labels

Town01 supplies training data, and Town03 selects epochs. Town05/Town06 are development domains. The primary and fresh Town10HD evaluations use separate scenario sets and separate selected checkpoints. Town07 selects the final model for the fresh evaluation; it is therefore a development domain for that procedure. The later learning comparisons reuse the previously examined Town07/Town10HD populations.

The primary collection made one attempt for each of 80 planned slots. Eligibility was fixed from engineering records before scientific scoring: a trajectory had to complete 300 simulation ticks and pass checks for event onset, sampling cadence, input consistency, BEV validity, and shadow-only execution. Ten runs exited normally after 243–296 ticks; seven infrastructure failures recorded zero ticks. The remaining 63 trajectories contain 63 events and 8,253 candidate ticks. The fresh cohort contains 72 complete episodes, 12,822 ticks, and 18 protocol-defined clusters of four episodes. These collections share CARLA and the base policy, with scenario sets organized by Town.

Events are derived from locked base-policy shadow traces and include priority conflicts, hard-braking leaders, pedestrian crossings, red-light approaches, and vehicle crossings/cut-ins. The onset is excluded from the target window in Equation 1. Training assigns an event to an example, whereas UWC counts distinct evaluation windows reached by a call set. Calls are classified as useful pre-onset, too early, boundary, late/post, or unmatched. Scorer inputs consist of the causal observation history and current BEV; event identities, onsets, future states, and evaluator labels are used for supervision and evaluation. Fail-open recovery calls, where enabled, are accounted for separately.

A.2 Reference Model and Training

The reference model encodes each causal history window independently, resetting the GRU state between windows. Its final GRU state and current BEV embedding form the shared representation. Table 2 gives the architecture and optimization settings. Only the TTE head supplies the deployed score: the probabilities of its first two classes are summed as in Equation 4. The binary window loss acts on the same head through Equation 5.

Table 2: Reference scorer and training settings. Loss coefficients multiply the five terms in Equation 6; class weights operate within those terms. Matched comparisons retain this recipe unless their subsection specifies a change.

Component	Setting
Structured history	20 steps $\times$ 17 features; GRU hidden size 64
Current BEV	$3 \times 64 \times 64$; float conversion and division by 255
BEV encoder	CNN channels $3 \to 8 \to 16 \to 24$, global average pooling, 32-dimensional projection
Shared representation	96 dimensions after concatenation
Output heads	Four TTE classes, four phase classes, three horizon outputs, four urgency classes
Loss coefficients	TTE 1.0; phase 0.5; binary window 0.5; horizon 0.1; urgency 0.1
Within-loss weights	TTE $(1, 1, 1, 1/3)$; phase $(1/3, 1, 1, 1)$; binary positive weight 2; unweighted horizon/urgency
Batch sampling	96 examples: 24 per timing group, sampled with replacement
Training length	25 epochs $\times$ 40 updates = 1,000 updates
Optimizer	AdamW; learning rate 10^{-3}; weight decay 10^{-2}; gradient-norm clipping at 5

The four sampling groups are useful, too early, boundary/post-onset, and off-event examples. Histories retain their frozen feature-cache representation. Matched conditions replay the same seed-indexed

batch plan. After each epoch, the locked Town03 selector is evaluated at 2% and $0.5\,\mathrm{s}$ cooldown, requiring call parity. Epochs are selected lexicographically by UWC, useful-call precision, narrow-window frame AUPRC, and earlier epoch. The fresh study adds a separate Town07 model-selection stage (Appendix B.2); matched development search instead selects one configuration using Town03 (Appendix C.4).

A.3 Integer Budgets and the Shared Selector

For eligible-tick counts n_j, both Town10HD evaluations set $K_\rho = \lfloor \rho \sum_j n_j \rfloor$. Each trajectory first receives $k_j = \lfloor \rho n_j \rfloor$; the remaining calls go to descending fractional remainders. The primary protocol balances remainder ties across frozen role, family, cluster, and weather strata, with SHA256 as the final tie-break. The fresh protocol uses SHA256 of episode identity for remainder ties. The resulting quota manifests are fixed before score access and shared across methods, with calls allocated separately to each trajectory.

Algorithm 1 applies to PRS and learned/risk score baselines. Reactive padded-k retains cooldown-compatible reactive-edge calls, truncating them by current risk if necessary and otherwise filling from the remaining risk ranking. Random-k and Periodic-k use randomized and phase-based proposals under the same constraints. The known-window Exact-DP oracle uses event labels to reconstruct a candidate-constrained coverage upper bound.

