PERCEPT-VR++: LEARNING PRE-UPDATE PERCEPTUAL RISK FOR RISK-AWARE VISUAL UPDATE SCHEDULING IN VR

Anonymous authors
Paper under double-blind review

ABSTRACT

A perceptual-risk score for a pending visual update may no longer describe the update after a scheduler changes its attributes. We study pre-update risk prediction conditioned jointly on multimodal behavioral history and the proposed update. PERCEPT-VR++ instantiates this formulation with M2P-CF, using eye dynamics, head motion, scene context, and update attributes. We construct VR-Mask++ through nine independently administered scene studies with 450 participants, collecting 9,000 candidate trials and retaining 8,301 valid multimodal events. Held-out AUROC increases from 0.766 for the strongest evaluated non-M2P baselines to 0.831 for M2P-CF; beta calibration reduces its ECE from 0.109 to 0.043. An independent closed-loop study recruits 55 new participants and records 7,425 candidate policy-event observations. False-safe rate decreases from 0.105 for calibrated prediction-only to 0.059 for current-score Safe-UARS and 0.030 for re-query Safe-UARS, while applied-update coverage rises from 0.231 to 0.265 and 0.280. Current-score and re-query share all frozen policy parameters, differing only in whether risk is re-estimated after update attributes change. Re-query yields a participant-paired FSR difference of -0.029 (95% CI $[-0.053, -0.005]$); both adjacent-policy paired intervals exclude zero, with consistent ordering across scene families. These results support re-estimating risk for the update actually proposed for application, rather than retaining the score of its original specification.

1 INTRODUCTION

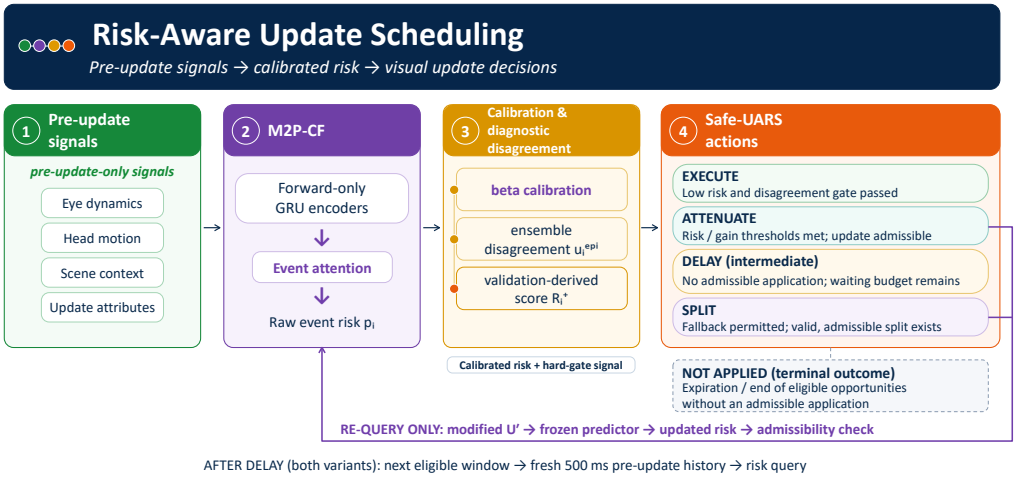

Figure 1: PERCEPT-VR++ overview. The predictor evaluates a pre-update state–update pair. In the re-query variant, modified update attributes are fed back to the frozen predictor before application; after Delay, both variants acquire a fresh pre-update window.

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.

Virtual reality (VR) rendering must balance visual quality, refresh rate, and computation under strict latency constraints. Perceptual and foveated rendering exploit non-uniform visual sensitivity to decide where fidelity can be reduced (Guenter et al., 2012; Patney et al., 2016; Wang et al., 2023). A complementary decision is *when* a pending visual update can be applied without being noticed.

Blink- and saccade-contingent methods provide discrete timing heuristics (Burr et al., 1994; Arabadzhiyska et al., 2017; Saeedpour-Parizi et al., 2024); scene- and attention-aware rendering incorporate richer contextual cues (Fan et al., 2024; Krajancich et al., 2023). Prediction, calibration, and downstream decision making are established building blocks (Guo et al., 2017; Kull et al., 2017; Sun et al., 2023); the complete related work appears in Appendix A.1.

The central learning question is what a perceptual-risk estimate should be conditioned on when the scheduler can modify the pending update. A score for the original update need not describe an attenuated or split update, even when the pre-update behavioral history is unchanged. We therefore treat risk as a property of the pre-update state–update pair, and distinguish predictive discrimination, low-risk ordering, and closed-loop evaluation after update modification.

Collecting evidence for this timing problem requires real users, calibrated eye tracking, synchronized pre-update measurements, controlled update delivery, and time-aligned perceptual responses. VR-Mask++ comprises nine independently administered scene studies with 450 participants, yielding 9,000 candidate trials and 8,301 valid multimodal events. PERCEPT-VR++ uses these data to learn M2P-CF risk scores and supplies them to the Safe-UARS scheduler.

An independent 55-participant closed-loop study compares current-score and re-query scheduling under shared frozen parameters. Re-estimating risk after update modification lowers FSR from 0.059 to 0.030; the participant-paired difference is -0.029 (95% CI $[-0.053, -0.005]$). This tests whether predictive conditioning remains aligned with the update proposed for application.

Our contributions are threefold: (1) a pre-update, update-conditioned perceptual-risk formulation and VR-Mask++, comprising 8,301 valid events from 450 participants across nine scenes; (2) a multimodal instantiation with matched-event regularization, evaluated through input and objective ablations that distinguish aggregate discrimination from low-risk ordering; and (3) an independent closed-loop comparison showing that re-estimating risk after update modification improves the observed detection–coverage tradeoff under otherwise identical frozen policy parameters.

2 Problem Formulation

Event-level perceptual risk. We model each visual update request as an event, rather than as a frame-level rendering artifact or a binary eye-state trigger. For an update event i occurring at time t_i, the online system observes only pre-update information,

$$x_i = \{E_i, H_i, C_i, U_i, G_i\}, \tag{1}$$

where $E_i \in \mathbb{R}^{T \times d_e}$ denotes the eye-dynamics window before the update, including gaze velocity, fixation/saccade/blink indicators, pupil- or openness-derived features when available, and gaze confidence; $H_i \in \mathbb{R}^{T \times d_h}$ denotes head-motion dynamics such as angular velocity, acceleration, and stability; $C_i \in \mathbb{R}^{T \times d_c}$ denotes the local task and scene context, including scene motion, target eccentricity, object density, and contrast; U_i describes the pending visual update, including its type, magnitude, duration, region, and expected rendering utility; and G_i denotes an optional session-covariate slot. In the reported configuration, $G_i = \mathbf{0}$ for every event, and participant identity is not used as an input. The supervision is a user feedback label $y_i \in \{0, 1\}$, where $y_i = 1$ indicates that the user detected the update. Detection is the primary supervision signal. Abruptness supplies an auxiliary ordinal training target when available, whereas discomfort and confidence are retained only for diagnostic analysis; the primary decision variable remains event-level detection risk.

The goal of perceptual risk prediction is to estimate

$$r_\theta(x_i) = P_\theta(y_i = 1 \mid E_i, H_i, C_i, U_i, G_i), \tag{2}$$

using only information available before execution; post-update reactions, task outcomes, and explicit feedback are excluded from predictors.

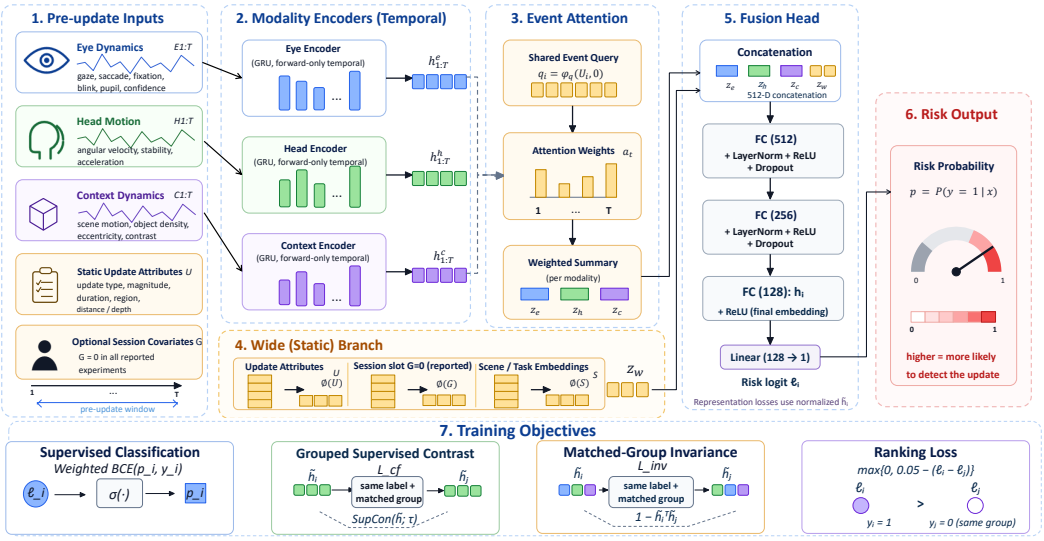

Figure 2: M2P-CF architecture. Forward-only GRUs, event attention, and a static branch feed the fusion network. Grouped supervised contrastive and same-label invariance losses use normalized 128-dimensional embeddings $\tilde{h}_i$; within-group ranking uses raw logits. An auxiliary ordinal abruptness head uses h_i (Appendix A.3). The optional G slot is zero.

Risk under update modification. For a modification $U_i \to U_i'$ evaluated at the same pre-update state, define

$$x_i' = \{E_i, H_i, C_i, U_i', G_i\}.$$

The corresponding risk query is $r_\theta(x_i')$, rather than the original $r_\theta(x_i)$. Current-score scheduling retains the original score, whereas re-query scheduling evaluates the modified input through a new forward pass with frozen model and calibration parameters, without retraining or refitting. After DELAY, both policies instead acquire a fresh pre-update history. This distinction concerns predictive conditioning, not identification of individual causal effects.

Risk-aware rendering scheduling. When a pending update arrives, the renderer must decide not only whether the update is useful, but also whether it is likely to be perceptually low-risk. We define a scheduling state

$$s_t = [E_t, H_t, C_t, U_t, G_t, b_t], \tag{3}$$

where b_t summarizes system state such as pending update queue, delay budget, and rendering load. The scheduler selects an action

$$a_t \in \{\text{EXECUTE}, \text{DELAY}, \text{ATTENUATE}, \text{SPLIT}\}. \tag{4}$$

Safe-UARS chooses actions using calibrated risk thresholds and fixed benefit, delay, and quality proxies. Its utility expression motivates these rules rather than defining an optimization solved online; the complete expression appears in Appendix A.7. A *false-safe* decision applies an update that the participant detects.

3 PERCEPT-VR++

PERCEPT-VR++ consists of M2P-CF, post-hoc calibration with a validation-derived threshold margin and a fixed ensemble-disagreement gate, and the Safe-UARS scheduler (Figs. 1 and 2). The design follows the online restriction in Eq. (2): all prediction inputs are available before executing the pending update.

M2P-CF multimodal predictor. M2P-CF first encodes eye, head, and context streams using modality-specific temporal encoders,

$$H_i^e = f_e(E_i), \quad H_i^h = f_h(H_i), \quad H_i^c = f_c(C_i), \tag{5}$$

where each $H_i^m = \{h_1^m, \ldots, h_T^m\}$ is a sequence of hidden states. Keeping modalities separated before fusion prevents early concatenation from hiding modality-specific failure modes such as gaze tracking instability, head-turn-induced motion, or context-driven masking. In our implementation we use forward-only GRU encoders for all three temporal streams, with the window length and training hyperparameters reported in the supplementary material.

To focus the representation around the pending update, M2P-CF applies event attention. A query vector $q_i = \phi_q(U_i, G_i)$ is derived from the pending-update attributes, with $G_i = \mathbf{0}$ in the reported configuration. For each temporal modality $m \in \{e, h, c\}$,

$$\alpha_t^m = \frac{\exp(q_i^\top W_m h_t^m)}{\sum_{\tau=1}^T \exp(q_i^\top W_m h_\tau^m)}, \qquad z_i^m = \sum_{t=1}^T \alpha_t^m h_t^m. \tag{6}$$

Event attention is one implementation of update conditioning: it lets the temporal summary depend on the proposed update. The static branch also supplies update attributes, including in variants without attention.

