BOUNDARY-STABLE POLICYFLIP: TRAINING POLICY-SENSITIVE SAFETY GUARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety guards deployed under changing policies must sometimes reverse decisions when trusted state changes even if the underlying event does not. We introduce *Boundary-Stable PolicyFlip*, a fine-tuning objective for matched policy-state interventions. Writing a pair by its unsafe–safe score gap g and score center c, a strict threshold crossing requires $g > 2|c|$: conventional pairwise ranking controls g but leaves c unconstrained. We therefore optimize the weaker-endpoint slack $b = g/2 - |c| = \min(-d_{\text{safe}}, d_{\text{unsafe}})$. At the default settings, its penalty is exactly worst-endpoint cross-entropy, yielding a threshold-aware aggregation of the same endpoint labels. BS-PF combines this boundary term with transition ranking; BG-PF additionally attenuates ranking as a pair moves inside the correct decision regions. Neither changes the guard architecture or inference path. On a controlled policy-intervention benchmark, BS-PF improves Qwen3.5-4B balanced accuracy over pointwise training by 5.25–6.50 percentage points across four raw-input evaluations. Against a ranking-plus-paired-CE control on Policy OOD, BS-PF improves transition direction by 4.08 points (95% interval [2.69, 5.47]). On a fresh machine-rendered holdout, BG-PF improves pair exactness by 2.30 points and transition direction by 5.25 points [3.04, 7.42] over Pair-CE. With unchanged loss constants on Llama-Guard-3-8B, it improves Policy-OOD BAcc over gap-only training by 7.46 points and pair exactness by 12.07 points. Across 12,434 public moderation records, aggregate macro BAcc remains within 0.03 points of Decision-only (85.27% versus 85.24%). Overall, threshold-aware pair optimization strengthens policy-transition tracking under a fixed training-step budget without adding inference-time machinery.

1 INTRODUCTION

An upload to an approved destination may become disallowed when that destination is removed from a trusted registry. The same pattern appears when an OAuth scope expires, a tool capability is revoked, a session changes privilege, or an application switches the active safety mode while the user request itself is unchanged. For a concrete example, consider the event "send `report.pdf` to destination D." If D is present in an application-supplied trusted-destination registry, the event can be safe; after the same registry entry is revoked, the unchanged event becomes unsafe. The supervision of interest is therefore not a new event label in isolation, but a controlled decision transition induced by trusted state.

We place the learned guard upstream of an authoritative policy engine. Deterministic enforcement handles resolved policy-relevant fields, while the learned component maps free-form prompts, model responses, retrieved context, or tool traces to the facts that drive the decision. Controlled state interventions then test whether fine-tuning makes this semantic front end track the governing state consistently. The benchmark is not intended to replace deterministic membership or authorization checks; it isolates the training problem of making a learned guard respond correctly when policy-relevant state changes while the event itself is held fixed.

Learned guards commonly use pointwise safe/unsafe supervision (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025). Policy-conditioned systems extend this interface to user-defined or compositional rules (Hoover et al., 2026; Li et al., 2026; Zhang et al., 2026). We study how to train the decision function when matched examples identify a specific state change that reverses the label. These pairs

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.

make the intended transition explicit and let the objective allocate pressure jointly across its two endpoints.

A natural auxiliary objective ranks the unsafe endpoint above the safe endpoint. We call this gap-only relation loss POLICYFLIP. Its geometric ambiguity is simple: scores $(d_{\text{safe}}, d_{\text{unsafe}}) = (1, 3)$ and $(-1, 1)$ have the same directional gap, but only the second pair crosses the classifier threshold at zero. More generally, if $g = d_{\text{unsafe}} - d_{\text{safe}}$ and $c = (d_{\text{unsafe}} + d_{\text{safe}})/2$, a strict safe/unsafe crossing occurs exactly when $g > 2|c|$. Ranking controls separation g but is blind to the common translation c. Pointwise cross-entropy supplies an absolute classification signal, but matched supervision gives us an opportunity to allocate that signal at the pair level. The design question is therefore not simply whether to add endpoint labels, but how to focus optimization on the endpoint that limits a valid threshold crossing.

We propose *Boundary-Stable PolicyFlip*. It combines transition ranking with the weaker endpoint margin,

$$b = \min(-d_{\text{safe}}, d_{\text{unsafe}}). \tag{1}$$

A positive b certifies a strict safe/unsafe crossing. At the default settings, the proposed boundary penalty equals the maximum of the two endpoint cross-entropies. Unlike their average, this pair-level aggregation directs its boundary update toward the endpoint with the larger classification loss. BS-PF adds this penalty to transition ranking. BG-PF further uses the boundary state to weight the ranking gradient, placing less emphasis on pairs whose endpoints already lie well inside their correct regions.

We evaluate the training rule under a fixed budget and matched Qwen LoRA initializations. Seven objectives distinguish pointwise training, matched endpoint CE, ranking, worst-endpoint optimization, and their combinations. The compiler deliberately uses exact state interventions and opaque identifiers: exact labels support controlled objective-level comparisons, while opaque names suppress pretrained lexical cues such as "approved" or "blocked." BS-PF improves BAcc over Decision-only on all four raw-input splits and has the highest mean on three. Against the exposure-matched PF+PAIRCE control, both boundary-aware variants improve transition direction, the quantity directly targeted by threshold-aware pair training. We then test representation transfer with machine-rendered inputs, a second guard backbone with canonical structured input, and a public moderation transfer check.

The paper makes three contributions. First, it gives a pair-coordinate account of policy transitions: ranking determines separation g, while correct threshold placement additionally requires $g > 2|c|$. This yields the weaker-endpoint slack $b = g/2 - |c|$ and an exact worst-endpoint-CE surrogate for crossing failure. Second, it introduces additive and boundary-gated objectives that use this geometry with existing guard scores and add no inference-time module. Third, it evaluates the objective with exposure-matched endpoint CE, ranking-plus-CE, boundary-only controls, matched initializations, representation shifts, and a second backbone, providing complementary comparisons of endpoint placement and transition ordering. The result is a training method for learned guards that must interpret events against trusted state while leaving final authorization to the application policy layer.

2 BOUNDARY-STABLE POLICYFLIP

2.1 MATCHED TRANSITIONS AND GUARD SCORES

Let x denote event facts, s trusted policy-relevant state, and v presentation variables. A renderer produces $z = R(x, s, v)$, and a policy function supplies $y = G(x, s) \in \{0, 1\}$, where 1 denotes unsafe. A matched transition is

$$p = (z_l, y_l, z_r, y_r), \qquad y_l \neq y_r, \tag{2}$$

with non-policy facts fixed and one governing state variable changed. The experimental compiler balances both flip directions within each profile/factor combination. The loss itself requires only the paired inputs and their endpoint labels.

For a tokenized candidate label c, the guard uses its length-normalized log-likelihood,

$$a_\theta(c; z) = \frac{1}{|c|} \sum_{t=1}^{|c|} \log p_\theta(c_t \mid z, c_{<t}), \qquad d_\theta(z) = a_\theta(\texttt{unsafe}; z) - a_\theta(\texttt{safe}; z). \tag{3}$$

The score d_θ is relative to the decision threshold zero. The two-candidate event loss is

$$\ell_{\mathrm{dec}}(z, y) = \mathrm{softplus}\big(-(2y-1)d_\theta(z)\big). \tag{4}$$

This is CE over the two candidate scores; it does not require generating a free-form response.

