INTERVENTION EFFECTS ARE EXECUTION-RELATIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation-level interventions are commonly compared and tuned through downstream task performance. Yet an intervention-induced state does not directly produce a task output: it is converted by a final execution rule, such as thresholding, decoding, suppression, or candidate scoring. We ask whether conclusions about an intervention family remain stable when this downstream rule changes while the intervened states are held fixed. We call this dependence execution-relative intervention effects and introduce matched-state evaluation, which computes each intervention state once and reuses the same cached state across task-valid executors. Across sound, language, vision, and speech tasks, changing only the executor can alter the measured effect size and, in stronger cases, reverse the ordering of intervention settings or change the selected setting. In sound event detection, fixed-threshold and hysteresis execution give opposite local orderings to the same intervention transition and select different settings under cross-fitting; transferring the fixed-threshold choice to hysteresis incurs a 0.0147 Event-F1 loss on held-out data. Boundary manipulations trace the fixed-threshold drop to sparse crossings of event-formation boundaries, revealing candidate instantiation as one mechanism. A complementary Qwen3-8B study keeps candidate statistics fixed and changes only the scoring rule, showing how candidate scoring reshapes competitive margins and intervention responses while preserving the selected coefficient. These results show that intervention evaluation and selection should account for the executor that turns intervened states into task outputs, and should test whether comparative conclusions remain stable across plausible task-valid execution rules.

1 INTRODUCTION

Representation-level interventions are increasingly used to analyze and control learned models by modifying internal states at inference time (Li et al., 2023a; Turner et al., 2023; Rimsky et al., 2024). Their effects are usually compared through downstream task performance: one steering strength is reported to outperform another, and the best-performing setting is often carried forward to later evaluation or deployment. But an intervention does not directly produce a task-level prediction. Its state passes through a final state-to-output mapping before the metric is computed. Continuous predictions may be thresholded into events, candidate answers ranked, boxes suppressed, or sequences decoded (Meister et al., 2020; Bodla et al., 2017; Cances et al., 2019). We call this final mapping the executor. This raises a basic evaluation question: if the intervention-induced states stay fixed, do comparisons among intervention settings stay the same when only the executor changes? The question matters because tuning, ranking, and selecting intervention settings all occur after this final mapping. The same upstream state change can therefore support different task-level comparisons depending on how it is executed.

For an intervention setting λ and executor D, let $U_D(\lambda)$ denote the resulting task utility. For two settings i and j, define

$$\Delta_D^{i \to j} = U_D(\lambda_j) - U_D(\lambda_i), \tag{1}$$

which measures the change from setting λ_i to λ_j under executor D. The executor index matters: even when the model, input, intervention, data, task, metric, and intervention-induced states are fixed, changing only D can change the magnitude of $\Delta_D^{i \to j}$, reverse its sign, or change which setting is selected from the family. Let λ_D^* denote the selected setting under executor D. If $\lambda_{D_1}^* \neq \lambda_{D_2}^*$, the same intervention family leads to different choices under the two executors. We refer to this dependence as execution relativity.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

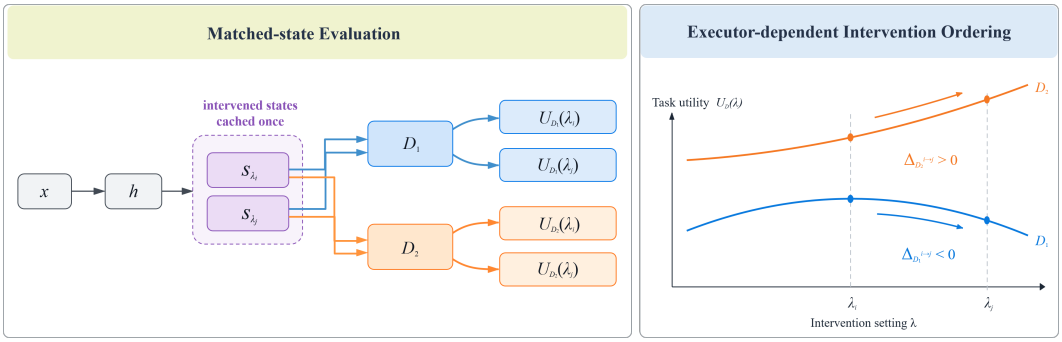


Figure 1: Matched-state evaluation and ordering relativity. Left: cached intervention states are reused across task-valid executors. Right: the same transition can decrease utility under D_1 and increase it under D_2.