Static features enter through a wide branch,

$$z_i^w = \phi_w([U_i, G_i, S_i]), \tag{7}$$

where S_i denotes static scene/task attributes. This branch preserves update magnitude, target eccentricity, distance/depth, and task group. The concatenated feature vector and prediction are

$$z_i = [z_i^e, z_i^h, z_i^c, z_i^w], \qquad \ell_i = f_{\text{fuse}}(z_i), \qquad p_i = \sigma(\ell_i), \tag{8}$$

where ℓ_i is the uncalibrated risk logit and p_i is the raw detection probability. The fusion network maps $z_i \in \mathbb{R}^{512}$ through widths $512 \to 256 \to 128$; its final ReLU embedding $h_i \in \mathbb{R}^{128}$ feeds a linear detection head. The normalized embedding $\tilde{h}_i$ is used for representation regularization.

Matched-event and risk-oriented objectives. The primary loss is binary cross-entropy on the detection label,

$$\mathcal{L}_{\text{cls}} = -w_+ y_i \log p_i - (1 - y_i) \log(1 - p_i), \tag{9}$$

Here w_+ is the training negative-to-positive ratio, fixed for validation and test evaluation; training averages this per-event loss over the mini-batch. To encourage learning across matched pre-update conditions, we train with auxiliary regularizers,

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\text{cls}} &+ \lambda_{\text{cf}} \mathcal{L}_{\text{cf}} + \lambda_{\text{inv}} \mathcal{L}_{\text{inv}} \\ &+ \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{abr}} \mathcal{L}_{\text{abr}}. \end{aligned} \tag{10}$$

Matched-event regularization comprises three terms. Grouped supervised contrastive regularization $\mathcal{L}_{\text{cf}}$ uses normalized embeddings $\tilde{h}_i$: positives share a detection label and a training-only matched-group identifier, while other in-batch events form the comparison set. Same-label invariance $\mathcal{L}_{\text{inv}}$ averages cosine distance $1 - \tilde{h}_i^\top \tilde{h}_j$ over valid matched pairs. Within-group ranking $\mathcal{L}_{\text{rank}}$ applies a margin-0.05 hinge to raw detected–undetected logit differences. Grouping keys use pre-update scene, update, eccentricity, and eye-state variables; detection labels select the relevant pairs. Groups are constructed within training only, and unmatched events remain in the classification objective. These observational matches regularize prediction rather than identify causal effects. The auxiliary ordinal loss $\mathcal{L}_{\text{abr}}$ supervises available abruptness ratings with $\lambda_{\text{abr}} = 0.05$; discomfort and confidence have zero loss weight. Appendix A.3 gives exact formulas and pair construction.

Frozen decision scores. Calibration supplies probabilities for thresholded decisions. For an ensemble of K predictors, we compute

$$\bar{p}_i = \frac{1}{K} \sum_{k=1}^K \sigma(\ell_i^{(k)}), \qquad u_i^{\text{epi}} = \text{SD}_{k=1}^K \left[\sigma(\ell_i^{(k)}) \right], \tag{11}$$

where u_i^{epi} is diagnostic across-member disagreement. Post-hoc calibration methods are compared on validation predictions; the final model uses beta calibration directly on the raw ensemble probability,

$$p_i^{\text{cal}} = g_\beta(\bar{p}_i), \tag{12}$$

with g_β fitted on validation only. For validation event j, let $s_j = [y_j - p_j^{\text{cal}}]_+$. Its higher empirical $(1-\alpha)$-quantile $q_{1-\alpha}$ defines the scheduling score

$$R_i^+ = \min(1, p_i^{\text{cal}} + q_{1-\alpha}) \tag{13}$$

with $q_{1-\alpha} = 0.06752$ and $\alpha = 0.10$. This global offset shifts thresholds without changing the risk ranking, apart from clipping ties. The hard-gate quantity is

$$u_i = \max\left\{u_i^{\text{epi}}, \left|R_i^+ - p_i^{\text{cal}}\right|\right\}. \tag{14}$$

The gate is hard: u_i is not added to R_i^+ and no λ_u is used. Since $|R_i^+ - p_i^{\text{cal}}| \le q_{1-\alpha} < u_{\max} = 0.08$, it tests $u_i^{\text{epi}} \le 0.08$. When the validation order-statistic threshold $\epsilon(c) < 1$, the same offset enters the score and threshold, so

$$R_i^+ \le \epsilon(c) \iff p_i^{\text{cal}} \le p_{(k)}^{\text{cal}}, \qquad \epsilon(c) = p_{(k)}^{\text{cal}} + q_{1-\alpha}.$$

Here $k = \text{round}(cN_{\text{val}})$ (Appendix A.7). Calibration, the validation-derived offset, and the disagreement gate are implementation components of the frozen scheduler; we do not attribute the re-query improvement to a new calibration method or a certified risk bound.

Safe-UARS scheduling. Safe-UARS converts calibrated risk into rendering actions. For an unexpired candidate at an eligible window, the scheduler queries M2P-CF, obtains $(p_i^{\text{cal}}, u_i, R_i^+)$, and evaluates the update utility. The following describes action selection; evaluation counts applications actually presented to the participant:

> **Safe-UARS.** If $R_i^+ \le \epsilon_s$ and $u_i \le u_{\max}$, EXECUTE the update. Otherwise, form an attenuated update when $R_i^+ \le \epsilon_a$ and the gain proxy is at least 0.05; ATTENUATE if admissible under the current-score or re-query rule. If no application is admissible and waiting remains within the budget of $B = 2$ candidate windows, DELAY to the next eligible window and re-query using a fresh 500-ms pre-update window. A permitted fallback applies SPLIT only when a valid, admissible split exists.

EXECUTE, ATTENUATE, and SPLIT are terminal application actions, whereas DELAY is an intermediate transition. A pending candidate that does not reach an admissible application before expiration or the end of its eligible decision opportunities is recorded as *Not applied*; in the closed-loop study, no perceptual label is imputed for it.

Here $\epsilon_s < \epsilon_a$ separates execute and attenuation thresholds, selected from validation score order statistics using direct-execution and action-budget targets $C_s = 0.16$ and $C_a = 0.24$, respectively; the hard cutoff is $u_{\max} = 0.08$. These settings and utility weights are frozen before evaluation, with no per-scene or per-user tuning (Table S20). In logged replay, ATTENUATE and SPLIT use the current $(p_i^{\text{cal}}, u_i, R_i^+)$ and fixed utility proxies without re-scoring hypothetical updates. The gate uses Eq. (14) without an additive uncertainty penalty. The replay comparison adds an ER-matched prediction-only point to separate execution coverage from policy selection.

4 VR-MASK++ DATASET AND EXPERIMENTAL SETUP

Large-scale multi-scene data collection. VR-Mask++ models the perceptual detectability of VR visual updates across nine scenes with distinct task demands and visual complexity. We independently recruited 50 participants for each scene, yielding 450 calibrated participant–scene sessions rather than repeatedly exposing one small cohort to all content. Each participant completed 20 trials balanced across five scheduling conditions, producing 9,000 candidate trials. After removing catch/no-update trials, 8,301 valid events remain, including 2,809 detected and 5,492 non-detected updates.

Every session required headset fitting, eye-tracker calibration, controlled update delivery, and synchronized recording of eye dynamics, head motion, scene context, update attributes, and perceptual feedback. Repeating this protocol across nine separately administered scenes provides cross-scene, cross-task, and cross-participant variation unavailable in a single-task or synthetic setting. Participant-disjoint groups yield 5,828/1,254/1,219 train/validation/test samples with no overlap; detailed distributions are in the supplement.

Tasks and visual updates. The nine scenes span controlled detection, natural search, and reading/UI tasks. Updates include LOD transitions, blur/sharpness and resolution changes, contrast/luminance changes, and depth/position micro-shifts. Appendix Fig. S6 and Tables S4–S5 give the complete scene and update taxonomy.

Online-available feature construction. The online feature table contains only information available before the pending update: eye dynamics (gaze velocity, fixation/saccade/blink indicators, pupil or openness features when available, and gaze confidence), head-motion dynamics, local scene context, and update attributes such as type, magnitude, duration, and region. To prevent leakage, we exclude post-update gaze reactions, response time, task outcomes, explicit feedback, participant identity, and the logging-condition field. The optional session slot is disabled, so $G = \mathbf{0}$ in all reported experiments; participant identity is used only for split construction and evaluation metadata. Missing numerical values are imputed with training-set medians and accompanied by missingness indicators, while categorical scene, task, and update variables are one-hot encoded. Split groups are stratified across logging conditions when possible. Logged replay preserves recorded pre-update states and labels but cannot reproduce action-induced changes in later gaze, head motion, or queue state; it is therefore used for logged-event selection diagnostics, with action effects evaluated in the independent closed-loop study. Dataset construction, preprocessing, and evaluation details are provided in the appendix.

Independent controlled closed-loop evaluation. To test whether replay trends persist when actions alter subsequent state, we recruited 55 new participants disjoint from VR-Mask++. Each completed a within-subject design crossing three policies with three representative scenes and 15 candidates per policy–scene block. Policy orders were near-balanced and scene order was Latin-square or near-balanced, yielding 2,475 candidates per policy and 7,425 policy-event observations. Additional catch/no-update trials were diagnostic and excluded from FSR denominators.

Current-score and re-query Safe-UARS share the same model, calibration, and frozen policy parameters (Table S20). Their only difference is the treatment of modified update attributes $U_i \to U_i'$: current-score retains the original $(p_i^{\mathrm{cal}}, R_i^+, u_i)$, whereas re-query recomputes all three quantities using U_i' before applying ATTENUATE/SPLIT. Both policies acquire a fresh pre-update window after DELAY. All parameters were fixed before collection. The study tests controlled replay-to-live transfer within three scenes on one platform.

Baselines and metrics. We compare eye-state and eye–head rules, saccade-landing prediction, classical tabular models, and tabular or temporal neural predictors. Tables S23–S24 define all variants, their input representations, and model-selection rules.

For replay, let N be the number of candidate events, $a_i \in \{0, 1\}$ indicate that update i is ultimately applied during its eligible decision opportunities, and $y_i \in \{0, 1\}$ denote its recorded detection label. Execute, Attenuate, and Split applications have $a_i = 1$; candidates recorded as Not applied have $a_i = 0$, whether or not they were previously delayed. We define execution ratio (ER), successful update ratio (SUR), and false-safe rate (FSR) as

$$\begin{aligned} \mathrm{ER} &= \frac{\sum_{i=1}^{N} a_i}{N}, \qquad \mathrm{SUR} = \frac{\sum_{i=1}^{N} a_i(1 - y_i)}{N}, \\ \mathrm{FSR} &= \frac{\sum_{i=1}^{N} a_i y_i}{\sum_{i=1}^{N} a_i}. \end{aligned} \tag{15}$$

ER measures the fraction of candidate updates that are ultimately applied, whereas SUR additionally requires that an applied update remain undetected. FSR is the fraction of applied updates that are detected. ER and SUR include attenuated and split updates and measure applied-update coverage,

not rendering speedup or completion of full-amplitude updates. Under these definitions, $\mathrm{SUR} = \mathrm{ER}(1 - \mathrm{FSR})$. For objective and modality ablations, *FS@20* is computed independently for each of five training runs. A separate beta calibrator is fitted to each run's validation predictions and labels, then applied unchanged to its predictions for the same 1,219 held-out events. Each run selects its 244 lowest-risk events ($\lceil 0.2 \times 1{,}219 \rceil$) and computes their detected fraction. We report the arithmetic mean of these five run-wise rates; event selection precedes averaging across runs. This fixed-coverage metric evaluates low-risk ordering before utility-aware actions.

Closed-loop endpoints and inference. In the live evaluation, FSR uses applied updates, whereas ER and SUR use all candidates. Added delay is measured over applied updates, assigning zero to immediate applications. We report pooled rates with participant-bootstrap 95% confidence intervals and prespecified participant-paired bootstrap intervals for the two adjacent-policy ΔFSR contrasts; these paired contrasts are the primary inference. Crude event-level odds ratios are secondary descriptive summaries (Appendix A.6). We report AUROC, AUPRC, and F1 for discrimination, Brier score and ECE for probability quality, and ER, SUR, and FSR for replay evaluation. Participant-bootstrap confidence intervals and detailed closed-loop action, scene, queue, and subjective-result breakdowns are provided in the supplementary material.