2.2 Crossing geometry and a threshold-aware surrogate

For pair p, let $t_p = y_r - y_l$ and $\Delta_p = d_\theta(z_r) - d_\theta(z_l)$. The ranking term is

$$\ell_{\mathrm{PF}}(p) = \mathrm{softplus}(m - t_p\Delta_p), \qquad m = 1. \tag{5}$$

Reordering endpoints by gold label gives $g = d_{\mathrm{unsafe}} - d_{\mathrm{safe}} = t_p\Delta_p$ and center $c = (d_{\mathrm{unsafe}} + d_{\mathrm{safe}})/2$. A valid strict transition requires $d_{\mathrm{safe}} < 0 < d_{\mathrm{unsafe}}$. In pair coordinates, its limiting slack is

$$\boxed{b = \frac{g}{2} - |c| = \min(-d_{\mathrm{safe}}, d_{\mathrm{unsafe}}).} \tag{6}$$

Consequently,

$$b > 0 \iff d_{\mathrm{safe}} < 0 < d_{\mathrm{unsafe}} \iff g > 2|c|. \tag{7}$$

Ranking sees only g and is invariant to the common translation c, although shifting c can place both endpoints on one side of the threshold. Thus b is threshold-derived crossing slack, not a generic hard-example score.

For a desired endpoint margin $\gamma \geq 0$,

$$b \geq \gamma \iff d_{\mathrm{safe}} \leq -\gamma \text{ and } d_{\mathrm{unsafe}} \geq \gamma. \tag{8}$$

We penalize insufficient crossing slack with

$$\ell_{\mathrm{B}}(p) = \tau\ \mathrm{softplus}\left(\frac{\gamma - b}{\tau}\right). \tag{9}$$

All experiments use $\gamma = 0$ and $\tau = 1$.

Threshold-aware endpoint aggregation. At the default settings, substituting Equation 6 into Equation 9 gives

$$\ell_{\mathrm{B}} = \mathrm{softplus}\Big(|c| - \frac{g}{2}\Big) = \max(\ell_{\mathrm{safe}}, \ell_{\mathrm{unsafe}}), \qquad (\gamma = 0,\ \tau = 1), \tag{10}$$

where $\ell_{\mathrm{safe}} = \mathrm{softplus}(d_{\mathrm{safe}})$ and $\ell_{\mathrm{unsafe}} = \mathrm{softplus}(-d_{\mathrm{unsafe}})$. Thus worst-endpoint CE follows from the crossing geometry rather than defining it. Pair-CE averages the same endpoint losses; away from ties, ℓ_{B} instead updates the endpoint limiting the crossing.

In pair coordinates the optimization effect is explicit. Away from $c = 0$, if $q = \sigma(|c| - g/2)$, then $\partial\ell_{\mathrm{B}}/\partial g = -q/2$ and $\partial\ell_{\mathrm{B}}/\partial c = \mathrm{sign}(c)q$, so gradient descent both enlarges the gap and recenters the pair toward the threshold. Mean Pair-CE also depends on both coordinates but distributes endpoint gradients additively. The boundary loss additionally upper-bounds failure of a strict crossing,

$$\mathbf{1}\{b \leq 0\} \ \leq\ \frac{\ell_{\mathrm{B}}}{\log 2}, \tag{11}$$

because $\ell_{\mathrm{B}} \geq \log 2$ whenever $b \leq 0$. Appendix B gives the derivation. A correct crossing already implies the right direction; combining boundary placement with ranking shapes finite-budget optimization without adding supervision beyond the matched endpoint labels. Section 4 tests these roles separately.

2.3 Additive and gated training

Let L_{dec} denote the event-stream loss and $L_{\mathrm{PF}}, L_{\mathrm{B}}$ the corresponding matched-pair averages. The additive objective is

$$L_{\mathrm{BS}} = L_{\mathrm{dec}} + \lambda(L_{\mathrm{PF}} + \beta L_{\mathrm{B}}). \tag{12}$$

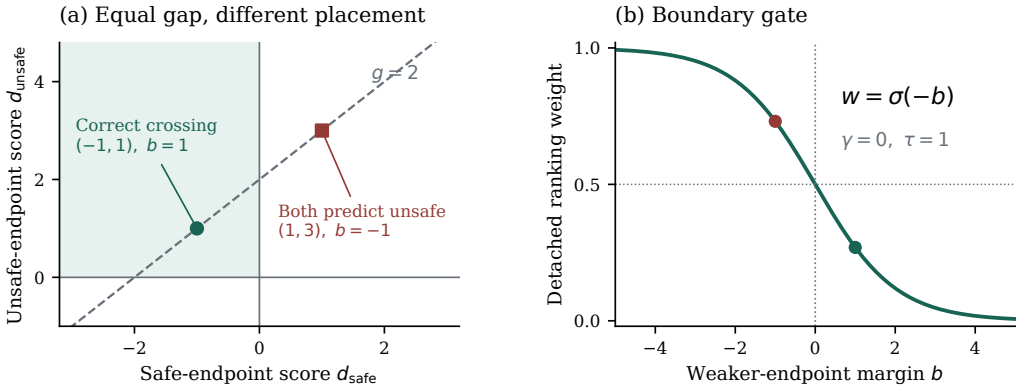

Figure 1: **Ranking and crossing are different geometric requirements.** Left: the two marked pairs have the same gap, $g = 2$, but only the point in the upper-left quadrant has $b > 0$. Right: the BG-PF weight decreases with the weaker endpoint margin. This is a schematic of the objective, not an empirical result.

The gated variant assigns each pair the detached weight

$$w_p = \mathrm{stopgrad}\left[\sigma\left(\frac{\gamma - b_p}{\tau}\right)\right], \qquad L_{\mathrm{BG}} = L_{\mathrm{dec}} + \lambda\left(\mathbb{E}_p[w_p \ell_{\mathrm{PF}}(p)] + \beta L_{\mathrm{B}}\right). \tag{13}$$

The gate approaches one for a strongly misplaced endpoint, equals $1/2$ at $b = \gamma$, and decreases as both endpoints move into their correct regions. Detaching w_p treats it as a fixed weight during each update. At the default settings, it equals the magnitude of the boundary loss derivative with respect to b. The gate therefore uses the same pair difficulty measure as the boundary penalty.

The softplus ranking term already has a gradient that decays with increasing gap. BG-PF adds a separate dependence on endpoint placement: two pairs with the same gap can receive different ranking weights (Figure 1). It smoothly attenuates ranking updates after crossing.

The two variants are pre-specified optimization regimes evaluated uniformly across all splits. BS-PF keeps transition-ordering pressure active throughout training, whereas BG-PF attenuates it as the weaker endpoint becomes well placed. Reporting both tests whether reducing ranking pressure after correct endpoint placement helps finite-budget adaptation and exposes the resulting tradeoff across representations.

We use $\lambda = 0.9$, $\beta = 1$, $\gamma = 0$, and $\tau = 1$. The relation weight and ranking margin follow the gap-only configuration; $\gamma = 0$ is exactly the classifier threshold, and unit β and τ keep the boundary term on the native candidate-score scale. These constants remain fixed across the raw, rendered-holdout, and Llama studies. Implementing the method requires matched transitions at training time and scalar operations on existing candidate scores. Inference uses the same guard and scoring rule.

3 Evaluation Protocol

Policy-state benchmark. Each latent program specifies a mechanism context, event surface, language, presented primary/scope codes, their registries, and one of four opaque policy profiles implementing fixed Boolean functions. These latent variables abstract deployment primitives such as allowlist membership, authorization scope, and policy mode; a matched example changes one governing state factor while holding the event fixed. Gold labels are computed by deterministic code from profile semantics and membership state. The profile mappings are learned during training and stay fixed across evaluation. Training contains 4,608 programs; dev and each of the four evaluation splits contain 768 programs. Appendix A gives the factorial construction and a concrete state transition.