Algorithm 1 Score-ranked selection under a fixed trajectory quota

Require: Eligible ticks $\mathcal{T}_j$ with scores; quota k_j; cooldown τ_c
$A_j \leftarrow \varnothing$
Sort $\mathcal{T}_j$ by descending score, then ascending frame index
for tick t in this order **do**
  if $|A_j| = k_j$ **then**
    break
  end if
  if $|t - u| \geq \tau_c$ for every $u \in A_j$ **then**
    $A_j \leftarrow A_j \cup \{t\}$
  end if
end for
Check quota fulfillment and cooldown; report any shortfall
return A_j in chronological order

The scan makes a single pass over score-ranked ticks, retaining cooldown-compatible candidates until the quota is filled or the candidate list is exhausted. All reported primary and fresh schedules fulfill their quotas. Scores use causal inputs, and the offline selector ranks the completed trace. Dataset-specific validation and test quota rules are retained: for example, the target audit uses per-trajectory maximum-one-or-floor rounding on Town03 and global Hamilton allocation at test time.

A.4 Budget Summaries and Statistical Units

For coverage u_j at budgets $\rho_1 < \cdots < \rho_J$, normalized log-budget area is

$$\mathrm{AUC}_{\log} = \frac{\sum_{j=1}^{J-1} (\log \rho_{j+1} - \log \rho_j)(u_{j+1} + u_j)/2}{\log \rho_J - \log \rho_1}. \tag{8}$$

The integration spans the five positive budgets above, from 0.5% to 10%. The primary 2% UWC contrast and the fresh AUC contrast concern different estimands and cohorts. Random and periodic entries may average repeated schedules or phases. The future-aware reservation probe is reported separately using its archived mean UWC-AUC summary (Appendix D.3).

The primary paired bootstrap resamples 35 groups defined by $(\text{event family}, \text{map anchor}, \text{weather})$: 28 groups contain two trajectories, and seven contain one, spanning nine map anchors. Each of 10,000 trials samples 35 groups with replacement, retains all their events with multiplicity, and computes the event-weighted paired difference. The 2.5th and 97.5th percentiles form the 95% interval. Fresh

comparisons resample the 18 event-family/geometry-anchor clusters, preserving paired methods and budgets. Secondary fresh intervals are pointwise and unadjusted for multiplicity.

Learning studies use their specified paired seeds and spatial units, as detailed in Appendix C. Town07 and primary Town10HD provide two and four conservative physical components in those analyses. Uncertainty summaries follow the resampling units specified for each study.

A.5 Supplementary Records

The supplementary numerical records include primary and fresh evaluation summaries, per-seed encoder and supervision curves, matched-search outputs, allocation sensitivity, and request-level lead-time data. Matched-search records contain configuration means, selected epochs for 25 models, and 250 evaluation cells (Appendix C.4). The lead-time diagnostic provides 21 archived RGB inputs and request-level timing and outcome records (Appendix E).

B Fixed-Budget and Fresh-Scenario Results

B.1 Primary and Fresh Budget Sweeps

Table 3 provides the numeric frontier for the primary Town10HD cohort. At 2%, all methods make 165 calls; the selected PRS covers 23 of 63 windows. Its paired UWC difference from Random-k is $+0.2671$ with interval $[0.1264, 0.4097]$. The oracle measures attainable window coverage under the same budget, candidate, and cooldown constraints.

Table 3: Primary-cohort UWC@$0.5\,\mathrm{s}$. All methods use the listed call counts and a $0.5\,\mathrm{s}$ cooldown.

Budget (%)	0.5	1	2	5	10
Calls	41	82	165	412	825
Selected PRS	0.0794	0.1905	0.3651	0.5873	0.8571
Random-k	0.0238	0.0486	0.0980	0.2474	0.4936
Periodic-k	0.0205	0.0435	0.0954	0.2470	0.5016
History-MLP	0.0000	0.0000	0.0635	0.1905	0.4921
Current-risk/TTC	0.0000	0.0000	0.0000	0.0000	0.3016
Reactive padded-k	0.0000	0.0000	0.0000	0.0000	0.3016
Known-window oracle	0.6508	1.0000	1.0000	1.0000	1.0000

The fresh evaluation uses a separately selected checkpoint and new scenarios. Table 4 reports results across all budgets. Its AUC advantage over Random-k is nonnegative in 14 of 18 protocol clusters. Paired comparisons with other baselines appear in Table 5. At 2%, the PRS-minus-Reactive UWC difference is $+0.1944$, with a paired interval of $[-0.0556, 0.4306]$.