5 Experiments

5.1 Can pre-update multimodal signals predict detectability?

Table 1 evaluates participant-disjoint prediction using pre-update inputs. Logistic regression and CatBoost reach AUROC 0.766; M2P-CF reaches 0.831. The main discrimination gain occurs at the M2P-family level: core reaches 0.826 and has the highest AUPRC, while Full has the highest observed AUROC. The 0.005 Full–core difference is descriptive, not evidence of statistically established superiority of the full objective set.

The variants separate temporal aggregation from training objectives (Table S23). M2P-core uses final GRU states without event attention and retains abruptness supervision; M2P-attn adds update-conditioned attention with the same objectives. BCE-only retains the full attention architecture but removes all four auxiliary objectives, whereas Full M2P-CF uses CF, invariance, ranking, and abruptness alongside BCE. Thus, core and BCE-only differ in both architecture and supervision; all four configurations retain the wide branch and use $K = 5$ members.

Table 1: Held-out prediction and calibration. *+Beta* is fitted on validation data; other rows report native probabilities. Native ECE is descriptive and is not the architecture-selection criterion.

Method	**AUROC** ↑	**AUPRC** ↑	**F1** ↑	**Brier** ↓	**ECE** ↓
Blink	.569	.397	.533	.232	.048
Saccade	.595	.451	.517	.235	.076
Eye-state	.530	.437	.533	.230	.039
Eye+Head	.588	.471	.533	.233	.070
Logistic regression	.766	.677	.637	.183	.042
Random forest	.753	.661	.618	.187	.043
XGBoost	.758	.675	.617	.185	.028
CatBoost	.766	.688	.630	.182	.036
Saccade-landing-aware	.712	.612	.601	.196	.049
MLP	.693	.573	.568	.210	.047
Temporal Transformer	.741	.651	.620	.190	.041
FT-Transformer	.762	.680	.629	.184	.033
M2P-core	.826	**.739**	.682	**.152**	.015
M2P-attn	.821	.721	.694	.155	**.012**
M2P-CF raw	**.831**	.731	**.700**	.163	.109
M2P-CF + Beta	**.831**	.731	**.701**	**.152**	.043

Table 1 compares complete prediction pipelines rather than identical-input architectures. We use it to establish predictive feasibility; within-family ablations and the frozen-policy comparison address inputs, objectives, and re-querying. Baselines use event summaries or cached sequences, and the

temporal Transformer has no separate update-attribute vector (Table S24). For M2P-CF, beta calibration reduces ECE from 0.109 to 0.043 and Brier from 0.163 to 0.152 without changing AUROC, a within-model calibration comparison.

5.2 WHAT INFORMATION MATTERS?

Table 2 separates the input modalities. Removing update attributes changes AUROC from 0.831 to 0.793; eye-only prediction reaches 0.754. Removing head motion or scene context gives 0.821 and 0.808. These results support conditioning on behavior and the pending update. They do not establish that event attention is necessary: core and attn retain the update-attribute wide branch and reach AUROC 0.826 and 0.821, respectively. Update conditioning defines the problem; attention is one instantiation.

Table 2: Modality ablations from Table S8. AUROC/AUPRC/F1 use native scores; Brier/ECE use variant-specific beta calibration. FS@20 averages five separately calibrated runs, each selecting 244 of the same 1,219 held-out events.

Input configuration	**AUROC** ↑	**ΔAUROC**	**AUPRC** ↑	**F1** ↑	**Brier** ↓	**ECE** ↓	**FS@20** ↓
Full M2P-CF	**.831**	.000	**.731**	**.700**	**.152**	**.043**	**.056**
Modality-dropout training	.827	-.004	.726	.696	.154	.045	.059
w/o head motion	.821	-.010	.715	.690	.155	.047	.065
w/o scene context	.808	-.023	.701	.677	.165	.052	.076
w/o update attributes	.793	-.038	.684	.661	.174	.058	.087
Eye + head only	.781	-.050	.666	.649	.181	.061	.096
Eye only	.754	-.077	.638	.622	.192	.064	.112

5.3 DO THE OBJECTIVES IMPROVE LOW-RISK ORDERING?

Table 3: Objective ablations. AUROC/AUPRC/F1 use native scores; FS@20 averages five separately beta-calibrated runs, each selecting 244 of the same 1,219 test events. BCE only retains class-weighted BCE and removes all four auxiliary losses.

Variant	**AUROC** ↑	**ΔAUROC**	**AUPRC** ↑	**F1** ↑	**FS@20** ↓
Full M2P-CF	**.831**	.000	**.731**	**.700**	**.056**
w/o grouped contrast	.821	-.010	.721	.690	.073
w/o invariance	.819	-.012	.716	.686	.077
w/o ranking	.825	-.006	.725	.691	.082
BCE only	.816	-.015	.712	.677	.091

Table 3 distinguishes aggregate discrimination from low-risk ordering. Without ranking, AUROC is 0.825 versus 0.831 for Full, while mean FS@20 rises from 0.056 to 0.082. Removing grouped contrastive regularization gives AUROC 0.821 and FS@20 0.073; removing same-label invariance gives AUROC 0.819. At fixed 20% coverage per run, these differences concern which events are selected, not a more conservative coverage choice. Each run is calibrated separately for FS@20; abruptness is retained in single-objective ablations and removed in BCE-only.

FS@20 evaluates a fixed operating point on recorded events; multi-threshold selective evaluation addresses performance over a broader coverage range (Traub et al., 2024). Calibration and logged-replay diagnostics are retained in Appendix A.10 (Fig. S7, Table S22).

5.4 DOES THE POLICY WORK UNDER INTERACTION?

The independent closed-loop study includes 55 participants and 2,475 candidates per policy. Prediction-only applies 572 updates with 60 detections; current-score Safe-UARS applies 656 with 39 detections, giving FSR 0.059 and ER/SUR 0.265/0.249. Their primary participant-paired contrast is $\Delta\text{FSR} = -0.045$ (95% CI $[-0.076, -0.017]$).

Table 4: Independent controlled closed-loop evaluation ($N = 55$; 2,475 candidates per policy). FSR uses applied updates; ER/SUR use candidates. CIs are participant-bootstrap; delay is median/P95 ms. Counts are from Table S14.

Policy	Applied	Detected	FSR [95% CI] ↓	ER ↑	SUR ↑	Delay (ms)
Calibrated prediction-only	572	60	.105 [.084, .127]	.231	.207	0/16
Safe-UARS current	656	39	.059 [.042, .078]	.265	.249	94/226
Safe-UARS re-query	693	21	**.030 [.018, .044]**	**.280**	**.272**	103/238

5.5 Why re-query modified updates?

Current-score and re-query share the model, calibration, risk thresholds, utility parameters, and delay budget. They differ only in whether $U_i \to U_i'$ triggers recomputation of $(p_i^{\mathrm{cal}}, R_i^+, u_i)$. Re-query lowers FSR from 0.059 to 0.030; the primary participant-paired difference is -0.029 (95% CI $[-0.053, -0.005]$).

The paired contrast averages within-participant FSR differences with equal participant weights, whereas Table 4 reports pooled rates. Its interval uses 10,000 paired participant resamples and percentile endpoints (seed 2027; Appendix A.6). Keeping each participant's policy observations together preserves the within-subject comparison; this uncertainty concerns participant sampling under the frozen system.

All three evaluated scene families show the same direction (Table S18). Table 5 decomposes applications and detections by action; Appendix Figs. S3–S4 visualize candidate accounting and paired contrasts. Table S19 reports subjective ratings and queue behavior.

Table 5: Action-level closed-loop results (Table S17): applied and detected event counts, with FSR = Detected/Applied. Subgroups are policy-selected descriptive summaries.

	Current-score			Re-query		
Application action	**Applied**	**Detected**	**FSR ↓**	**Applied**	**Detected**	**FSR ↓**
Execute	395	18	.046	405	16	.040
Attenuate	205	16	.078	225	4	.018
Split	56	5	.089	63	1	.016

The comparison isolates the policy-level effect of re-querying under shared frozen parameters: no additional model training or calibration is introduced. It supports evaluating risk for the modified update rather than retaining its original score. Action-specific rates are descriptive because the policies select different events. The improvement concerns the observed detection–coverage tradeoff and comes with additional decision delay.

Model-side runtime. The $K = 5$ model-side pipeline averages 12.6 ms per event on an RTX 4090 / Intel i9-13900K workstation (Table S21). Profiling uses FP32, one event per batch, and batched/vectorized ensemble execution in PyTorch 2.10.0 with CUDA 12.8. Timing includes model forward passes, calibration, and scheduling over 1,000 synchronized event calls after 100 warm-up calls. This event-triggered measurement excludes rendering and motion-to-photon latency and is not a per-frame 90-Hz timing result.

6 Conclusion

PERCEPT-VR++ treats perceptual risk as a property of the pre-update state–update pair. VR-Mask++ provides nine-scene evidence from 450 participants, and M2P-CF reaches AUROC 0.831. Input and objective ablations distinguish the value of update attributes from low-risk ordering. In an independent 55-participant closed-loop study, re-query improves the observed detection–coverage tradeoff under shared frozen parameters. The paired comparison supports a concrete scheduling principle: when an action changes update attributes, evaluate risk for that changed update.

ETHICS STATEMENT

All studies received institutional ethics approval (details withheld for double-blind review), and all participants provided informed consent, had normal or corrected-to-normal vision, and reported no condition preventing VR use. VR-Mask++ included 225 women and 225 men, while the 55-participant closed-loop cohort included 27 women and 28 men. Across both cohorts, participant age was $M = 26.3$ years ($SD = 6.2$). Participants received fixed monetary compensation. Withdrawal and exclusion procedures followed the approved protocol; reported cohort sizes are the completed analyzed samples.

Eye tracking and synchronized behavioral recordings are sensitive data. Participant identity is excluded from model inputs; participant identifiers are used for split construction and evaluation. Any release of processed data or raw high-frequency gaze streams is subject to the approved consent and data-sharing requirements, with anonymization and access restrictions where needed.

REPRODUCIBILITY STATEMENT

Appendix A.2 describes dataset construction and participant accounting; Appendix A.3 specifies feature construction, leakage controls, and regularization groups. Table S6 gives model and training hyperparameters, and Table S23 defines the model variants. Appendix A.6 documents the closed-loop protocol, event accounting, and participant-bootstrap analysis; Appendix A.7 and Table S20 specify the frozen policy parameters. These sections distinguish model prediction, logged-event selection diagnostics, and independently collected closed-loop outcomes. The code and a partial processed-data release have been uploaded to the submission system; the data release is partial because of the supplementary-file size limit.

AI USE STATEMENT

In preparing this submission, we used generative AI tools for LaTeX template migration, targeted language revision, literature lookup, code-to-method documentation, formatting, and figure layout. We did not use generative AI tools to generate new participant data or model-training results for this revision. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Foundations and Trends in Machine Learning*, 16(4): 494–591, 2023. doi: 10.1561/2200000101.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=33XGfHLtZg`.

Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Transactions on Graphics*, 36(4):1–12, 2017. doi: 10.1145/3072959.3073642.

David C. Burr, M. Concetta Morrone, and John Ross. Selective suppression of the magnocellular visual pathway during saccadic eye movements. *Nature*, 371(6497):511–513, 1994. doi: 10.1038/371511a0.

Runze Fan, Xuehuai Shi, Kangyu Wang, Qixiang Ma, and Lili Wang. Scene-aware foveated rendering. *IEEE Transactions on Visualization and Computer Graphics*, 30(11):7097–7106, 2024. doi: 10.1109/TVCG.2024.3456157.

Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3d graphics. *ACM Transactions on Graphics*, 31(6):1–10, 2012. doi: 10.1145/2366145.2366183.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. Towards attention-aware foveated rendering. *ACM Transactions on Graphics*, 42(4):1–10, 2023. doi: 10.1145/3592406.

Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 623–631. PMLR, 2017.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics*, 35(6):1–12, 2016. doi: 10.1145/2980179.2980246.

Ronald A. Rensink, J. Kevin O'Regan, and James J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997. doi: 10.1111/j.1467-9280.1997.tb00427.x.

Mohammad R. Saeedpour-Parizi, Niall L. Williams, Tim Wong, Phillip Guan, Dinesh Manocha, and Ian M. Erkelens. Perceptual thresholds for radial optic flow distortion in near-eye stereoscopic displays. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2570–2579, 2024. doi: 10.1109/TVCG.2024.3372075.