The opacity is intentional. Randomized profile and code names remove semantic shortcuts from tokens such as “approved,” “revoked,” or “malicious,” so a model cannot solve the controlled task

from identifier sentiment alone. The compiler is therefore a controlled abstraction for exact state interventions and objective-level comparisons, not a claim that literal set membership should be delegated to a neural model. A separate holdout uses machine-rendered presentations generated from fresh latent policy states, and the public moderation suite tests compatibility with conventional prompt-safety inputs. Inputs omit program/event IDs, labels, relation roles, generator identifiers, and mechanism-bearing administrative fields. Fixed pre-training checks cover metadata and character features, shuffled labels, and zero-shot performance; all satisfy the protocol thresholds (Appendix A).

Predefined shifts. Table 1 defines the evaluation axes. We retain the protocol names Attack OOD, Policy OOD, and Tool OOD. They denote held-out mechanism wording, policy-state representation, and event surface, respectively. In particular, Policy OOD is a policy-state representation shift under fixed policy semantics: it changes registry cardinality and code morphology while preserving the policy functions. Together, the splits probe generalization across mechanism wording, policy-state representation, and event surface under fixed policy semantics.

Table 1: Raw-input shifts under fixed compiler policy semantics. Training uses four mechanism contexts and three event surfaces; the split names follow the experimental protocol.

Split	Held-out axis	Construction
Test	New programs + evaluation-template family	Program-disjoint; no train-template reuse
Attack OOD	Held-out mechanisms	injection_provenance, content_harm; absent from train
Policy OOD	Registry/code morphology shift	Registry cardinality 7–8 vs. 3–4; distinct opaque-code morphology
Tool OOD	Held-out event surface	tool_call; absent from train

Training and controls. Qwen3.5-4B (Qwen Team, 2026) is the main backbone. For each seed (4307, 4317, 4327), all seven arms load the same deterministic fresh LoRA initialization (Hu et al., 2022). Training uses a common 320-step optimizer budget, BF16, learning rate 10^{-5}, 5% warmup, weight decay 0.01, gradient accumulation 8, and the fixed step-320 checkpoint. Held-out scores do not select checkpoints. Auxiliary pair exposure is matched among the pair-supervised arms. The seven arms are Decision-only; gap-only PF, which adds L_{PF} to Decision CE; Pair-CE, which adds matched endpoint CE; BS-PF; BG-PF; PF+PAIRCE, which replaces L_{B} in Equation 12 with mean endpoint CE; and Boundary-only, which retains L_{B} without L_{PF}. The two latter control objectives were fixed before their training runs. Pair-CE uses the same one-pair auxiliary exposure as the transition arms. Appendix B lists the optimization settings and objective structure.

Rendered holdout. After fixing the method and Qwen checkpoints, a separate run generates 4,608 fresh latent programs with seed 9917. GLM-4.7-Flash (Z.ai, 2026) renders their inputs, and Nemotron-Cascade-2-30B-A3B (NVIDIA Nemotron Team, 2026) verifies them. Both receive public specifications without private gold directions. The pipeline accepts 4,429 candidates (96.12%); balanced selection retains 768 programs and 1,536 events, covering every directional cell once. The selected set is disjoint from earlier diagnostic data at the program, event, rendering, query-IR, and public-specification levels. We call this set the rendered holdout (External-B) and use it to test representation shift under the same latent policy semantics. Gold labels are produced independently by deterministic compiler code; Appendix A.2 illustrates the underlying compiler-state intervention for a selected pair.

Second backbone. The Llama-Guard-3-8B study (Meta, 2024) uses canonical EVENTIR inputs and the same objective constants. Decision-only and PF use existing checkpoints from the 320-step protocol. Pair-CE, BS-PF, and BG-PF use matched fresh LoRA initializations for seeds 5307, 5317, and 5327. We report Policy OOD in the main text because it directly evaluates a shift in policy-state representation under fixed policy semantics; Appendix H provides per-seed Policy-OOD results and structured-input ceiling references.

Metrics and uncertainty. Balanced accuracy is $\mathrm{BAcc} = (r_{\mathrm{unsafe}} + r_{\mathrm{benign}})/2$. Pair exactness requires both endpoints to be classified correctly. Direction accuracy checks $t_p\Delta_p > 0$, and signed margin averages $t_p\Delta_p$; boundary margin averages b_p. Direction and endpoint correctness capture distinct requirements: a pair can have the right ordering without placing both endpoints on the correct sides of the threshold. We therefore report direction together with BAcc and pair exactness. The two margins characterize score geometry on each model's native scale. We report three-seed means, 5,000-draw program-clustered paired bootstrap comparisons, and hierarchical intervals that resample observed seeds and program IDs. Intervals characterize this three-run design; empirical p values are reported at the comparison level. Appendix C describes the resampling units, and Appendices E and H give per-seed results.

4 RESULTS

4.1 TRAINING ON RAW POLICY-STATE INPUTS

Table 2: Qwen3.5-4B raw-input BAcc (%). Means over three matched-initialization seeds. Bold marks the highest mean; paired intervals appear in Appendix C.

Method	Test	Attack OOD	Policy OOD	Tool OOD
Decision-only	89.94	87.43	86.36	90.38
Gap-only PF	94.30	92.48	90.43	93.62
Pair-CE	94.02	92.70	**92.01**	94.27
PF+PairCE	94.24	92.66	89.64	94.58
Boundary-only	88.97	87.27	84.06	89.44
BS-PF	**95.55**	**93.93**	91.61	**95.79**
BG-PF	94.98	91.82	90.84	94.89

Pair-CE is a strong matched-label baseline (94.02/92.70/92.01/94.27% BAcc). BS-PF improves over Decision-only by +5.61 [1.39, 8.73], +6.50 [1.64, 10.30], +5.25 [1.60, 9.33], and +5.41 [3.06, 7.69] points. To isolate aggregation beyond matched endpoint supervision, we use PF+PAIRCE, which also shares pair exposure and transition ranking; Pair-CE endpoint comparisons appear in Appendix C.

BS-PF and PF+PAIRCE differ only in worst- versus mean-endpoint CE. On Policy OOD, their BAcc difference is +1.98 points [-0.91, 5.08], while BS-PF improves direction by 4.08 points [2.69, 5.47] and signed margin by 2.13 [0.88, 3.83]. BG-PF, which additionally gates ranking from the same boundary state, improves direction by 5.64 points [2.43, 8.33]. Thus transition direction shows the clearest separation from additive supervision in this control.

Boundary-only asks whether boundary aggregation suffices without ranking. Adding ranking raises Policy-OOD BAcc from 84.06% to 91.61% for BS-PF (+7.55 [2.64, 10.81]) and direction by 8.68 points [4.08, 15.76]. Together, these controls support complementary roles under the fixed 320-step protocol: boundary aggregation allocates endpoint supervision at the pair level; ranking supplies transition-ordering pressure. Appendix D reports the full comparisons.

4.2 MACHINE-RENDERED REPRESENTATION SHIFT

Table 3: Qwen rendered-holdout results against compiler labels. BAcc, Pair, and Direction are percentages; Margin and b are score units. Means over three seeds.