Table 4: Fresh Town10HD coverage under exact call parity. Random entries average the locked repetitions. Selection precedes fresh test-score access.

Budget	Calls	Selected PRS	Random-k	Difference
0.5%	64	0.0833	0.0250	$+0.0583$
1%	128	0.2083	0.0493	$+0.1591$
2%	256	0.3194	0.1007	$+0.2187$
5%	641	0.6250	0.2478	$+0.3772$
10%	1282	0.8056	0.4983	$+0.3072$

Table 5: Fresh selected-PRS minus baseline differences. Brackets are pointwise 95% paired protocol-cluster bootstrap intervals. AUC uses all five budgets; the 2% comparison uses 256 calls per method. Additional baseline contrasts are secondary.

Comparator	ΔUWC-AUC [95% CI]	ΔUWC at 2% [95% CI]
History-MLP	$+0.2285$ $[0.0966, 0.3657]$	$+0.1944$ $[0.0278, 0.3750]$
Reactive padded-k	$+0.2616$ $[0.0324, 0.4683]$	$+0.1944$ $[-0.0556, 0.4306]$
Random-k	$+0.2392$ $[0.1159, 0.3690]$	$+0.2187$ $[0.0655, 0.3851]$

B.2 Town07 Selection for Fresh Evaluation

For the fresh evaluation, ten independent seeds are trained on Town01, with epochs selected on Town03. The final checkpoint is selected using the Town07-only development rule before evaluation on fresh Town10HD scenarios. Complete per-seed evaluation records are retained in the supplementary numerical records.

B.3 Sensitivity to the Evaluation Window

Table 6 changes only the window used to evaluate the same primary call sets. It measures the timing tolerance of the selected calls. Coverage is nondecreasing with window width for a fixed call set. Training targets, evaluation windows, and response deadlines are specified separately.

Table 6: Window sensitivity with fixed calls: primary cohort, 165 calls, $0.5\,\mathrm{s}$ cooldown. Model checkpoints and call sets remain fixed.

Method	$\Delta = 0.25$ s	$\Delta = 0.5$ s	$\Delta = 1.0$ s
Selected PRS	0.1429	0.3651	0.4762
Random-k	0.0553	0.0980	0.1936
Periodic-k	0.0552	0.0954	0.2002
History-MLP	0.0635	0.0635	0.1429
Current-risk / Reactive padded-k	0.0000	0.0000	0.0000

The archived sensitivity records use separate one-factor sweeps for window width and cooldown. Seven rows at $1\,\mathrm{s}$ cooldown carry nominal budget-cap infeasibility flags, with zero reported hard-constraint violations.

C Controlled Learning Comparisons

C.1 Frozen Input Interventions and Matched Retraining

The frozen intervention changes only the history supplied to the selected scorer: ordered history, a current-repeat sequence, reversed past, or a locked permutation. The checkpoint, candidates, call counts, cooldown, and evaluator are fixed; scores and call times are recomputed. The reported equal-domain, equal-budget paired-bootstrap differences favor ordered history over current-repeat by $+0.1136$ ($[0.0733, 0.1523]$), reversed past by $+0.0814$ ($[0.0460, 0.1164]$), and permuted past by $+0.0895$ ($[0.0651, 0.1146]$). On Town07, ordered-minus-current-repeat effects are positive at three of five budgets (Table 7); the prespecified criterion requires nonnegative effects at four of five budgets.

Table 7: Town07 frozen-history intervention. All entries are UWC at the same exact-call budget.

Budget	Ordered	Current-repeat	Reversed	Permuted
0.5%	0.0894	0.1545	0.1138	0.0650
1%	0.1707	0.2033	0.2276	0.1301
2%	0.3089	0.2195	0.3252	0.2276
5%	0.4715	0.2764	0.4390	0.3740
10%	0.5528	0.4065	0.5203	0.4878

A separately reported three-seed experiment holds the architecture, data, optimizer, training steps, and checkpoint rule fixed across CURRENT-ONLY, ORDERED, and SHUFFLED training. ORDERED minus CURRENT-ONLY is -0.095509 (95% interval $[-0.153944, -0.036732]$); ORDERED minus SHUFFLED is -0.049860 ($[-0.124756, 0.019310]$). Current-only therefore has higher aggregate coverage under this recipe. Frozen interventions measure the selected checkpoint's input sensitivity; matched retraining compares learning conditions across seeds.