Xuehuai Shi, Lili Wang, Jian Wu, Wei Ke, and Chan-Tong Lam. Locomotion-aware foveated rendering. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 471–481, 2023. doi: 10.1109/VR55154.2023.00062.

Xuehuai Shi, Yucheng Li, Jiaheng Li, Jian Wu, Jieming Yin, Xiaobai Chen, and Lili Wang. Audio-visual aware foveated rendering. *IEEE Transactions on Visualization and Computer Graphics*, 31 (10):7711–7726, 2025. doi: 10.1109/TVCG.2025.3554737.

Chunlin Sun, Linyu Liu, and Xiaocheng Li. Predict-then-calibrate: A new perspective of robust contextual LP. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0778. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/397271e11322fae8ba7f827c50ca8d9b-Abstract-Conference.html.

Shengpu Tang and Jenna Wiens. Counterfactual-augmented importance sampling for semi-offline policy evaluation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0504. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/25b15618c98ff0c4655df0c5a277e1c6-Abstract.html.

Jeremias Traub, Till J. Bungert, Carsten T. Lüth, Michael Baumgartner, Klaus H. Maier-Hein, Lena Maier-Hein, and Paul F. Jäger. Overcoming common flaws in the evaluation of selective classification systems. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0076. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/047c84ec50bd8ea29349b996fc64af4b-Abstract-Conference.html.

Okan Tarhan Tursun, Elena Arabadzhiyska-Koleva, Marek Wernikowski, Radoslaw Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics*, 38(4):1–14, 2019. doi: 10.1145/3306346.3322985.

Lili Wang, Xuehuai Shi, and Yi Liu. Foveated rendering: A state-of-the-art survey. *Computational Visual Media*, 9(2):195–228, 2023. doi: 10.1007/s41095-022-0306-4.

André Zenner, Kristin Ullmann, Oscar Ariza, Frank Steinicke, and Antonio Krüger. Induce a blink of the eye: Evaluating techniques for triggering eye blinks in virtual reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. Association for Computing Machinery, 2023. doi: 10.1145/3544548.3580888.

A Appendix

A.1 Related Work

Perceptual rendering and masking. Foveated rendering allocates high fidelity near gaze and reduced quality elsewhere (Guenter et al., 2012; Patney et al., 2016; Wang et al., 2023); subsequent work incorporates contrast, attention, locomotion, scene content, gaze motion, and audiovisual context (Tursun et al., 2019; Krajancich et al., 2023; Shi et al., 2023; Fan et al., 2024; Shi et al., 2025). These methods mainly optimize spatial fidelity or reconstruction, whereas we address the temporal decision of when a pending update can be applied. Saccadic suppression and change blindness motivate blink- and saccade-contingent strategies (Burr et al., 1994; Rensink et al., 1997; Arabadzhiyska et al., 2017; Saeedpour-Parizi et al., 2024; Zenner et al., 2023), but binary eye-state rules omit head motion, task context, eccentricity, and update attributes.

Calibration and risk-aware decisions. Thresholded decisions require reliable probabilities, motivating temperature scaling, beta calibration, and deep ensembles (Guo et al., 2017; Kull et al., 2017; Lakshminarayanan et al., 2017). Conformal prediction and risk control provide related perspectives (Angelopoulos & Bates, 2023; Angelopoulos et al., 2024), and predict-then-calibrate separates predictive modeling from downstream robustness calibration (Sun et al., 2023). We use these established components for a pre-update VR scheduling problem with update-conditioned multimodal inputs and independent closed-loop evaluation.

Selective decisions and policy evaluation. Selective-classification evaluation distinguishes a fixed coverage operating point from performance integrated over thresholds (Traub et al., 2024); FS@20 addresses the former. Offline policy evaluation must also distinguish observed event labels from outcomes under changed actions. Semi-offline evaluation introduces explicit counterfactual annotations and estimators to address that distinction (Tang & Wiens, 2023). Our replay retains recorded labels for selection diagnostics, while the independent closed-loop study observes outcomes after actual actions.

Table S1: VR-Mask++ dataset statistics; valid update events exclude catch/no-update trials.

Item	Value
Scenes	9
Scheduling conditions	5
Candidate trials	9,000
Valid update events	8,301
Detected / not detected	2,809 / 5,492
Train / val / test	5,828 / 1,254 / 1,219
Split groups	450
Split-group overlap	None

Table S2: Scene-level primary sample counts.

Scene	A1	A2	A3	B1	B2	B3	C1	C2	C3
Samples	933	1000	894	910	910	923	910	910	911

A.2 VR-MASK++ DATASET DETAILS

Event definition. VR-Mask++ is organized around visual-update events rather than full sessions. Each valid update event corresponds to a pending visual update executed under a specific scene, task, update type, and scheduling condition. Catch/no-update trials belong to the full candidate-trial collection but are excluded from the primary event set because no visual update was executed. Predictive inputs contain pre-update eye dynamics, head motion, context dynamics, and static scene/update features. The optional user/session slot is fixed to $G = \mathbf{0}$ in all reported experiments; participant identifiers are retained only for split construction and evaluation metadata. Detection is the primary supervision signal. Abruptness is used as an auxiliary ordinal training target with $\lambda_{\mathrm{abr}} = 0.05$, whereas discomfort and confidence have zero loss weight and are retained only for diagnostic analysis. This event-level design matches the online decision problem: predict whether an update is likely to be noticed before it is executed.

Dataset scale. The full collection contains 9,000 candidate trials from nine scenes and five scheduling conditions. The default primary set contains 8,301 valid update events after excluding catch/no-update trials. The primary label distribution is 2,809 detected events and 5,492 not-detected events. Train, validation, and test partitions contain 5,828, 1,254, and 1,219 events, respectively, with no split-group overlap.

Table S3: Scheduling-condition counts in the primary set.

Condition	BlinkOnly	EyeStateRule	FixedInterval	PERCEPTVR	Random
Samples	1663	1666	1635	1693	1644

PERCEPTVR collection condition. The PERCEPTVR condition used during VR-Mask++ collection was a frozen predecessor scheduler developed before training the M2P-CF model evaluated in this paper. It was used only to determine update timing during data collection and was not the final PERCEPT-VR++ model. The collection-time scheduler was trained/configured using an earlier, separately collected dataset with no participant or event overlap with VR-Mask++; its parameters were fixed before the corresponding VR-Mask++ sessions. The final M2P-CF models reported in this paper were trained only after VR-Mask++ collection was completed. The logging-condition identity is excluded from all predictive inputs.

Scene families. The scenes are grouped into controlled, natural search, and reading/UI task families. Controlled scenes provide cleaner labels and systematic variation in target eccentricity and update magnitude. Natural search scenes expose the model to unconstrained gaze and head motion under clutter and scene motion. Reading/UI scenes evaluate whether risk-aware scheduling preserves task performance and user comfort.

Table S4: Nine VR scenes used in VR-Mask++.

ID	Family	Task	Update examples	Pre-update or diagnostic factors
A1	Controlled	Central fixation update	LOD, blur/sharpness, contrast	fixation stability, magnitude, contrast
A2	Controlled	Peripheral update	peripheral LOD/resolution/position	eccentricity, gaze drift, head speed
A3	Controlled	Distant small-target update	depth/position, local clarity	distance, target size, background complexity
B1	Natural search	Indoor object search	object LOD, local resolution	clutter, occlusion, head motion
B2	Natural search	Outdoor far-marker search	far-target depth/position, contrast	distance, low contrast, eccentricity
B3	Natural search	Dynamic crowd search	avatar LOD, blur, texture updates	crowd density, target motion, scene motion
C1	Reading/UI	Text reading update	text sharpness, contrast, local rearrangement	text length, fixation, local contrast
C2	Reading/UI	UI panel interaction	button edge, icon, panel resolution	focus region, icon salience, panel contrast
C3	Reading/UI	Multi-panel monitoring	multi-region LOD, chart refresh	panel switching, information density, scene motion

Table S5: Visual update families and controlled factors.

Update family	Implementation	Magnitude control	Risk factors
LOD transition	mesh/material/texture detail switch	small/medium/large detail gap	distance, eccentricity, object salience
Blur / sharpness	shader or local post-processing	blur radius or sharpness strength	text readability, fixation, contrast
Resolution change	local downsampling or foveated shader	region size and sampling scale	gaze drift, head motion, region coverage
Contrast / luminance	material/color-grading change	percentage contrast/luminance delta	background brightness, local contrast, task attention
Depth / position shift	small transform or depth-cue change	angular or depth displacement	target size, distance, motion sensitivity

Visual update families. We use update families that are controllable in Unity and representative of adaptive rendering: LOD transitions, blur/sharpness or resolution changes, contrast/luminance shifts, and depth/position micro-shifts. Update magnitude is discretized or normalized so that the model can learn how perceptual risk scales with visual-change strength.

Trial flow. Each trial begins with a task prompt and sensor logging. The participant then performs a controlled fixation/search/reading/UI task. A visual update is triggered by a predefined policy or by the evaluated scheduler. The participant continues the task to avoid immediate interruption, then reports whether a change was noticed, rates abruptness and discomfort, and optionally provides confidence. Logs store sensor streams, update parameters, model outputs, action choice, detection outcome, and optional diagnostic feedback or performance fields when available.

Ethics and participant accounting. All studies received institutional ethics approval (details withheld for double-blind review), and all participants provided informed consent, had normal or corrected-to-normal vision, and reported no condition preventing VR use. VR-Mask++ included 225 women and 225 men, while the 55-participant closed-loop cohort included 27 women and 28 men. Across both cohorts ($N = 505$), participant age was $M = 26.3$ years ($SD = 6.2$). Participants received fixed monetary compensation. Withdrawal and exclusion procedures followed the approved protocol; reported cohort sizes are the completed analyzed samples.

A.3 Feature Construction and Leakage Control

Feature windows. For each event, eye, head, and context streams are aligned to the update time. The online model uses only the pre-update window and static event attributes. The analysis table may include post-update and full-window statistics for diagnostics, but these variables are excluded from online prediction and main test metrics.

Online inputs. The final model uses pre-update eye dynamics, head motion, scene context, and pending-update attributes. The optional user/session slot is fixed to zero. Participant identifiers and all post-update outcomes are excluded from prediction and retained only for split construction or diagnostic analysis.

Grouped supervised contrastive regularization. Let $h_i \in \mathbb{R}^{128}$ be the final ReLU embedding of the fusion network, before the linear detection head, and $\tilde{h}_i = h_i / \max(\|h_i\|_2, 10^{-12})$. The training runner applies $\mathcal{L}_{\mathrm{cf}}$ and $\mathcal{L}_{\mathrm{inv}}$ to this embedding, rather than to the 512-dimensional concatenated modality vector. The grouped supervised contrastive term uses normalized embeddings $\tilde{h}_i$. Let

$$\mathcal{P}(i) = \{p \neq i : y_p = y_i,\ g_p = g_i \geq 0\}$$

denote in-batch samples with the same detection label and matched-group identifier as anchor i. For anchors with $|\mathcal{P}(i)| > 0$,

$$\mathcal{L}_{\mathrm{cf}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(\tilde{h}_i^\top \tilde{h}_p / \tau)}{\sum_{a \neq i} \exp(\tilde{h}_i^\top \tilde{h}_a / \tau)}, \qquad \tau = 0.10.$$

where $\mathcal{I}$ is the set of valid anchors; the loss is zero if that set is empty. This term enters the total objective as $\lambda_{\mathrm{cf}} \mathcal{L}_{\mathrm{cf}}$ with $\lambda_{\mathrm{cf}} = 0.01$.

Matched invariance. For a training mini-batch, define the unordered same-label pair set

$$\mathcal{P}_{\mathrm{inv}} = \{(i, j) : i < j,\ g_i^{\mathrm{inv}} = g_j^{\mathrm{inv}} \geq 0,\ y_i = y_j\}.$$

The implementation averages cosine distance over all valid pairs, not over group means:

$$\mathcal{L}_{\mathrm{inv}} = \frac{1}{|\mathcal{P}_{\mathrm{inv}}|} \sum_{(i,j) \in \mathcal{P}_{\mathrm{inv}}} \left(1 - \tilde{h}_i^\top \tilde{h}_j\right).$$

Each unordered pair contributes once; groups with more valid pairs consequently receive more weight. The term is zero if no valid pair exists.