Method	BAcc	Pair	Direction	Margin	Boundary b
Decision-only	73.48	57.20	77.69	9.80	0.83
Gap-only PF	70.96	51.52	70.79	8.85	0.34
Pair-CE	72.83	54.82	75.35	8.18	0.09
BS-PF	73.48	**57.60**	77.08	10.71	1.38
BG-PF	**74.54**	57.12	**80.60**	**11.19**	**1.44**

Under the rendering shift, BG-PF reaches the highest mean direction accuracy and the highest mean BAcc (Table 3). It improves over gap-only PF by 9.81 direction points and 3.58 BAcc points [0.63, 6.81]. BS-PF also improves over PF, by 6.08 pair-exactness points and 2.52 BAcc points. Boundary-aware training therefore improves on ranking-only auxiliary supervision under this representation change.

Against Pair-CE, BG-PF improves pair exactness by 2.30 points ($p_{\text{emp}} = .0260$) and direction by 5.25 points [3.04, 7.42], while mean BAcc is 1.71 points higher. Pair exactness requires both members of a matched transition to be classified correctly, so these pair-level gains directly measure whether the intended transition survives the representation shift.

The direct gate ablation is BG-PF versus BS-PF: direction improves by 3.52 points [0.26, 7.77], while mean BAcc increases by 1.06 points. This pattern is consistent with the intended role of the gate: reducing residual ranking pressure as endpoint placement improves. The appendix retains every per-seed result and the complete reported comparisons.

4.3 Transfer to Llama-Guard-3-8B

Table 4: Llama-Guard-3-8B Policy OOD with canonical EventIR input. BAcc is mean ± sample SD over three seeds; Pair is pair exactness. Margin and b are score units.

Method	BAcc	Pair	Margin	Boundary b
Decision-only	93.00±5.33	84.64	17.48	6.44
Gap-only PF	89.48±3.97	80.69	17.49	6.10
Pair-CE	93.97±6.77	84.72	19.85	7.67
BS-PF	93.95±5.71	88.11	19.53	7.84
BG-PF	96.94±1.64	92.75	20.45	8.82

With unchanged loss constants, BG-PF reaches 96.94% mean BAcc and 92.75% pair exactness on Llama Policy OOD (Table 4). Its BAcc gain over PF is +7.46 points [3.79, 12.72], positive in all three seeds (+12.97, +5.70, +3.70), and pair exactness improves by 12.07 points. BG-PF also has the highest mean BAcc and pair exactness among the five methods. Because the loss constants transfer unchanged, this study tests the objective in a second model/representation setting using the same loss configuration. Appendix H reports per-seed Policy-OOD results and the External-B comparison.

4.4 Conventional moderation transfer

We evaluate the seven Qwen arms on 12,434 prompt-safety records from Aegis2, PolyGuard English/Chinese, ToxicChat, WildGuardTest, and XSTest (Ghosh et al., 2025; Kumar et al., 2025; Lin et al., 2023; Han et al., 2024; Röttger et al., 2024). Macro BAcc is the unweighted mean over these five evaluation sets and spans 84.67–85.27%. BG-PF reaches 85.27% macro BAcc after policy-transition fine-tuning, within 0.03 points of the 85.24% Decision-only value. Appendix F gives dataset provenance, counts, and all dataset-level results.

5 Related work

Policy-conditioned guards. Llama Guard, WildGuard, and Qwen3Guard train learned safety classifiers (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025). DynaGuard supports user-defined policies at inference (Hoover et al., 2026); LPG uses latent policy reasoning (Li et al., 2026); SafePyramid studies compositional policy guardrailing (Zhang et al., 2026); and PL-Guard separates neural grounding from probabilistic rule inference (Chatterji et al., 2026). Recent ICLR work also broadens guard modeling through knowledge-enhanced reasoning, severity-aware moderation, harmful-data augmentation for guard distillation, confidence calibration, and policy-following multimodal guardrails (Kang & Li, 2025; Yin et al., 2025; Lee et al., 2025; Liu et al., 2025; Chen et al., 2025). Our focus is complementary: matched changes in trusted state under a governing policy, with the policy function held fixed so that the training geometry of decision reversals can be studied directly.

Matched policy supervision. PolicyShiftGuard's BP-Adapt combines pass/block pairs, endpoint supervision, and a hinge pairwise margin for image guardrails (Song et al., 2026). We build on the value of matched policy decisions and study a different pair aggregation: the weaker endpoint's margin relative to the classification threshold, with an optional boundary-dependent ranking weight. PF+PAIRCE instantiates endpoint supervision plus transition ranking as an internal control, providing an exposure-matched additive comparison for threshold-aware aggregation in the text-guard setting. Counterfactual augmentation and behavioral testing likewise use controlled changes to expose label-relevant features (Kaushik et al., 2020; Ribeiro et al., 2020; Joshi & He, 2022).

Pairwise preference and metric objectives. Pairwise and triplet losses learn relative order or distance from matched examples, including metric-learning triplets and preference objectives such as DPO and IPO (Schroff et al., 2015; Rafailov et al., 2023; Azar et al., 2024). These methods motivate learning from comparisons, but their relative signal does not by itself anchor a pair to an externally meaningful classifier threshold. In our setting both endpoints have absolute safe/unsafe labels, and the failure mode is a common translation of their scores that preserves order. Equation 7 isolates this difference: ranking depends on g, whereas a correct policy flip additionally requires controlling $|c|$.

Hard-example and worst-case optimization. Online hard-example mining selects difficult training examples, and focal loss reduces the weight of easy examples (Shrivastava et al., 2016; Lin et al., 2017). Group DRO instead optimizes worst-group performance (Sagawa et al., 2020). The identity in Equation 10 connects our boundary surrogate to this family, while the pair-coordinate view identifies the relevant "hard" endpoint as the one limiting the inequality $g > 2|c|$. BG-PF then uses the same weaker-endpoint state to modulate ranking, yielding a threshold-anchored coupling of endpoint placement and transition ordering for matched policy interventions.

6 DISCUSSION AND CONCLUSION

Using policy transitions in training. Applications can construct matched supervision from changes to registry membership, scope, approval, or delegated authority while holding the event fixed. Pairing such a change with the same event exposes the decision transition that a policy-sensitive guard should learn. Boundary-Stable PolicyFlip turns this structure into training signal: the worst-endpoint term concentrates boundary pressure on the endpoint that limits a valid crossing, while BG-PF attenuates additional ranking pressure as the pair becomes well placed. The method requires matched transitions only during fine-tuning and adds no inference-time module.

The learned guard and deterministic policy layer have complementary roles. The policy layer remains authoritative for resolved fields and runtime enforcement; the learned guard handles semantic interpretation when policy-relevant facts must be recovered from free-form prompts, responses, retrieved context, or tool traces. In the controlled benchmark, the compiler fixes downstream semantics so that the experiments isolate this training problem under exact interventions. Opaque identifiers reduce lexical priors, the machine-rendered holdout changes presentation while preserving latent policy semantics, and the public suite separately checks conventional moderation after adaptation.

Empirical findings. On the controlled benchmark, BS-PF improves raw-input BAcc over pointwise training across all four splits. Comparisons with exposure-matched additive pair supervision show gains in Policy-OOD transition direction and rendered-holdout pair behavior. With the same loss constants, the Llama study yields a 7.46-point Policy-OOD BAcc gain over gap-only ranking. The public moderation suite provides a separate check of conventional safety classification after adaptation. Together, these evaluations probe endpoint placement, transition ordering, presentation shift, and transfer across two guard backbones while retaining the original inference path.

BS-PF and BG-PF are pre-specified optimization regimes that differ in how they allocate ranking pressure. BS-PF maintains it throughout adaptation, whereas BG-PF attenuates it as the weaker endpoint becomes well placed. Their relative performance varies across representations, so we report both rather than selecting a single variant post hoc. All objective constants remain fixed across the reported evaluations, and the Qwen comparisons use a common 320-step optimizer protocol.