C.2 Matched-Objective Encoder Comparison

GRU, causal TCN, causal Transformer, and current-state models share the 20×17 input schema, current BEV encoder, output heads, TTE-derived score, Town01 training data, Town03 epoch rule, balanced sampling, and 1,000 updates. Five paired seed-indexed batch plans produce 20 completed runs. Table 8 reports means over all five seeds, parameter counts, and median batch-1 GPU p95 latencies. Full seed/budget curves remain in the supplementary records.

Table 8: Matched-objective encoders. Parentheses give mean AUC difference from Random-k and the number of positive paired seeds. Random-k mean AUC is 0.1643 on Town07 and 0.1861 on Town10HD.

Model	Params	GPU p95 (ms)	Town07 AUC (Δ; +seeds)	Town10HD AUC (Δ; +seeds)
GRU	23,447	1.26	0.2137 ($+0.0494; 3/5$)	0.2081 ($+0.0220; 3/5$)
TCN	24,151	1.93	0.1581 ($-0.0063; 2/5$)	0.4616 ($+0.2754; 4/5$)
Transformer	23,591	2.83	0.1657 ($+0.0014; 3/5$)	0.1188 ($-0.0673; 1/5$)
Current state	24,759	1.32	0.2695 ($+0.1052; 4/5$)	0.3187 ($+0.1326; 4/5$)

TCN exceeds Random-k at all five mean Town10HD budget points. Its AUC difference is $+0.2754$, with marginal crossed seed–route interval $[0.0700, 0.4771]$; the Holm-adjusted p value across 12 prespecified domain–comparator contrasts is 0.0876. Against current-state training, the difference is $+0.1428$ with interval $[-0.0087, 0.2938]$. Current-state training has the highest mean AUC on Town07. The intervals are marginal, and the Holm-adjusted tests use the 12 prespecified contrasts; route-level summaries use the two/four physical components.

C.3 Matched Supervision Targets

Broad two-second classification (A), half-second window classification (B), and the full PRS objective (C) use the same GRU/BEV model, paired seeds 202609071–202609075, sampling plans, and 1,000 updates. Each model has 23,447 parameters; A and B each have 22,380 trainable parameters, whereas C has 23,447. Town03 selects epochs by the rule in Appendix A.2, and all five paired seeds are evaluated.

The common training-source amendment contains 271 merged onsets and 3,757 retained rows after jointly excluding 75 right-censored rows. Shared labels are reconstructed consistently across objectives; C retains the legacy horizon/urgency auxiliaries. Evaluation uses the previously examined Town07 (81 trajectories, 123 events) and primary Town10HD (63 trajectories, 63 events) populations, with common Hamilton quotas and half-second windows.

Broader supervision yields higher mean UWC-AUC in both domains. The mean difference between the full objective and half-second classification is $+0.001794$, with a 95% interval of $[-0.057686, +0.065844]$. Seed intervals quantify training variation on these fixed populations. All three objectives are evaluated using the same half-second event-window coverage metric. Supplementary records retain all seed/budget cells, domain contrasts, and physical-component sensitivity analyses.

C.4 Matched Development Search

Five methods each use seeds 202609091–202609095 and four configurations: learning rate $\{3 \times 10^{-4}, 10^{-3}\}$ crossed with weight decay $\{0, 10^{-2}\}$. Each of the 100 runs receives 25 epochs of 40 updates and batch size 96. Epoch selection follows the Town03 rule above. For each method, the

Table 9: Matched supervision targets. Top: five-seed summaries of equal-domain normalized log-budget AUC. Bottom: paired-seed contrasts with 95% percentile intervals from 10,000 draws on fixed test populations. B–A is primary; the other contrasts are secondary.

Objective	Mean AUC	Median AUC	Seed SD
A: Broad 2 s BCE	0.281452	0.223245	0.143308
B: Window 0.5 s BCE	0.117998	0.116155	0.028440
C: Full PRS	0.119792	0.083678	0.074068
Contrast	**Mean difference**	**95% interval**	**Positive seeds**
B–A	-0.163455	$[-0.291391, -0.061842]$	0/5
C–B	$+0.001794$	$[-0.057686, +0.065844]$	3/5
C–A	-0.161660	$[-0.231956, -0.111016]$	0/5

configuration with the highest mean Town03 normalized log-budget AUC across its five selected-seed checkpoints is retained, breaking ties by ascending configuration ID. All five seeds of the chosen configuration are evaluated, giving 25 models.