Within-group ranking. Let ℓ_i be the raw detection logit, before sigmoid or calibration. For

$$\mathcal{P}_{\mathrm{rank}} = \{(i, j) : g_i^{\mathrm{rank}} = g_j^{\mathrm{rank}} \geq 0,\ y_i = 1,\ y_j = 0\},$$

the pairwise hinge loss is

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{|\mathcal{P}_{\mathrm{rank}}|} \sum_{(i,j) \in \mathcal{P}_{\mathrm{rank}}} \max\{0,\ 0.05 - (\ell_i - \ell_j)\}.$$

All detected–undetected pairs within the same valid group are included with equal pair weight. The margin 0.05 is in logit units. This term is zero for an empty pair set.

Auxiliary abruptness objective and total loss. Abruptness has seven ordered levels. The data loader maps recorded ratings $1, \ldots, 7$ to $r_i = 0, \ldots, 6$; negative encoded targets are masked from the loss. A bias-free linear head gives $v_i = w_{\mathrm{abr}}^\top h_i$. Six ordered cutpoints are parameterized by

$$\theta_0 = b, \qquad \theta_k = b + \sum_{j=1}^{k} \mathrm{softplus}(\delta_j), \quad k = 1, \ldots, 5,$$

where b and δ_j are learned. The six logits $o_{ik} = v_i - \theta_k$ model the cumulative targets $t_{ik} = \mathbf{1}[r_i > k]$, $k = 0, \ldots, 5$. For the rated-event set $\mathcal{V} = \{i : r_i \geq 0\}$,

$$\mathcal{L}_{\text{abr}} = -\frac{1}{6|\mathcal{V}|} \sum_{i \in \mathcal{V}} \sum_{k=0}^{5} \left[t_{ik} \log \sigma(o_{ik}) + (1 - t_{ik}) \log(1 - \sigma(o_{ik})) \right].$$

The implementation uses binary cross-entropy with logits and mean reduction across rated events and cutpoints. An unrated mini-batch contributes zero. Cutpoints are initialized with $b = -1$ and positive gaps of $2/5$. Discomfort and confidence have zero loss weight. The complete objective is

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{cf}} \mathcal{L}_{\text{cf}} + \lambda_{\text{inv}} \mathcal{L}_{\text{inv}} \\ + \lambda_{\text{rank}} \mathcal{L}_{\text{rank}} + \lambda_{\text{abr}} \mathcal{L}_{\text{abr}}, \qquad \lambda_{\text{abr}} = 0.05. \end{aligned}$$

Cleaning and encoding. We standardize scene and policy names, merge event-window features by composite keys, retain catch/no-update trials only for false-alarm checks, split windows into pre-update and post-update segments, impute missing numerical values using training-set medians, add missingness indicators, and one-hot encode core categorical fields. Feature dimensionality and category mappings are stored with each checkpoint's feature specification.

Split construction. Rows are assigned to train, validation, and test sets through predefined split groups, with no split-group overlap across partitions. Participant identifiers are used only to construct and audit these groups, never as predictive features. The validation split is used for model selection and probability calibration; the test split is reserved for final held-out split-group reporting.

A.3.1 Regularization Pair Construction

Group identifiers are factorized within the training split only. CF and ranking groups use the tuple (scene family, update type, eccentricity bin, eye state); invariance groups use (update type, update magnitude, eccentricity bin, eye state). The three eccentricity bins use training-set quantile edges, with duplicate edges merged. Validation and test rows retain invalid group identifiers. Group construction uses pre-update metadata, not post-update gaze reactions or task outcomes. Detection labels then select same-label CF/invariance pairs or detected–undetected ranking pairs within a mini-batch. All valid in-batch pairs are used; no separate pair sampler is applied. Unmatched events remain in the class-weighted BCE objective. These observational matches provide regularization rather than causal identification.

A.4 Baselines, Metrics, and Inference

Prediction baselines. We compare M2P-CF with four baseline families. Rule baselines include Blink, Saccade, Eye-state (blink, saccade, fixation, and gaze validity), and Eye+Head, which adds head angular velocity and stability. Logistic regression, random forest, XGBoost, CatBoost, MLP, and FT-Transformer consume event-level pre-update summary features and encoded scene/update attributes. The temporal Transformer instead consumes cached pre-update sequences; its forward pass does not concatenate the tabular update-attribute vector. Table S24 makes these input differences explicit. Finally, a saccade-landing-aware baseline predicts the next fixation landing region from recent gaze velocity and acceleration and treats updates outside it as safer. It captures anticipatory eye dynamics but omits multimodal scene context, update attributes, calibration, and ensemble disagreement.

Scheduling baselines. The evaluated logged-replay policies include fixed interval, blink-only, eye-state rule, prediction-only scheduling, and Safe-UARS. Random timing is retained as a data-collection condition rather than as part of the final policy comparison. Prediction-only uses raw or calibrated risk probabilities directly; Safe-UARS uses the validation-calibrated conservative risk score and may execute, delay, attenuate, or split pending updates.

Confidence intervals. The closed-loop evaluation reports pooled event-level rates, participant-bootstrap intervals, and paired participant-level bootstrap contrasts. Pooled and participant-weighted estimates are distinguished explicitly because their point estimates need not coincide.

Table S6: M2P-CF architecture and training settings. All calibration models are fitted using validation predictions only; test predictions are transformed without refitting.

Item	Setting
Pre-update window	500 ms pre-update-only window; no post-update samples
Sampling rate / sequence length	120 Hz / 60 time steps
Temporal encoder	One-layer forward-only GRU; padded or invalid time steps are masked before event-conditioned temporal attention. The attention-weighted hidden-state sum provides the final modality summary.
GRU hidden dimension	128 for each temporal stream
Event attention	128-dimensional single event-conditioned temporal attention per modality; the pending-update representation is used as the event query, and static features enter only through the wide branch
Fusion network	Concatenation of the eye, head, context, and wide representations, followed by fully connected layers of widths $512 \to 256 \to 128$. LayerNorm, ReLU, and dropout follow the 512- and 256-dimensional layers; ReLU follows the 128-dimensional layer. A final linear layer maps the fused representation to the risk logit.
Dropout	0.10 after the 512- and 256-dimensional fusion layers
Optimizer	AdamW
Learning rate	3×10^{-4}
Batch size	128
Weight decay	1×10^{-4}
Maximum epochs	120
Early stopping	Patience of 20 epochs, selected by validation AUROC
Class weighting	Automatic positive-class weight, computed as the training negative-to-positive ratio
Gradient clipping	Maximum gradient norm of 5.0
λ_{cf}	0.01
λ_{inv}	0.01
λ_{rank}	0.02
λ_{abr}	0.05 for the auxiliary ordinal abruptness objective; discomfort and confidence use zero loss weight
Ranking margin	0.05
Contrastive temperature	$\tau = 0.10$ for the grouped supervised contrastive term $\mathcal{L}_{\mathrm{cf}}$
Ensemble construction	Five independently initialized runs using seeds $\{11, 22, 33, 44, 55\}$ with the same split and training configuration
Ensemble aggregation	Arithmetic mean of member probabilities; diagnostic ensemble disagreement is the across-member probability standard deviation, $\bar{p}_i = K^{-1} \sum_{k=1}^{K} \sigma(\ell_i^{(k)})$ and $u_i^{\mathrm{epi}} = \mathrm{SD}_{k=1}^{K}[\sigma(\ell_i^{(k)})]$
Beta-calibration input	Raw ensemble probability p (individual-run probability for FS@20), clipped to $[10^{-6}, 1 - 10^{-6}]$ and represented as $[\log p, \log(1-p)]$
Beta-calibration fitting	L2 logistic regression with $C = 1$ and `max_iter=1000`, fitted separately for each evaluated model on the 1,254-event validation calibration split
Final M2P-CF ensemble beta mapping	$g_\beta(p) = \sigma(1.67885 \log p - 0.41256 \log(1-p) + 0.35919)$; the fitted mapping is frozen for test and closed-loop use
Calibration selection	Temperature, beta, and isotonic calibration are compared on validation predictions; beta calibration is used for the final reported M2P-CF probabilities

Run-wise FS@20. For run k, let S_k be the 244 test events with the lowest risk after fitting that run's beta calibrator on validation data. The reported quantity is

$$\mathrm{FS@20} = \frac{1}{5} \sum_{k=1}^{5} \frac{\sum_{i \in S_k} y_i}{244}.$$

The five sets may overlap and need not be identical. This averages five run-wise rates; it is distinct from averaging member probabilities first and selecting a single ensemble-ranked set. The five-member ensemble is used for the separate ensemble-prediction and scheduling results.

Table S7: Metrics used for offline prediction, calibration, and policy evaluation.

Category	Metrics	Purpose
Discrimination	AUROC, AUPRC, F1	detectability ranking and classification
Calibration	Brier, ECE, reliability curve	reliability of predicted probabilities
Risk outcomes	false-safe rate, execution ratio	compare replay risk and coverage
Applied coverage	successful update ratio and added delay	undetected application coverage and decision delay
User experience	visual distraction, comfort	diagnostic perceptual quality and comfort

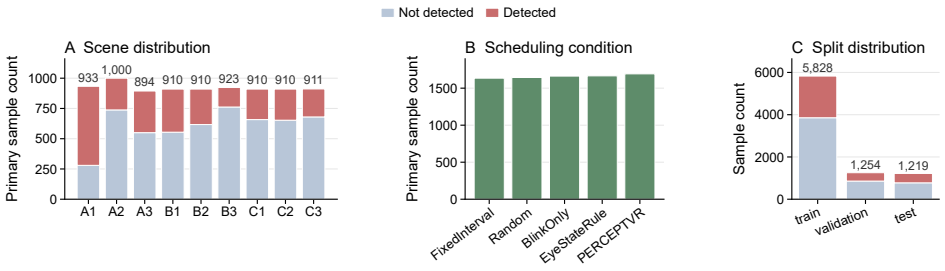

Figure S1: Dataset composition of VR-Mask++. (A) Valid update events across nine VR scenes. (B) Scheduling-condition distribution. (C) Split-group train/validation/test distribution. Panels A and C are stacked by detected and not-detected labels, whereas Panel B reports total primary counts by scheduling condition.

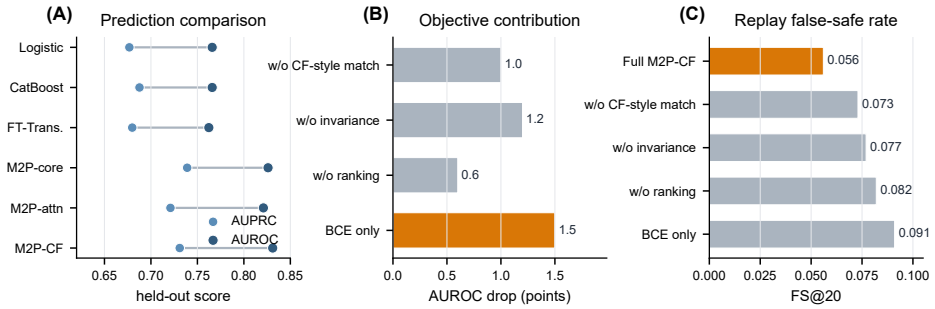

Figure S2: Prediction and objective-ablation results. (A) AUROC and AUPRC for representative baselines and M2P variants on the held-out split-group test set. (B) AUROC decrease after removing each auxiliary objective. (C) FS@20, averaged over five separately beta-calibrated training runs, each selecting its 244 lowest-risk events from the same 1,219-event test set; lower is better. Values reproduce Tables 1 and 3 of the main paper. The ordinal abruptness loss is held fixed in the single-objective ablations and removed in BCE only.

A.5 Additional Results

Figure S1: dataset composition. Figure S1 shows the scene, scheduling-condition, and split distributions across the primary event set.

Figure S2: prediction and objective ablations. Figure S2 summarizes the final held-out prediction comparison and the objective-level ablations reported in the main paper. The full model provides the strongest representative AUROC, while removal of the auxiliary objectives degrades either aggregate discrimination, replay false-safe ordering, or both. In particular, the ranking objective has a modest effect on AUROC but a larger effect on FS@20, consistent with its scheduler-facing role.