Boundary-Stable PolicyFlip is a fine-tuning objective for safety guards that must respond to trusted policy-state changes. Its central observation is geometric: pairwise ranking controls separation, but a valid policy flip also requires the pair center to remain close enough to the classifier threshold for the endpoints to occupy opposite decision regions. Optimizing the slack $g/2 - |c|$ incorporates this threshold-crossing requirement directly into training; at the default settings, the resulting boundary penalty is exactly worst-endpoint cross-entropy. Combining this term with transition ranking yields threshold-aware pair optimization using existing guard scores, with no additional inference-time component. The method can be used in standard guard fine-tuning whenever matched policy-state interventions are available.

REPRODUCIBILITY STATEMENT

The appendix records the compiler setting, split definitions, objective constants, optimization settings, and reported resampling protocol. Reported training comparisons use fixed step-320 checkpoints. Qwen arms share a deterministic fresh LoRA initialization within each seed; the Llama checkpoint protocol is described separately. Per-seed rendered-holdout and Llama Policy-OOD tables accompany the aggregate comparisons. The accompanying manuscript source contains the LaTeX files, table fragments, bibliography, and method figure needed to rebuild the paper.

ETHICS STATEMENT

This work studies offline safety classification under application-supplied policy and trusted state. The compiler-generated benchmark contains no human-subject data. Public moderation data may contain harmful text and are used for safety evaluation. A trained guard supplies classification signals; runtime authorization and enforcement remain responsibilities of the deterministic policy layer.

AI USE STATEMENT

We used generative AI tools to assist with literature search, methodology and experimental-design feedback, implementation, result analysis, mathematical exposition, and language editing and polishing of the manuscript. Generative models also rendered and verified the machine-rendered holdout without access to gold labels, which were computed independently by deterministic code. The authors reviewed the AI-assisted work, made all final research decisions, and take responsibility for the paper and supporting artifacts.

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455, 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Satchit Chatterji, Shihan Wang, Giovanni Sileno, and Erman Acar. PL-Guard: Probabilistic logic reasoning for LLM guardrails. *arXiv preprint arXiv:2608.15673*, 2026. URL https://arxiv.org/abs/2608.15673.

Zhaorun Chen, Francesco Pinto, Minzhou Pan, and Bo Li. SafeWatch: An efficient safety-policy following video guardrail model with transparent explanations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/beac6bfb7eac3d651307c16ac747df01-Abstract-Conference.html.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of*

the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 5992–6026, 2025. doi: 10.18653/v1/2025.naacl-long.306. URL https://aclanthology.org/2025.naacl-long.306/.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. *arXiv preprint arXiv:2406.18495*, 2024. URL https://arxiv.org/abs/2406.18495.

Monte Hoover, Vatsal Baherwani, Neel Jain, Khalid Saifullah, Joseph J. Vincent, Chirag Jain, Melissa Rad, C. Bayan Bruss, Ashwinee Panda, and Tom Goldstein. DynaGuard: A dynamic guardian model with user-defined policies. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8d2aad53fa148791ffd004a7c9cce5ad-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023. URL https://arxiv.org/abs/2312.06674.

Nitish Joshi and He He. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3668–3681, 2022. doi: 10.18653/v1/2022.acl-long.256. URL https://aclanthology.org/2022.acl-long.256/.

Mintong Kang and Bo Li. R^2-Guard: Robust reasoning enabled LLM guardrail via knowledge-enhanced logical reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a07e87ecfa8a651d62257571669b0150-Abstract-Conference.html.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1909.12434.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. PolyGuard: A multilingual safety moderation tool for 17 languages. *arXiv preprint arXiv:2504.04377*, 2025. URL https://arxiv.org/abs/2504.04377.

Seanie Lee, Haebin Seong, Dong Bok Lee, Minki Kang, Xiaoyin Chen, Dominik Wagner, Yoshua Bengio, Juho Lee, and Sung Ju Hwang. HarmAug: Effective data augmentation for knowledge distillation of safety guard models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ac2d4db1615bf3736f44a1b4889e666b-Abstract-Conference.html.

Nanxi Li, Zhengyue Zhao, and Chaowei Xiao. LPG: Balancing efficiency and policy reasoning in latent policy guardrails. *arXiv preprint arXiv:2605.17329*, 2026. URL https://arxiv.org/abs/2605.17329.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. URL https://arxiv.org/abs/1708.02002.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 4694–4702, 2023. doi: 10.18653/v1/2023.findings-emnlp.311. URL https://aclanthology.org/2023.findings-emnlp.311/.

Hongfu Liu, Hengguan Huang, Xiangming Gu, Hao Wang, and Ye Wang. On calibration of LLM-based guard models for reliable content moderation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a99f732df9b668284b449da0214a3286-Abstract-Conference.html.

Meta. Llama Guard 3 8B model card, 2024. URL https://huggingface.co/meta-llama/Llama-Guard-3-8B.

NVIDIA Nemotron Team. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation, 2026. URL https://research.nvidia.com/labs/nemotron/nemotron-cascade-2/.

Qwen Team. Qwen3.5-4B model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1911.08731.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015. URL https://arxiv.org/abs/1503.03832.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://arxiv.org/abs/1604.03540.

Mingyang Song, Luxin Xu, Haoyu Sun, Minzhou Pan, Yu Cheng, and Bo Li. PolicyShift-Guard: Benchmarking and improving policy-adaptive image guardrails. *arXiv preprint arXiv:2607.05910*, 2026. URL https://arxiv.org/html/2607.05910v1.

Fan Yin, Philippe Laban, Xiangyu Peng, Yilun Zhou, Yixin Mao, Vaibhav Vats, Linnea Ross, Divyansh Agarwal, Caiming Xiong, and Chien-Sheng Wu. BingoGuard: LLM content moderation tools with risk levels. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/782b6152c04e9948c2cb3833e9a288ef-Abstract-Conference.html.

Z.ai. GLM-4.7-Flash model card, 2026. URL https://huggingface.co/zai-org/GLM-4.7-Flash.

Jiacheng Zhang, Haoyu He, Sen Zhang, Shen Wang, Xiaolei Xu, Yuhao Sun, Meng Shen, and Feng Liu. SafePyramid: A hierarchical benchmark for in-context policy guardrailing. *arXiv preprint arXiv:2606.29887*, 2026. URL https://arxiv.org/abs/2606.29887.

Haiquan Zhao, Chenhan Yuan, Fei Huang, et al. Qwen3Guard technical report. *arXiv preprint arXiv:2510.14276*, 2025. URL https://arxiv.org/abs/2510.14276.

A BENCHMARK CONSTRUCTION AND OOD CONTRACTS

A.1 LATENT-STATE COMPILER

Each program contains one opaque policy profile, one risk mechanism, one event surface, one language, one policy-flip factor, and one flip direction. The hidden policy profiles implement four Boolean functions over membership of the presented primary and scope codes in their corresponding registries. The profile names themselves are opaque (PX7, QM4, RV9, TN2), and the Boolean mapping is not rendered into model input. The learned mappings are held fixed across all compiler splits; changes in registry state can change the label without changing the policy function. Gold labels are computed by deterministic code. For every program the compiler emits an anchor event, a surface-equivalent event, a nuisance event, a causal sibling, and a policy-flip sibling; the formal method experiments use the event stream and the policy-flip relations.

Training contains 4,608 programs. Dev, test, Attack OOD, Policy OOD, and Tool OOD contain 768 programs each. The program counts are integer multiples of the pre-specified factorial cells, which balances profile, mechanism, event surface, language, flip factor, and flip direction.