Configurations c_0, c_1, c_2, c_3 enumerate the above Cartesian product in ascending learning-rate then weight-decay order. Selection chooses c_0 for GRU/PRS, TCN/PRS, and GRU/broad 2 s, c_1 for Current/PRS, and c_3 for Transformer/PRS. The configuration scores, selected epochs, and all 250 evaluation cells are retained in the supplementary records. Both evaluation domains were previously examined.

Table 10: Matched-search results, retaining all five selected-configuration seeds per method. Top: mean normalized log-budget AUC. Bottom: GRU/PRS minus comparator, with unadjusted percentile intervals and separately Holm-adjusted p values. Several intervals exclude zero, whereas the corresponding adjusted p values do not. The adjusted values govern the multiplicity-controlled conclusions. Positive seeds favor GRU/PRS.

Method	Town07	Town10HD	Equal-domain
GRU/PRS	0.156053	0.044102	0.100077
Current/PRS	0.166140	0.299600	0.232870
TCN/PRS	0.164996	0.315866	0.240431
Transformer/PRS	0.275864	0.153445	0.214654
GRU/broad 2 s	0.353233	0.324368	0.338801

Comparator	Difference	95% interval	Holm p	Positive
Current/PRS	-0.132793	$[-0.279691, -0.006250]$	0.116588	0/5
TCN/PRS	-0.140353	$[-0.226277, -0.075296]$	0.004800	0/5
Transformer/PRS	-0.114577	$[-0.234014, 0.009917]$	0.116588	1/5
GRU/broad 2 s	-0.238723	$[-0.380660, -0.027912]$	0.038096	0/5

The primary statistic averages event-weighted AUC equally over domains and then over seeds. The 10,000 paired bootstrap draws (seed 202609099) resample seeds and within-domain physical components. Selected configurations remain fixed. Percentile intervals are unadjusted; two-sided centered-bootstrap p values use a plus-one correction and Holm adjustment over four primary contrasts. The bootstrap summaries condition on the selected configurations and the two/four physical components.

The TCN and broad-supervision contrasts in Table 10 pass the adjusted 0.05 threshold. These results summarize the matched-search models; Appendix B reports the selected-checkpoint evaluations.

D From Opportunity Ranking to Online Calls

D.1 Local Scores and Coverage

For fixed windows, $F_\Delta(A) = N\ \mathrm{UWC}\,@\Delta(A)$ has marginal gain

$$F_\Delta(A \cup \{t\}) - F_\Delta(A) = \sum_i \mathbf{1}[t \in W_i(\Delta)]\mathbf{1}[A \cap W_i(\Delta) = \varnothing]. \tag{9}$$

An ideal immediate-coverage score would estimate the conditional expectation of this quantity given observations and previous calls. PRS predicts assigned-event window membership from the causal observation history; the selector applies the budget and cooldown constraints. When $\Delta = \tau_c = 0.5\,\mathrm{s}$, each half-open window can contain at most one feasible call. Each selected tick consumes quota and excludes candidates within its cooldown interval.

D.2 Offline and Causal Evaluation Contracts

Offline exact-k ranks the full trace and makes K_ρ calls. Causal realized-k commits using present/past information and remaining budget, making $\widehat{K} \le K_\rho$ calls. Offline matched-k reruns offline selection at the realized count $\widehat{K}$.

At 2% on Town10HD, the causal record reports `schedule_ceiling_k`=126 alongside a nominal count of 165 and 32 realized calls. The matched-k comparison uses 32 calls, matching the aggregate causal call count.

Table 11: Nominal-2% conversion from offline ranking to the evaluated causal allocator. Retention is causal UWC divided by offline exact-k UWC. Matched-k uses the realized causal call count. All rows have zero reported budget and cooldown violations.

Domain	Exact-k UWC	Causal call rate	Causal UWC	Matched-k UWC	Retention
Development	0.3146	1.42%	0.1984	0.2312	63.1%
Town01	0.3087	1.28%	0.1870	0.2087	60.6%
Town03	0.3443	1.60%	0.2217	0.2689	64.4%
Town05	0.2804	1.45%	0.1824	0.2264	65.1%
Town06	0.3596	1.26%	0.2193	0.2193	61.0%
Town10HD	0.3651	0.39%	0.0000	0.0159	0.0%
Town07	0.3089	1.79%	0.0488	0.2846	15.8%

Coverage retention is roughly 60–65% in four development Towns, 0% on Town10HD, and 15.8% on Town07 (Table 11). Town10HD makes 32 calls, all unmatched under the frozen association. The archive retains all 105 domain/budget/contract cells in `sensitivity/sensitivity_table.csv` (`axis=window_pre_s`, `value=0.5`).