Modality ablations. Table S8 tests whether the full-model gains can be attributed to eye dynamics alone. Modality-dropout training causes only a small degradation, whereas removing head motion,

Table S8: Modality ablations on the held-out split-group test set. AUROC, AUPRC, and F1 use native scores; Brier and ECE use beta calibration fitted separately for each variant. FS@20 is the mean of five run-wise false-safe rates, with separate validation-fitted beta calibration and selection of 244 of the same 1,219 test events in each run.

Variant	AUROC ↑	ΔAUROC	AUPRC ↑	F1 ↑	Brier ↓	ECE ↓	FS@20 ↓
Full M2P-CF	.831	.000	.731	.700	.152	.043	.056
Modality-dropout training	.827	-.004	.726	.696	.154	.045	.059
w/o head motion	.821	-.010	.715	.690	.155	.047	.065
w/o scene context	.808	-.023	.701	.677	.165	.052	.076
w/o update attributes	.793	-.038	.684	.661	.174	.058	.087
Eye + head only	.781	-.050	.666	.649	.181	.061	.096
Eye only	.754	-.077	.638	.622	.192	.064	.112

Table S9: Descriptive held-out replay comparison. ER-matched prediction-only uses the full-policy execution ratio.

Policy	FSR ↓	ER ↑
Raw prediction-only	.089	.230
Prediction-only, ER-matched	.105	.265
Safe-UARS full	**.050**	**.265**

scene context, or update attributes progressively reduces discrimination and increases FS@20. The eye-only model performs worst among the modality variants, with AUROC decreasing from 0.831 to 0.754 and FS@20 increasing from 0.056 to 0.112. These results support retaining head motion, scene context, and pending-update attributes alongside eye dynamics.

A.6 Online Policy Evaluation

Action semantics. Safe-UARS chooses among four actions. EXECUTE applies the original update immediately. DELAY is an intermediate transition: within the two-candidate-window budget, it postpones the update and re-queries the model at the next eligible window using a fresh 500-ms pre-update window. ATTENUATE reduces the update magnitude, such as using a smaller LOD gap or weaker blur change. SPLIT decomposes a large update into several smaller steps when fallback is permitted and a valid, admissible split exists. Execute, Attenuate, and Split are terminal application actions. Candidates reaching expiration or the end of eligible decision opportunities without an admissible application are recorded as Not applied. In logged replay, ATTENUATE and SPLIT are selected using the current event-level conservative risk score and fixed utility proxies rather than by re-scoring hypothetical modified updates.

Logged policy outcomes. Core replay records contain the policy, selected action, raw and calibrated probabilities, conservative risk score, hard-gate uncertainty u_i, application flag, and detection outcome. For prediction-only, ER-matched, a post-hoc score threshold is chosen to match the held-out ER of full Safe-UARS; this descriptive point is not used for policy tuning. Delay, post-action update attributes, interaction signals, comfort feedback, and rendering diagnostics are analyzed only in experiments where those fields are available; they are not assumed in the main logged-replay comparison.

Replay policy results. The replay analyses use the held-out records summarized in Table S22. At the descriptive ER-matched point, prediction-only gives FSR 0.105 at ER 0.265, while full Safe-UARS gives FSR/ER of 0.050/0.265. These are selection diagnostics over recorded events: all selected events retain their original detection labels, including events assigned ATTENUATE or SPLIT. The rates describe the selected sets, not detection outcomes for hypothetical modified updates. The independent closed-loop study evaluates outcomes after the selected actions are actually applied.

A.6.1 Closed-Loop Evaluation Protocol and Results

Purpose and scope. The controlled closed-loop evaluation tests whether the logged-replay policy ordering persists under interactive operation, where actions can alter subsequent rendering state,

Table S10: Logged-replay policy comparison conditions. Closed-loop conditions are defined in Table S13.

Policy	Decision rule	Diagnostic purpose
Fixed interval	periodic execution	no perceptual timing
Blink-only	execute during detected blinks	blink-masking rule
Eye-state rule	hand-coded blink, saccade, fixation, and gaze-validity thresholds	stronger heuristic
Prediction-only	execute below a risk threshold	direct threshold scheduling
Safe-UARS	actions from R_i^+, hard gate u_i, and utility proxies	full scheduler

Table S11: Closed-loop evaluation block and event counts.

Item	Value
Participants	55
Policy conditions	3
Representative scenes	3
Candidate updates per policy-scene block	15
Candidate updates per policy per participant	45
Candidate updates per policy	2,475
Candidate updates per participant	135
Total policy-event observations	7,425
Catch/no-update trials	$\sim$10% additional

delay state, and queue state. The evaluation is not used to train, calibrate, or tune PERCEPT-VR++; all model weights, beta-calibration parameters, uncertainty cutoffs, risk thresholds, utility weights, and delay budgets were fixed from the original validation split before data collection began. It provides within-study replay-to-live evidence under the evaluated scenes and platform, but it is not intended to establish a population-wide or cross-device deployment guarantee.

Participants and block design. The evaluation included 55 new participants (27 women and 28 men) who were not part of VR-Mask++. Each participant completed a within-subject design crossing three policies with three representative scenes. Each policy-scene block contained 15 candidate visual updates sampled from update types and magnitudes represented in the training data, yielding 45 candidates per policy per participant, 2,475 candidates per policy, and 7,425 policy-event observations in total. Approximately 10% additional catch/no-update trials were included for attention and false-report checks, but they were not mixed into the applied-update false-safe denominator. The six possible policy orders were allocated as evenly as the final sample permitted, and scene order was Latin-square or near-balanced within each policy order.

Scenes and closed-loop execution. The three evaluation scenes sample one controlled detection scene, one natural visual-search scene, and one reading/UI interaction scene. Candidate events are balanced across update type, magnitude bin, eccentricity bin, and scene family as much as possible. Unlike logged replay, the selected policy action changes the actual rendered scene. EXECUTE applies the pending update, ATTENUATE reduces update magnitude, SPLIT decomposes a larger update into smaller sub-updates, and DELAY waits for a new eligible window while collecting a fresh pre-update eye, head, and scene context. Participants provide detection labels only for actually presented updates; detection outcomes are not imputed for candidates that are never applied. For the re-query condition, modified update attributes are kept within the range covered by the original training distribution.

Endpoints and metrics. The primary endpoint is closed-loop false-safe rate, defined as the proportion of applied updates that are detected. Secondary endpoints are execution ratio, successful update ratio, median and 95th-percentile added decision delay, queue-length summaries, action distribution, per-action and per-scene false-safe rate, and block-level visual-distraction and comfort ratings. Execution ratio is computed over all candidate events, and successful update ratio counts candidate events that are both applied and undetected.

Table S12: Counterbalancing scheme. A: calibrated prediction-only; B: current-score Safe-UARS; C: Safe-UARS re-query.

Design element	Implementation
Policy order	Six permutations (ABC, ACB, BAC, BCA, CAB, CBA), allocated as evenly as the final sample permitted
Scene order	Latin-square or near-balanced within each policy-order stratum

Table S13: Closed-loop policy conditions. Current-score and re-query Safe-UARS share all frozen parameters in Table S20; they differ only in whether modified update attributes are re-scored.

Condition	Actions	Closed-loop rule
Calibrated prediction-only	Execute, Delay	Beta-calibrated probability with a fixed threshold; Delay acquires a fresh pre-update state and re-queries.
Current-score Safe-UARS	Execute, Delay, Attenuate, Split	After $U_i \to U_i'$, retains the original calibrated probability, conservative score, and hard-gate quantity u_i for Attenuate/Split.
Safe-UARS re-query	Execute, Delay, Attenuate, Split	After $U_i \to U_i'$, recomputes calibrated probability, conservative score, and u_i before applying Attenuate/Split.

Closed-loop metric definitions and bootstrap. For closed-loop results,

$$\mathrm{FSR} = \frac{N_{\mathrm{detected,applied}}}{N_{\mathrm{applied}}}, \qquad \mathrm{ER} = \frac{N_{\mathrm{applied}}}{N_{\mathrm{candidate}}},$$
$$\mathrm{SUR} = \frac{N_{\mathrm{undetected,applied}}}{N_{\mathrm{candidate}}}.$$

Added delay is computed only over ultimately applied updates; immediate applications are assigned zero delay. Pooled ER/SUR use candidate counts, whereas pooled FSR uses applied-update counts. The participant-paired contrasts are the primary inferential analysis. Crude event-level odds ratios are secondary descriptive summaries; their conventional intervals do not model within-participant dependence.

Bootstrap implementation and estimands. Both procedures use 10,000 bootstrap replicates, NumPy's `default_rng` with seed 2027, and percentile intervals at the 2.5th and 97.5th percentiles (linear quantile interpolation). For a single policy, whole participants are resampled with replacement, their complete event counts are pooled, and the rate is recomputed. Thus the reported single-policy point estimate and bootstrap statistic are pooled event-level rates.

For the two adjacent-policy contrasts, let D_{sp} and A_{sp} be detected and applied counts for participant s under policy p. The paired estimand is

$$\widehat{\Delta}_{q-p} = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{D_{sq}}{A_{sq}} - \frac{D_{sp}}{A_{sp}}\right), \qquad S = 55.$$

The same sampled participants are used for both policies, and each participant-level difference receives equal weight. In code, multinomial bootstrap multiplicities w_s sum to S, and each replicate is $S^{-1}\sum_s w_s(D_{sq}/A_{sq} - D_{sp}/A_{sp})$. This is distinct from subtracting two pooled FSRs. The function requires matching participant sets and a nonzero applied count in both policies for every participant; otherwise it raises an error rather than imputing a ratio.

Prespecified validation gates. Before analysis, four validation criteria were specified: (i) the participant-level paired-bootstrap 95% CI for current-score versus prediction-only ΔFSR had to lie entirely below zero; (ii) the corresponding CI for re-query versus current-score had to lie entirely below zero; (iii) both prespecified crude detection odds-ratio 95% CIs had to lie below one with Holm-adjusted $p < 0.05$; and (iv) the FSR direction had to be consistent across all three scene families. The paired-bootstrap contrasts constitute the primary inference. Although prespecified, the crude OR checks are secondary because they do not model repeated observations within participants. Action-, scene-, system-, and subjective-result tables are descriptive subgroup analyses.

Table S14: Closed-loop event accounting. Not applied denotes candidates reaching expiration or the end of eligible decision opportunities without an admissible application; no perceptual label is imputed for them. Applied includes Execute, Attenuate, and Split.

Policy	Candidates	Applied	Not applied	Detected	Undetected
Calibrated prediction-only	2,475	572	1,903	60	512
Safe-UARS current	2,475	656	1,819	39	617
Safe-UARS re-query	2,475	693	1,782	21	672

Table S15: Primary results from the 55-participant closed-loop evaluation. FSR is computed among applied updates; ER and SUR use all candidates. CIs are participant-bootstrap intervals.

Policy	FSR [95% CI] ↓	ER ↑	SUR ↑	Median/P95 delay
Calibrated prediction-only	.105 [.084, .127]	.231	.207	0/16 ms
Safe-UARS current	.059 [.042, .078]	.265	.249	94/226 ms
Safe-UARS re-query	.030 [.018, .044]	.280	.272	103/238 ms

Primary closed-loop outcome. Compared with calibrated prediction-only scheduling, current-score Safe-UARS lowers pooled FSR from 0.105 to 0.059 while increasing ER from 0.231 to 0.265 and SUR from 0.207 to 0.249. Its prespecified paired contrast is $\Delta\text{FSR} = -0.045$ (95% CI $[-0.076, -0.017]$), with a crude detection odds ratio of 0.54 [0.35, 0.82] and Holm-adjusted $p = 0.0079$. Re-querying after modifying update attributes further lowers FSR to 0.030 and increases ER to 0.280 and SUR to 0.272. Relative to current-score Safe-UARS, the prespecified contrast is $\Delta\text{FSR} = -0.029$ ($[-0.053, -0.005]$), with a crude detection odds ratio of 0.49 [0.29, 0.85] and Holm-adjusted $p = 0.0108$. Both adjacent-policy gates pass, and the same ordering is observed across all three scene families. The lower observed FSR and higher ER/SUR come with additional decision latency, especially when modified updates are re-scored.

Inferential scope. Both primary participant-paired confidence intervals lie below zero. The prespecified secondary crude OR checks and scene-ordering check also pass. The within-study conclusion rests on the paired analysis; action-, scene-, system-, and subjective-result tables remain descriptive because no separately multiplicity-adjusted subgroup tests were prespecified.