A.2 A COMPILER-DEFINED MATCHED STATE TRANSITION

Table 5 summarizes the compiler state underlying one selected rendered-holdout pair. The operation token is `OMHBJMRA6PA`, the policy profile is `PX7`, and the presented primary code is `PJTEB6TEA5K`. The scope code `SN9RTCGXNPN` remains in its registry. Only membership of the presented primary code changes. The table is a field summary, not a replacement input used for evaluation.

Table 5: State fields of the illustrated policy transition.

Field	Safe endpoint	Unsafe endpoint
Primary registry	`PCK4CTPYPNQ`, `PJTEB6TEA5K`, `PT7JF24QQZG`, `PYKVAFFKX3C`	`PCK4CTPYPNQ`, `PT7JF24QQZG`, `PYKVAFFKX3C`
Presented primary code in list	Yes	No
Presented scope code in list	Yes	Yes
Compiler label	Safe	Unsafe

Compiler-state interpretation. Under policy profile `PX7`, the safe endpoint contains primary code `PJTEB6TEA5K` in its primary registry, whereas the unsafe endpoint does not. The operation token, policy profile, presented codes, and scope membership remain fixed. Table 5 therefore illustrates a single governing-state intervention and its deterministic compiler labels. It summarizes the latent state rather than reproducing or validating the natural-language evaluation input. Rendering fidelity is a separate property assessed by the renderer/verifier pipeline described in Appendix E.

A.3 RAW-INPUT SPLIT DEFINITIONS

Table 6 expands the split contract summarized in the main text.

Table 6: Raw-input split construction.

Split	Training support	Evaluation shift
Test	Four train mechanisms; user-prompt, model-response, retrieved-context surfaces	Program-disjoint examples rendered with an evaluation-template family not used for training.
Attack OOD	Train mechanisms exclude injection provenance and content harm	Evaluation uses only `injection_provenance` and `content_harm`; these mechanism contexts never occur in the training split.
Policy OOD	Base opaque-code morphology; registry cardinality 3–4 after membership insertion	Registry cardinality 7–8 and a distinct lowercase/length code morphology; the hidden Boolean profile semantics are unchanged.
Tool OOD	User-prompt, model-response, retrieved-context	Evaluation uses only `tool_call`, which is absent from the training surfaces.

The canonical EVENTIR view removes scenario wording while retaining the same policy state. Consequently, Attack OOD on raw input tests unseen mechanism language, whereas the corresponding structured-input split mainly tests whether the canonical state representation removes that surface shift. The primary Qwen study uses raw input; the Llama study uses canonical EVENTIR input.

A.4 PRE-TRAINING SHORTCUT CHECKS

Before the reported training runs, fixed checks evaluate whether administrative metadata, shallow lexical features, shuffled labels, or the untrained base model can predict the benchmark labels. Because the opaque profile mappings are learned from supervision, zero-shot performance provides a reference level. Together, these checks cover the predefined metadata, character-level, label-shuffle, and base-model shortcut classes.

Table 7: Pre-training shortcut checks. All thresholds were fixed before the reported training runs.

Check	Observed	Threshold	Result
Metadata-only BAcc (max)	50.00	< 58	Pass
Raw char 3–5gram BAcc (max)	54.48	< 62	Pass
IR char 3–5gram BAcc (max)	50.31	< 62	Pass
Shuffled-label BAcc (range)	49.74–51.25	42–58	Pass
Zero-shot Raw BAcc	51.53	< 75	Pass
Zero-shot IR BAcc	49.95	< 75	Pass

B TRAINING PROTOCOL

B.1 MATCHED INITIALIZATION AND OPTIMIZATION

All Qwen3.5-4B arms for a given seed load the same deterministic fresh LoRA initialization. The LoRA rank is 32, alpha is 64, and dropout is zero. The three Qwen seeds are 4307, 4317, and 4327. Training uses eight GPUs, BF16, a common budget of 320 optimizer steps, a fixed step-320 checkpoint, learning rate 10^{-5}, weight decay 0.01, 5% warmup, gradient accumulation 8, and maximum sequence length 768. No held-out result is used for checkpoint selection. This design evaluates every objective at the same optimizer-step checkpoint; pair-supervised arms additionally share the same one-pair auxiliary exposure schedule.

Table 8: Objective constants for Boundary-Stable PolicyFlip.

Parameter	Value	Protocol rationale
Relation weight λ	0.90	Inherited from the gap-only-PF configuration.
PF margin m	1.0	Inherited from the gap-only-PF configuration.
Boundary weight β	1.0	Unit coefficient for the boundary term.
Boundary target γ	0.0	Matches the classifier decision threshold.
Boundary temperature τ	1.0	Uses the same score scale as the PF margin.

The five primary training arms are Decision-only, gap-only POLICYFLIP, Pair-CE, BS-PF, and BG-PF. Pair-CE uses the same one-pair auxiliary exposure as the transition arms and applies endpoint-label cross-entropy. We additionally evaluate two targeted objective-structure controls whose definitions were fixed before their training runs. PF+PAIRCE uses $\ell_{\mathrm{dec}}+\lambda(\ell_{\mathrm{PF}}+\beta\,\mathrm{mean}(\ell_{\mathrm{safe}},\ell_{\mathrm{unsafe}}))$, combining transition ordering with conventional additive endpoint CE. Boundary-only uses $\ell_{\mathrm{dec}}+\lambda\beta\ell_{\mathrm{B}}$ and removes the transition term. The seven Qwen arms separately test pointwise supervision, matched endpoint supervision, transition ordering, coupled boundary control, and boundary gating.

B.2 BOUNDARY SURROGATE AND GRADIENTS

Write $a=-d_{\mathrm{safe}}$ and $u=d_{\mathrm{unsafe}}$. The identity $\min(a,u)=(a+u-|a-u|)/2$ gives Equation 6. Also, $\min(a,u)\geq\gamma$ exactly when both $a\geq\gamma$ and $u\geq\gamma$, proving Equation 8. At the default parameters, softplus is increasing, so

$$\mathrm{softplus}(-b)=\mathrm{softplus}(\max(d_{\mathrm{safe}},-d_{\mathrm{unsafe}}))=\max(\ell_{\mathrm{safe}},\ell_{\mathrm{unsafe}}). \tag{14}$$

If $b\leq 0$, this loss is at least $\log 2$; if $b>0$, the crossing-failure indicator vanishes. This proves Equation 11. The bound treats a score exactly at the boundary conservatively and does not depend on which candidate wins an exact tie in the implementation.

The score-level boundary gradients are

$$(\partial_{d_{\mathrm{safe}}}\ell_{\mathrm{B}},\partial_{d_{\mathrm{unsafe}}}\ell_{\mathrm{B}})=\begin{cases}(\sigma(d_{\mathrm{safe}}),0), & \ell_{\mathrm{safe}}>\ell_{\mathrm{unsafe}},\\ (0,-\sigma(-d_{\mathrm{unsafe}})), & \ell_{\mathrm{unsafe}}>\ell_{\mathrm{safe}}.\end{cases} \tag{15}$$

At a tie, the max/min is subdifferentiable; an autodiff implementation chooses a subgradient. These formulas describe the boundary term's score gradient. Shared model parameters, the event loss, and the ranking loss jointly determine the actual parameter update.