D.3 Future-Aware Reservation

With scorer, candidates, cooldown, and evaluator frozen on the development Towns, 116 of 351 calls at 2% lie outside useful windows. A known-future reservation probe increases the recorded mean UWC-AUC across five budgets from 0.5931 to 0.7858, a gain of 0.1927. This offline diagnostic reserves calls using known future opportunities. The probe uses 351 reference calls at 2%; the default development exact-k contract in Table 11 uses 355. The AUC figures above reproduce the archived summary for this probe.

E Request Lead Time and Response Availability

E.1 Diagnostic Protocol

Four previously exposed engineering trajectories provide one lead-braking, pedestrian-crossing, vehicle-crossing/cut-in, and traffic-light event. Qwen2.5-VL-7B-Instruct (Bai et al., 2025) runs on an

RTX4090 with a maximum of 200,704 pixels, an output-token cap of 64, and a fixed prompt and strict parser. Target lead times are $\{0.25, 0.5, 1, 1.5, 2, 3\}$ s. The 24 planned slots yield 21 archived RGB inputs: four events at each of the first five lead times and one vehicle-crossing event at 3 s. Each available slot receives one attempt in fixed order; the strict parser processes the original output. The backend receives one RGB image and the fixed prompt. Calls are evaluated in an offline response diagnostic.

Times are chosen retrospectively using recorded onsets. Each trial starts with an idle inference slot and sets a monotonic-clock deadline using the actual frame-to-onset interval at $1\times$ speed. The latest available frame at or before the target is used; the protocol allows approximately 0.1 s of sampling lag, while the maximum observed lag in the 21 records is approximately 0.05 s. Response elapsed time includes admission journaling, IPC, preprocessing, inference, and strict parsing, but excludes model loading, original sensor capture, and deployment-network delay. The first inference request is retained.

E.2 Content, Timeliness, and Joint Outcomes

Let Q indicate schema-valid agreement with the frozen simulator event/action rubric, T a parsed response strictly before the actual deadline, and $C = QT$ their conjunction. All 21 outputs are reported as schema-valid. Table 12 retains every planned lead-time bin and identifies the sole joint success.

Table 12: Complete lead-time diagnostic. Rates use available events as denominators. The first five rows use the same four events, whereas the 3 s row contains only vehicle crossing. All nonzero Q values belong to lead braking.

Lead (s)	Available	Missing	Q	T	C	Joint-success event
0.25	4	0	1/4	0/4	0/4	None
0.5	4	0	1/4	0/4	0/4	None
1	4	0	1/4	0/4	0/4	None
1.5	4	0	1/4	4/4	1/4	Lead braking
2	4	0	0/4	4/4	0/4	None
3	1	3	0/1	1/1	0/1	None

The prespecified paired contrast $C(2\,\mathrm{s}) - C(0.5\,\mathrm{s})$ is zero for each event. The lead-braking event achieves joint success at 1.5 s. Response elapsed time has a median of 1.270795 s, linearly interpolated p95 of 1.328019 s, and range 0.974653–2.013201 s. At the input times, the source-state summaries report the target light still green, the pedestrian stationary, the crossing vehicle at zero/near-zero horizontal velocity, and the lead vehicle not yet applying its event brake. The supplementary records also include descriptive observations from a single reviewer.

Other backend experiments. The earlier equal-call content pilot has 47,185 unique inputs and 44,835 valid outputs; its historical content-usefulness AUC gain over Current-only is $+0.008887$ ($[0.000000, 0.024742]$), with about one to two correctly covered events per domain and 60.5% of the gain from one route. The content pilot uses a 30 s batch timeout. A separate six-configuration service screen uses 210 requests (16 format-test and 19 window images per configuration, the latter from four events); minimum service time is 1.207 s and retrospective timely-and-consistent coverage is 0/19 per configuration. Service-screen and lead-time measurements are reported separately under their respective input and timing protocols. The archived outcome of the earlier formal closed-loop replication is classified as `negative/null`.

AI Use Statement

Generative AI tools were used for language editing, code assistance, and figure-layout iteration. The authors retained responsibility for experimental design and manually checked all reported numbers, claims, citations, and generated artifacts against frozen local evidence.