A.7 Replay Utility Model

The action may preserve the original update, postpone it, reduce its magnitude, or decompose it into several smaller updates. Let $g(a_t)$, $d(a_t)$, and $q(a_t)$ denote fixed benefit, delay, and quality proxies. The following utility expression motivates the policy design:

$$\max_{\pi} \mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \left[g(a_t) - \lambda_d d(a_t) - \lambda_q q(a_t) \right], \tag{S1}$$

subject to a perceptual risk constraint,

$$R^+(s_t) \leq \epsilon, \tag{S2}$$

where $R^+(s_t)$ is a validation-adjusted score for the current pending update and ϵ is a selected decision threshold. Safe-UARS implements the fixed rules below rather than solving this constrained optimization online; the proxy terms encode action preferences rather than measured GPU savings or retained visual quality. M2P-CF estimates perceptual risk, and Safe-UARS uses it to select a rendering action. The central policy failure mode is a *false-safe* decision: executing a visually disruptive update because its risk was underestimated.

The fixed proxy terms for rendering benefit $g(a)$, delay cost $d(a)$, and quality loss $q(a)$ motivate the scheduler's action preferences. Safe-UARS implements the rules in Table S20; the utility expression is a design objective rather than an optimization solved online. The proxy terms are normalized in advance; equivalently, the implementation uses $\lambda_d = \lambda_q = 1$ after normalization. Uncertainty is used only as a hard gate:

$$u_i = \max\{u_i^{\text{epi}}, |R_i^+ - p_i^{\text{cal}}|\}.$$

It is not added to p_i^{cal} or R_i^+, and there is no uncertainty penalty or λ_u. With the reported $q_{1-\alpha} = 0.06752 < u_{\max} = 0.08$, the margin term never exceeds the gate cutoff. Thus $u_i \leq 0.08$ is

Table S16: Prespecified adjacent-policy contrasts. Paired-bootstrap ΔFSR intervals provide primary inference. Crude event-level ORs, their conventional intervals, and Holm-adjusted p-values are secondary descriptive checks and do not model within-participant dependence.

Contrast	ΔFSR [95% CI]	OR [95% CI]	Raw p	Holm p	Gate
Current vs. prediction-only	-0.045 [-0.076, -0.017]	0.54 [0.35, 0.82]	0.0040	0.0079	PASS
Re-query vs. current	-0.029 [-0.053, -0.005]	0.49 [0.29, 0.85]	0.0108	0.0108	PASS

Table S17: Action-specific closed-loop detection results. These are descriptive subgroup summaries rather than separately multiplicity-adjusted tests.

Policy	Action	Applied	Detected	FSR
Calibrated prediction-only	Execute	572	60	.105
Safe-UARS current	Execute	395	18	.046
Safe-UARS current	Attenuate	205	16	.078
Safe-UARS current	Split	56	5	.089
Safe-UARS re-query	Execute	405	16	.040
Safe-UARS re-query	Attenuate	225	4	.018
Safe-UARS re-query	Split	63	1	.016

equivalent to $u_i^{\text{epi}} \leq 0.08$; the gate tests ensemble disagreement, while the global margin shifts risk thresholds.

Validation-derived offset and threshold selection. For validation event j, the one-sided residual is $s_j = [y_j - p_j^{\text{cal}}]_+$. We compute $q_{1-\alpha}$ as the higher empirical $(1-\alpha)$-quantile of these residuals and use $R_i^+ = \min(1, p_i^{\text{cal}} + q_{1-\alpha})$. The reported configuration uses $q_{1-\alpha} = 0.06752$ with $\alpha = 0.10$. Because this is a global additive offset, it shifts absolute thresholds without changing event ordering, apart from clipping ties. For a target validation execution budget c, the policy threshold is selected as

$$\epsilon(c) = R^+_{(k)}, \qquad k = \text{round}(cN_{\text{val}}),$$

where $R^+_{(k)}$ is the kth order statistic of validation conservative scores. We set $\epsilon_s = \epsilon(C_s)$ with direct-execution budget target $C_s = 0.16$, and $\epsilon_a = \epsilon(C_a)$ with action-budget target $C_a = 0.24$; $u_{\max} = 0.08$. With $N_{\text{val}} = 1,254$, these are order statistics 201 and 301, respectively. The targets select thresholds; ties, gating, later windows, and transformed actions mean that they are not equalities constraining observed ER. Current-score and re-query share ϵ_s, ϵ_a, $u_{\max}$, $q_{1-\alpha}$, B, λ_d, λ_q, minimum gain, and split limit. All are frozen before closed-loop collection.

In logged replay and current-score scheduling, the score tuple is retained after attenuation or split attributes change. In closed-loop re-query, replacing U_i with U_i' gives $x_i' = \{E_i, H_i, C_i, U_i', G_i\}$, and the model recomputes $p^{\text{cal}}(x_i')$, $R^+(x_i')$, and $u(x_i')$ before application. The frozen thresholds, gate cutoff, and utility parameters remain identical across the two closed-loop Safe-UARS variants; both acquire a fresh pre-update window after Delay.

A.8 Reproducibility, Privacy, and Limitations

Implementation. The stored implementation uses a Unity/Quest Pro logging pipeline with synchronized eye gaze, gaze confidence, head pose, scene state, visual-update parameters, user feedback, and system-performance fields. Offline training uses Python machine-learning and deep-learning libraries. ONNX export is a possible path for future on-device integration but is not evaluated as part of the reported runtime evidence.

Runtime profiling. Table S21 reports the module-level mean latency corresponding to the 12.6 ms model-side inference overhead reported in the main paper. Profiling used an NVIDIA RTX 4090 GPU and Intel Core i9-13900K CPU with PyTorch 2.10.0 and CUDA 12.8. Inference used FP32 and a batch size of one event, with each event comprising a 500-ms, 60-step window. The $K = 5$ ensemble members were evaluated using batched/vectorized GPU execution under `model.eval()` and `torch.no_grad()`; batching across ensemble members is distinct from the single-event input batch. After 100 warm-up calls, latency was averaged over 1,000 measured event inferences, using

Table S18: Closed-loop results by scene family. Each policy contains 825 candidate events per scene family.

Policy	Scene family	Applied	FSR	ER	SUR
Calibrated prediction-only	Controlled detection	190	.100	.230	.207
Calibrated prediction-only	Natural search	191	.115	.232	.205
Calibrated prediction-only	Reading/UI	191	.099	.232	.208
Safe-UARS current	Controlled detection	219	.059	.265	.250
Safe-UARS current	Natural search	219	.068	.265	.247
Safe-UARS current	Reading/UI	218	.050	.264	.251
Safe-UARS re-query	Controlled detection	231	.026	.280	.273
Safe-UARS re-query	Natural search	231	.039	.280	.269
Safe-UARS re-query	Reading/UI	231	.026	.280	.273

Table S19: Closed-loop system behavior and subjective ratings. Distraction and comfort use seven-point scales (mean $\pm$ SD); re-query and budget exhaustion are over candidate events, and action percentages are over applied events. Budget exhaustion records the waiting limit, whereas Not applied records the final application status.

Policy	Queue med./P95	Re-query	Budget ex.	Atten.	Split	Distract.	Comfort
Calibrated prediction-only	1/3	0.00	3.0%	0.0%	0.0%	3.31 ± 0.55	4.38 ± 0.41
Safe-UARS current	1/3	0.00	2.3%	31.2%	8.5%	2.69 ± 0.47	4.83 ± 0.42
Safe-UARS re-query	1/3	0.14	1.9%	32.5%	9.1%	2.63 ± 0.45	4.94 ± 0.36

Table S20: Frozen Safe-UARS parameters shared by current-score and re-query. Utility terms are fixed proxies rather than measured perceptual scores.

Rule	Value	Policy implementation
ϵ_s	$\epsilon(C_s), C_s = 0.16$	Validation direct-execution threshold; Execute when $R_i^+ \le \epsilon_s$ and $u_i \le u_{\max}$.
ϵ_a	$\epsilon(C_a), C_a = 0.24$	Validation action threshold; Attenuate requires $R_i^+ \le \epsilon_a$ and the minimum gain proxy.
$u_{\max}$	.08	Hard gate on $u_i = \max\{u_i^{\text{epi}}, \lvert R_i^+ - p_i^{\text{cal}}\rvert\}$.
$q_{1-\alpha}$	.06752 ($\alpha = .10$)	Higher empirical quantile of $[y_j - p_j^{\text{cal}}]_+$; $R_i^+ = \min(1, p_i^{\text{cal}} + .06752)$.
Minimum gain	.05	Fixed gain proxy required for Attenuate.
Delay budget	2 candidate windows	Exhaustion ends further waiting; application remains conditional on an admissible action.
Re-query interval	Next eligible window	A delayed event acquires a fresh 500-ms pre-update window.
λ_d, λ_q	1 after normalization	Delay and quality proxies are normalized before utility evaluation.
Split limit	One terminal Split per candidate	Fallback requires a valid, admissible split; replay does not recursively simulate sub-update steps.

`torch.cuda.synchronize()` for synchronized GPU timing. The timed scope included model forward computation, calibration, and scheduling; sensor acquisition, rendering, motion-to-photon latency, networking, and headset-side deployment were excluded.

Table S21: Runtime profiling of the complete PERCEPT-VR++ inference pipeline.

Module	**Mean latency (ms)**
Eye encoder	1.7
Head encoder	1.5
Context encoder	2.1
Event attention	0.7
Fusion head	1.5
Additional ensemble-member forward passes and aggregation ($K = 5$)	4.8
Calibration	0.2
Safe-UARS	0.1
Total	12.6

Data availability and privacy. Processed-data release is subject to participant consent, anonymization, and the approved data-sharing requirements. Eye-movement data are sensitive behavioral data; raw high-frequency streams therefore require stronger anonymization and any additional consent required by the governing protocol.

Limitations. The dataset uses one hardware platform and a finite set of VR scenes and update types, so cross-device and application transfer remain open questions. Logged replay preserves recorded pre-update state and labels but cannot reproduce action-induced gaze, head-motion, or queue changes. The 55-participant closed-loop evaluation provides controlled same-platform evidence with prespecified adjacent-policy contrasts, but it covers only three representative scenes and does not establish longitudinal, cross-device, or population-wide deployment performance. The conservative score and replay utilities remain validation-derived heuristics rather than certified controls or direct hardware-level rewards. Detection and comfort reports may contain response bias, while catch/no-update trials are diagnostic rather than a balanced signal-detection study.

A.9 Additional Visualizations

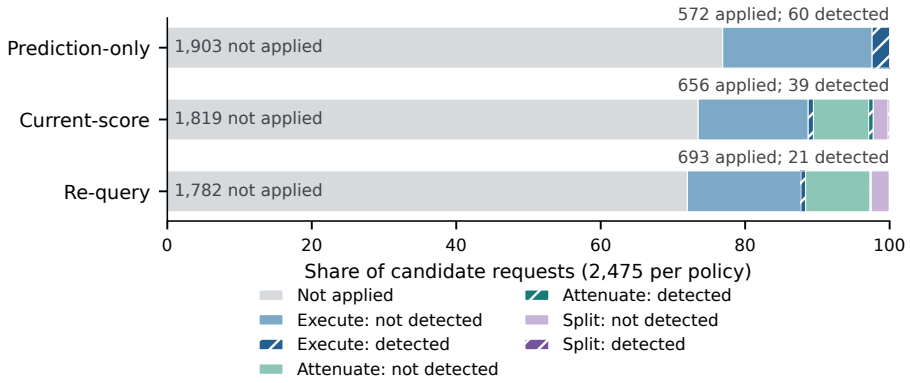

Figure S3: Closed-loop action and outcome counts. Each bar represents 2,475 candidates; hatched segments denote detected applied updates. Counts reproduce Tables S14 and S17. Action subgroups describe policy-selected updates.

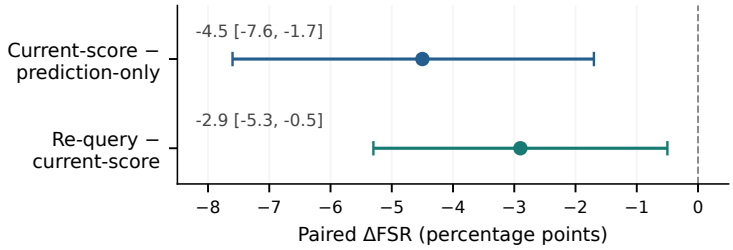

Figure S4: Prespecified participant-paired FSR contrasts for 55 participants. Points and intervals reproduce the estimates and participant-bootstrap 95% CIs in Table S16, in percentage points. Negative values favor the first policy.