The pair-coordinate form makes the threshold anchoring explicit. Substituting $d_{\mathrm{safe}}=c-g/2$ and $d_{\mathrm{unsafe}}=c+g/2$ gives

$$\ell_{\mathrm{B}}=\mathrm{softplus}\Big(|c|-\frac{g}{2}\Big). \tag{16}$$

Hence, away from $c=0$ and with $q=\sigma(|c|-g/2)$,

$$\frac{\partial\ell_{\mathrm{B}}}{\partial g}=-\frac{q}{2},\qquad \frac{\partial\ell_{\mathrm{B}}}{\partial c}=\mathrm{sign}(c)q. \tag{17}$$

The boundary term therefore both increases endpoint separation and moves the pair center toward the decision threshold. By contrast, $\ell_{\mathrm{PF}}=\mathrm{softplus}(m-g)$ has zero derivative with respect to c. This is the geometric distinction illustrated in Figure 1.

Let $\ell_{\mathrm{pairCE}}=(\ell_{\mathrm{safe}}+\ell_{\mathrm{unsafe}})/2$. Since both CE losses are nonnegative,

$$\ell_{\mathrm{pairCE}}\leq\ell_{\mathrm{B}}\leq 2\ell_{\mathrm{pairCE}}. \tag{18}$$

The control at unit β therefore compares the two aggregation rules with the same coefficient $\beta=1$, while each objective retains its native gradient geometry. For the gated variant,

$$-\frac{\partial\ell_{\mathrm{B}}}{\partial b}=\sigma\Big(\frac{\gamma-b}{\tau}\Big)=w \tag{19}$$

before detachment. In contrast, the magnitude of the ranking derivative with respect to gap is $\sigma(m-g)$. The two weights depend on different pair properties: gap and weaker-endpoint placement.

Table 9: Objective structure. The PolicyShiftGuard row summarizes its reported BP-Adapt objective (Song et al., 2026) as a structural literature reference; empirical comparisons use the internal controls.

Method	Matched pair	Additive Pair-CE	Pair ordering	Worst-endpoint boundary
Pair-CE	✓	✓	–	–
Gap-only POLICYFLIP	✓	–	✓	–
PF+PAIRCE	✓	✓	✓	–
Boundary-only	✓	–	–	✓
PolicyShiftGuard BP-Adapt	✓	✓	✓	–
BS-PF	✓	–	✓	✓
BG-PF	✓	–	gated	✓

C STATISTICAL PROTOCOL AND CONTROLLED COMPARISONS

For per-seed comparisons, the bootstrap resamples program IDs and keeps all events/relations for a sampled program together. Cross-seed comparisons use the reported hierarchical resampling of training seeds and program IDs, with 5,000 draws. The reported intervals describe variation over the three observed runs and their evaluation programs. Empirical p values are reported at the comparison level under this resampling protocol, with intervals as the primary uncertainty summary. Per-seed rendered-holdout and Llama Policy-OOD results accompany the aggregate estimates.

Table 10: Core BAcc differences between the boundary-aware variants and exposure-matched Pair-CE. Differences are percentage points; intervals reflect both seed and program variation.

Comparison	Split	ΔBAcc	95% CI	p_{emp}
BS-PF − Pair-CE	Test	+1.53	[-2.89, +7.36]	0.5440
BS-PF − Pair-CE	Attack OOD	+1.23	[-3.56, +7.56]	0.6664
BS-PF − Pair-CE	Policy OOD	-0.40	[-6.17, +5.00]	0.8312
BS-PF − Pair-CE	Tool OOD	+1.52	[-3.36, +6.36]	0.6076
BG-PF − Pair-CE	Test	+0.96	[-4.99, +7.02]	0.7552
BG-PF − Pair-CE	Attack OOD	-0.88	[-7.51, +5.29]	0.7600
BG-PF − Pair-CE	Policy OOD	-1.17	[-5.53, +6.06]	0.5292
BG-PF − Pair-CE	Tool OOD	+0.62	[-4.61, +5.88]	0.7844

Table 10 reports endpoint-level effect estimates against exposure-matched Pair-CE. Table 11 separately reports structural comparisons against Boundary-only and PF+PAIRCE; transition direction and pair exactness provide complementary measures of pair behavior, while score margins are retained as geometry diagnostics in the targeted comparisons.

D TARGETED OBJECTIVE-STRUCTURE CONTROLS

Two additional Qwen arms probe the structure of the proposed loss. Boundary-only retains the coupled worst-endpoint term but removes transition ordering. PF+PAIRCE combines the original transition term with conventional additive endpoint CE. Their definitions, seeds, optimization schedule, and comparison plan are fixed before these control runs.

Table 11: Policy-OOD hierarchical effects for the targeted controls. BAcc and Direction are percentage-point differences; Margin is in score units. Brackets give 95% hierarchical bootstrap intervals over seeds and programs.

Comparison	ΔBAcc	ΔDirection	ΔMargin
BS-PF − Boundary-only	+7.55 [+2.64, +10.81]	+8.68 [+4.08, +15.76]	+4.75 [+3.41, +5.78]
BG-PF − Boundary-only	+6.78 [+3.41, +11.51]	+10.24 [+7.07, +14.37]	+4.03 [+2.96, +4.89]
BS-PF − PF+PairCE	+1.98 [-0.91, +5.08]	+4.08 [+2.69, +5.47]	+2.13 [+0.88, +3.83]
BG-PF − PF+PairCE	+1.21 [-2.56, +6.26]	+5.64 [+2.43, +8.33]	+1.40 [-1.53, +3.20]

Combining threshold-relative endpoint placement with transition ordering yields +7.55 BAcc and +8.68 direction points for BS-PF over Boundary-only, while BG-PF yields +6.78 and +10.24 points, respectively. Against the additive PF+PAIRCE control, BS-PF improves direction by 4.08 points and signed margin by 2.13 units under matched pair exposure; the additional effect is concentrated in transition geometry, matching the threshold-crossing quantity targeted by the proposed objective.

E EXTERNAL HOLDOUT

The fresh rendered holdout is created after the boundary-aware design and final Qwen checkpoints are fixed. Its latent seed is 9917. A gold-blind renderer/verifier pipeline processes 4,608 candidate programs; 4,429 are verified, and the balanced compiler selects 768 programs covering all 768 directional cells exactly once. Gold labels are never provided to either rendering model. The final set contains fresh latent programs and machine-rendered inputs under the same compiler-defined policy semantics and was not used to choose Qwen checkpoints or objective constants. Appendix A.2 gives the underlying compiler-state intervention for a selected pair. External-B is evaluated against deterministic compiler labels, and all reported values use the frozen evaluated inputs.

Table 12: External-holdout results by seed. BAcc, Pair, and Direction are percentages; Margin is in score units.

Method	Seed	BAcc	Pair	Direction	Margin
Decision-only	4307	71.61	53.26	74.87	8.65
Decision-only	4317	77.73	63.28	82.42	11.65
Decision-only	4327	71.09	55.08	75.78	9.09
Gap-only PF	4307	70.05	49.74	72.66	8.54
Gap-only PF	4317	70.25	49.74	66.15	8.36
Gap-only PF	4327	72.59	55.08	73.57	9.67
Pair-CE	4307	70.90	54.82	75.91	7.83
Pair-CE	4317	74.09	53.65	72.01	7.52
Pair-CE	4327	73.50	55.99	78.12	9.20
BS-PF	4307	71.61	54.30	73.70	9.69
BS-PF	4317	71.94	56.12	74.35	10.74
BS-PF	4327	76.89	62.37	83.20	11.71
BG-PF	4307	76.95	58.07	81.64	10.99
BG-PF	4317	70.96	55.34	75.65	10.87
BG-PF	4327	75.72	57.94	84.51	11.72

Table 12 reports each training seed separately. The hierarchical analysis resamples both seed and program ID so that the aggregate intervals reflect the variation visible across runs and programs.