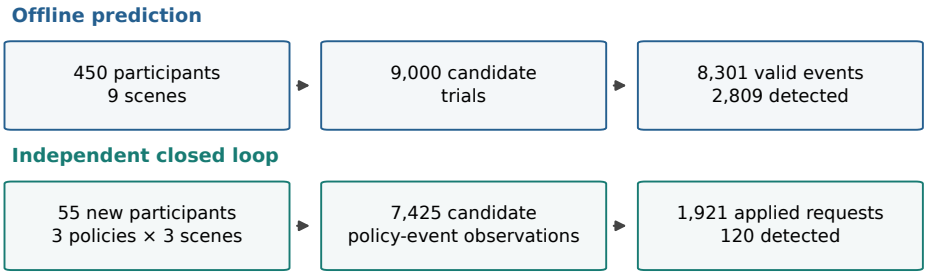

Figure S5: Study coverage. The offline study includes 450 participants and 8,301 valid events. The independent closed-loop study includes 55 new participants and 7,425 candidate policy-event observations; 1,921 updates were applied and labeled.

A.10 Additional Scenes and Logged-Replay Diagnostics

Scene coverage. The complete nine-scene illustration is preserved here. Controlled detection, natural search, and reading/UI tasks cover central and peripheral updates and static and dynamic visual content.

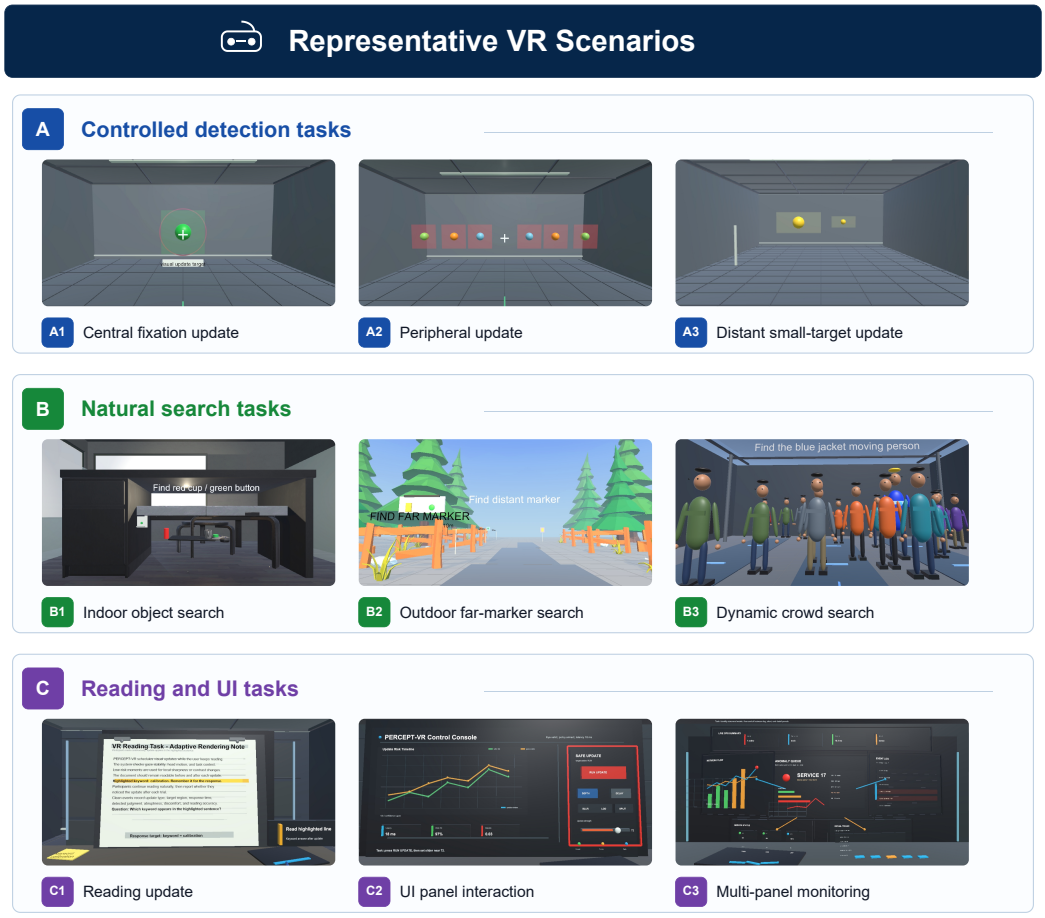

Figure S6: Representative VR-Mask++ scenes spanning controlled detection, natural search, and reading/UI tasks.

Calibration-to-policy replay analysis. Beta calibration lowers ECE from 0.109 to 0.043. At a descriptive ER-matched operating point, prediction-only gives FSR 0.105 at ER 0.265. Calibration and the global offset preserve low-risk ordering here, so neither receives an independent matched-coverage FSR gain. Full Safe-UARS selects a set with FSR/ER of 0.050/0.265, compared with 0.089/0.230 for raw prediction-only. These rates use original recorded-update labels, including when replay selects Attenuate or Split; they describe event selection rather than counterfactual outcomes of those actions. Figure S7B–D shows the endpoints, and Table S22 gives the matched-coverage comparison. The closed-loop experiment provides the evidence for outcomes under actual policy actions.

Logged labels support diagnostics of the selected event sets. They do not provide the unobserved outcomes required to estimate counterfactual policy value; semi-offline evaluation explicitly distinguishes such counterfactual information (Tang & Wiens, 2023).

Table S22: Descriptive held-out replay comparison. The ER-matched prediction-only point uses the full-policy execution ratio.

Policy	FSR ↓	ER ↑
Raw prediction-only	.089	.230
Prediction-only, ER-matched	.105	.265
Safe-UARS full	**.050**	**.265**

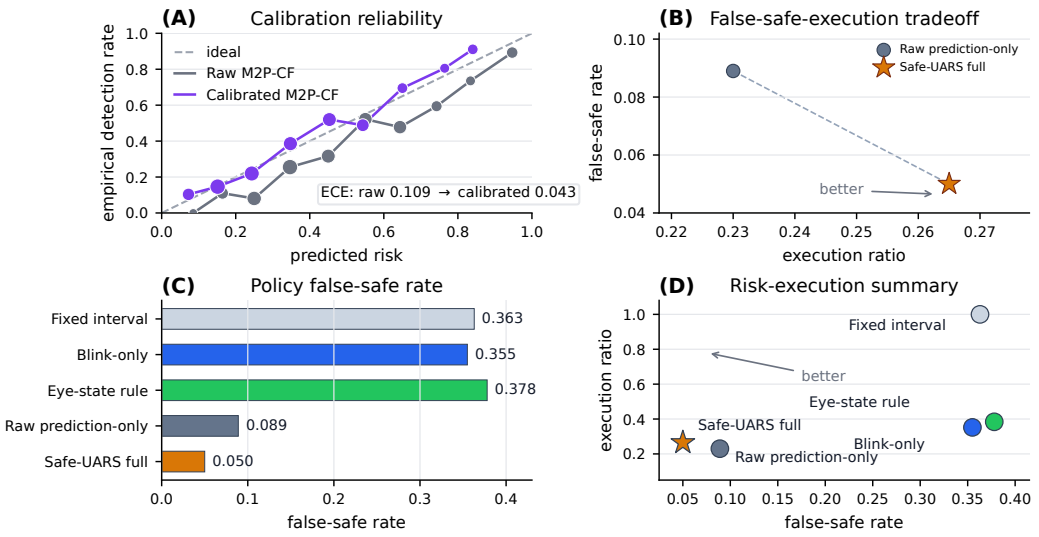

Figure S7: Calibration and logged-event selection diagnostics. (A) Reliability. (B) False-safe–execution tradeoff. (C) Selected-event FSR. (D) FSR versus ER. Replay panels retain original recorded-update labels; they do not estimate outcomes of hypothetical modified updates.

A.11 Model Variant Configurations

All four configurations use the same pre-update input modalities, three one-layer 128-dimensional forward GRU encoders, a 128-dimensional wide branch, and the $512 \to 256 \to 128$ fusion network. Core summarizes each temporal stream by its last valid GRU state. The other configurations use update-conditioned attention. Table S23 gives the operational differences specified by the training configuration.

Table S23: Operational model configurations. Every row uses five independently trained members (seeds 11, 22, 33, 44, 55), selected separately by validation AUROC. BCE is class-weighted.

Variant	**Temporal**	**Attention**	**Wide**	**Training objectives**	K
M2P-core	GRU	No	Yes	BCE + abruptness	5
M2P-attn	GRU	Yes	Yes	BCE + abruptness	5
BCE-only	GRU	Yes	Yes	BCE only	5
M2P-CF	GRU	Yes	Yes	BCE + CF + invariance + ranking + abruptness	5

M2P-core and BCE-only differ in both temporal aggregation and abruptness supervision: core removes event attention but retains the abruptness objective; BCE-only retains the full forward architecture while setting all auxiliary weights to zero. M2P-attn and M2P-CF share their forward architecture, with the latter enabling CF, invariance, and ranking in addition to abruptness. Single-objective ablations retain the full architecture and set only the named auxiliary weight to zero; the abruptness weight remains 0.05.

Each member is selected using its validation AUROC. The five sigmoid probabilities are averaged for ensemble prediction. Native AUROC/AUPRC/F1 are distinguished from beta-calibrated probabilities in the reported tables. FS@20 follows the five-run, separately calibrated fixed-coverage definition in the main text. Model names therefore specify both the forward computation and the training objective, not an interchangeable set of basic-model labels.

A.11.1 Baseline Input and Selection Configuration

The tabular feature matrix has one row per event: it contains the supplied pre-update numerical summaries, missingness indicators, and encoded categorical scene/update attributes, not a flattened 60-step sequence. Numerical standardization is fitted on training rows. The temporal loader concatenates eye and head sequences, plus context sequences when present in the cache; it transfers training-derived normalization to validation and test. Its default cache construction retains up to the last 60 samples in the 500-ms pre-update window and masks padded samples.

Table S24: Baseline input and model-selection configuration. Native denotes the probabilities reported in Table 1; calibration is statistical post-processing, not a different predictor. Sequence and tabular inputs differ in temporal representation.

Predictor	**Temporal representation**	**Explicit update attrs.**	**Table 1 probability**	**Fitting / selection**
Logistic regression	Event summary vector	Yes	Native	Fixed estimator; training convergence
Random forest	Event summary vector	Yes	Native	Fixed estimator configuration
XGBoost	Event summary vector	Yes	Native	Fixed estimator configuration
CatBoost	Event summary vector	Yes	Native	Fixed estimator configuration
MLP	Event summary vector; hidden widths 128, 64	Yes	Native	Training-loss stopping; max. 400 iterations
Temporal Transformer	Cached pre-update sequence; masked temporal mean	No separate update vector	Native	Validation AUROC; patience 6
FT-Transformer	One token per event-level feature; CLS pooling	Yes	Native	Validation AUROC; patience 6
M2P-CF	Three 60-step GRUs; update-conditioned attention	Yes	Native / +Beta	Validation AUROC; patience 20

Both Transformer baselines use two encoder layers, width 64, four attention heads, feed-forward width 192, GELU, and dropout 0.1. The temporal model adds learned positional embeddings and applies a valid-step attention mask before masked-mean pooling. FT-Transformer embeds each scalar feature through a feature-specific affine map and predicts from a learned CLS token. Their training uses class-weighted BCE, AdamW with learning rate 10^{-3} and weight decay 10^{-4}, and gradient clipping at 5.0; validation AUROC selects these neural models.

The baseline evaluation implementation exports native, temperature-scaled, beta-calibrated, and isotonic probabilities, each calibrator fitted on validation data only. Temperature scaling uses $p_i^{\mathrm{temp}} = \sigma(\ell_i / T_{\mathrm{cal}})$, where $T_{\mathrm{cal}} > 0$ is fitted by minimizing validation negative log-likelihood. It is a calibration alternative, not a stage preceding the final beta calibrator. Table 1 uses the native rows except for the explicitly labeled M2P-CF +Beta row. F1 thresholds are selected on validation predictions. The ECE reduction from 0.109 to 0.043 therefore describes calibration within M2P-CF. The prediction comparison includes differences in both representation and temporal processing; it is not an identical-input architecture ablation.