Table 13: Seed-aware hierarchical effects on the external holdout. BAcc, Pair, and Direction are percentage-point differences; Margin is in score units.

Comparison	ΔBAcc	ΔPair	ΔDirection	ΔMargin
BS-PF − Gap-only PF	+2.52	+6.08	+6.29	+1.86
BG-PF − Gap-only PF	+3.58	+5.60	+9.81	+2.34
BG-PF − Pair-CE	+1.71	+2.30	+5.25	+3.01
BG-PF − BS-PF	+1.06	-0.48	+3.52	+0.48
BG-PF − Decision	+1.06	-0.09	+2.91	+1.40

Comparison	p_{BAcc}	p_{Pair}	$p_{\text{Direction}}$	p_{Margin}
BS-PF − Gap-only PF	.0012	$< .0002$	.0056	$< .0002$
BG-PF − Gap-only PF	.0080	$< .0002$	$< .0002$	$< .0002$
BG-PF − Pair-CE	.4896	.0260	$< .0002$	$< .0002$
BG-PF − BS-PF	.6224	.8148	.0248	.2404
BG-PF − Decision	.5964	.9904	.5240	.1396

F CONVENTIONAL PROMPT-SAFETY TRANSFER

We freeze a public moderation suite before evaluating the seven Qwen systems. It contains 12,434 evaluation records: Aegis2 (1,915; Ghosh et al., 2025), PolyGuard English/Chinese (3,398; Kumar et al., 2025), ToxicChat (4,972; Lin et al., 2023), WildGuardTest (1,699; Han et al., 2024), and XSTest (450; Röttger et al., 2024). These sets evaluate conventional prompt moderation, without matched policy-state transitions.

PolyGuardPrompts derives from WildGuard test data and multilingual translations (Kumar et al., 2025), so the English/Chinese subset and WildGuardTest have related provenance. The reported total therefore counts evaluation records, and the five-set macro is used as an aggregate transfer summary across these evaluation sets.

Table 14: Public prompt-safety balanced accuracy (%). Macro is the unweighted mean over the five datasets; values are means over three Qwen seeds.

Method	Macro	Aegis2	PolyGuard	ToxicChat	WildGuard	XSTest
Decision-only	85.24	76.89	83.52	**90.43**	85.81	89.53
Gap-only PF	85.02	76.40	83.92	89.77	86.25	88.77
Pair-CE	85.06	77.62	**84.06**	89.42	**86.30**	87.90
PF+PairCE	84.67	76.41	83.79	89.80	86.15	87.17
Boundary-only	85.15	**77.64**	83.19	88.47	85.70	**90.75**
BS-PF	84.83	77.12	83.66	90.35	85.69	87.33
BG-PF	**85.27**	77.47	83.75	88.99	86.15	89.97

Macro BAcc spans 84.67–85.27% across all seven arms. BG-PF reaches 85.27%, within 0.03 points of the 85.24% Decision-only value. Table 14 gives the full dataset-level transfer profile.

G EXTERNAL PAIR CORRECTNESS BY MECHANISM

Pair exactness requires both endpoints to be correct and is therefore stricter than direction accuracy. Table 15 reports the three-seed mean within each mechanism. Both boundary-aware variants exceed gap-only PF in every mechanism stratum; the table also shows the mechanism-specific pair-exactness profile.

Table 15: External pair exactness (%) by risk mechanism.

Mechanism	Decision	PF	Pair-CE	BS-PF	BG-PF
capability_authorization	57.03	51.82	51.30	55.47	55.73
content_harm	60.68	53.91	61.98	59.90	59.11
injection_provenance	60.68	53.65	57.55	61.20	59.38
intent_boundary	54.17	45.57	45.83	52.60	50.78
privacy_exfiltration	54.95	53.12	56.51	57.81	58.33
trajectory_composition	55.73	51.04	55.73	58.59	59.38

H STRUCTURED-INPUT AND SECOND-BACKBONE EXPERIMENTS

The evaluation suite also includes a canonical structured EVENTIR representation. Decision-only training nearly saturates endpoint BAcc under this representation, while transition-aware training continues to alter relation margins. Table 16 provides this Qwen structured-input reference.

Table 16: Qwen3.5-4B structured-input ceiling. Values are three-seed means.

Split	Dec. BAcc	PF BAcc	Δ	Dec. margin	PF margin	Δ
Test	99.87	99.98	+0.11	20.19	**23.63**	**+3.44**
Attack OOD	99.92	99.97	+0.05	20.23	**23.62**	**+3.39**
Policy OOD	97.56	98.26	+0.70	18.84	**21.29**	**+2.45**
Tool OOD	99.96	99.90	-0.06	20.25	**23.42**	**+3.17**

H.1 Five-arm Llama study

The boundary-aware objective is additionally evaluated on Llama-Guard-3-8B with the canonical EventIR representation used by the second-backbone protocol. Decision-only and gap-only PolicyFlip use existing checkpoints from the same 320-step protocol. Pair-CE, BS-PF, and BG-PF are trained from matched fresh LoRA initializations for seeds 5307, 5317, and 5327 with the same 320-step schedule and fixed objective constants as the Qwen study. Policy OOD is the main second-backbone comparison because it directly changes policy-state representation under fixed policy semantics; per-seed Policy-OOD results and the External-B ceiling comparison are reported below.

Table 17: Llama-Guard-3-8B Policy-OOD BAcc (%) by seed.

Seed	Decision	PF	Pair-CE	BS-PF	BG-PF
5307	87.16	85.49	97.73	99.11	98.46
5317	97.60	89.51	98.02	87.81	95.21
5327	94.24	93.44	86.15	94.92	97.14

Table 18: Llama Policy-OOD hierarchical effects; BAcc and Pair are percentage-point differences.

Comparison	ΔBAcc	95% CI	p_{BAcc}	ΔPair	p_{Pair}
Gap-only PF − Decision-only	-3.52	[-7.84,-0.52]	0.0056	-3.95	0.1864
BS-PF − Gap-only PF	+4.47	[-1.62,+13.41]	0.3232	+7.42	0.5724
BG-PF − Gap-only PF	+7.46	[+3.79,+12.72]	$< .0002$	+12.07	0.0092
BG-PF − Pair-CE	+2.97	[-2.63,+10.68]	0.4944	+8.03	0.2836
BG-PF − BS-PF	+2.99	[-0.59,+7.15]	0.0888	+4.64	0.2724

BG-PF improves 7.46 BAcc points over gap-only PF under Policy OOD (95% CI [3.79, 12.72]) and improves pair exactness by 12.07 points. The BAcc effect is positive in every seed: +12.97, +5.70, and +3.70 points for seeds 5307, 5317, and 5327. Mean BAcc rises from 89.48% with gap-only PF to 96.94% with BG-PF. The External-B results below provide a structured-input ceiling reference.

H.2 Llama on the external holdout

We evaluate all five Llama systems on the canonical EventIR view of External-B.

Table 19: Llama-Guard-3-8B on the canonical EventIR view of the rendered holdout. Values are three-seed means.

Method	BAcc	Pair	Direction	Margin
Decision-only	99.50	99.00	100.00	20.99
Gap-only PF	99.70	99.39	100.00	23.06
Pair-CE	99.96	99.91	99.96	24.75
BS-PF	99.74	99.48	100.00	23.57
BG-PF	99.91	99.83	100.00	22.88

All arms reach at least 99.5% mean BAcc, so Table 19 is a ceiling reference; Policy OOD is the main second-backbone comparison because it directly instantiates the protocol-defined policy-state representation shift.