We isolate execution relativity with matched-state evaluation. Each intervention state s_λ is computed once, cached, and then passed to every executor. The model, input, intervention, data, task, metric, and upstream state are therefore identical; only the final state-to-output rule changes. This design lets us compare executors without recomputing or perturbing the intervention itself. We distinguish three levels of consequence. Effect-size relativity means that executors measure different changes while preserving the same preference. Ordering relativity means that they prefer different settings for at least one comparison. Selection relativity means that they ultimately choose different settings from the intervention family. These levels separate a numerical change in measured effect from a change in the conclusion drawn from the comparison. Prior work has shown that post-processing and decoding rules can change task outputs and evaluation scores (Cances et al., 2019; Ebbers et al., 2023; Meister et al., 2020). Our focus is different: we ask whether those rules change the comparison among intervention settings when the upstream states are identical.

The experiments separate these outcomes and then trace where the dependence enters the state-to-output mapping. In sound event detection (SED), fixed thresholding and hysteresis reverse the local ordering of one intervention transition and select different settings. Targeted boundary manipulations show that the fixed-threshold drop is caused by sparse changes in which events are formed, identifying candidate instantiation as one mechanism. We then move the executor later in the pipeline: on Qwen3-8B (Yang et al., 2025), the candidate set and cached candidate statistics stay fixed, while only the scoring rule changes. The resulting changes in competitive margins and intervention response reveal candidate scoring as a second mechanism, although both scorers retain the same selected coefficient. Segmentation and detection extend the formation-boundary pattern to spatial outputs, while CTC decoding provides another case in which execution changes effect size without changing the selected setting.

Contributions.

- We formulate execution-relative intervention effects and distinguish three levels at which an executor can matter: measured effect size, pairwise ordering, and the setting selected from an intervention family.
- We introduce matched-state evaluation, which reuses identical cached intervention states across task-valid executors so that differences in the intervention response can be attributed to the final execution rule.
- We identify two mechanisms. Candidate instantiation explains how executor-specific formation boundaries can change which task objects exist; candidate scoring explains how different comparison rules can reshape competition among already-fixed alternatives.
- We validate these mechanisms across temporal-event, language, spatial, and sequence outputs. SED reaches ordering and selection changes, while Qwen3-8B and CTC show cases in which execution dependence remains at the level of the measured response.

2 Related Work

Representation-level interventions and behavioral evaluation. Representation-level interventions modify internal activations to alter model behavior. Inference-Time Intervention, Activation Addition, and Contrastive Activation Addition steer behavior through hidden-state changes (Li et al., 2023a; Turner et al., 2023; Rimsky et al., 2024). Recent work extends this paradigm through activation transport, conditional steering, feedback control, dynamic rejection, fine-grained activation steering, distribution-matched interchange interventions, and decoding-integrated control (Rodriguez et al., 2025; Lee et al., 2025; Nguyen et al., 2026; Kang and Kim, 2026; Feng et al., 2026; Bao et al., 2026; Su et al., 2025). Other work analyzes steering dynamics and preference–utility tradeoffs across intervention strength (Wang et al., 2025; Xu et al., 2026). Across these methods, downstream behavior remains the primary basis for comparing intervention settings and selecting operating points. Methodological work shows that activation-intervention conclusions can depend on evaluation metrics, corruption choices, and representation choices (Zhang and Nanda, 2024; Antverg and Belinkov, 2022). Work on subspace interventions, causal abstraction, model editing, and probing further shows that behavioral conclusions depend on how representations and interventions are operationalized (Makelov et al., 2024; Geiger et al., 2025; Meng et al., 2022; Rauch et al., 2026). We study a complementary source of dependence at the end of the pipeline: with the upstream intervention states fixed, we ask how the executor changes the measured response of the intervention family.

Execution, post-processing, and comparative evaluation. Downstream execution shapes task outputs throughout structured prediction. Holtzman et al., Meister et al., and Li et al. study decoding and candidate-comparison rules in language generation (Holtzman et al., 2020; Meister et al., 2020; Li et al., 2023), while Bodla et al. show how candidate suppression changes object-detection outcomes through Soft-NMS (Bodla et al., 2017). In sound event detection, Cances et al. study post-processing effects on event-level results, while PSDS and post-processing-independent evaluation reduce dependence on specific thresholds and post-processing settings (Cances et al., 2019; Bilen et al., 2020; Ebbers et al., 2023). Xie et al. show that guidance comparisons in text-to-image generation depend on evaluation calibration (Xie et al., 2026). Related statistical work formalizes interaction and effect heterogeneity when measured effects depend on evaluation context (VanderWeele and Knol, 2014; Athey and Imbens, 2016; Imai and Ratkovic, 2013). Our unit of analysis is the intervention-family response $U_D(\lambda)$ under matched upstream states. Rather than asking only whether the absolute task score changes, we ask whether changing the executor changes the measured transition, the ordering among settings, or the setting selected from the family.

3 Execution-Relative Intervention Effects

An intervention family is evaluated through a composition: the intervention produces a decision state, and an executor maps that state to a task output,

$$x \to h \to s_\lambda \to D(s_\lambda) \to \hat{y}. \tag{2}$$

Here, h is the hidden representation, s_λ is the decision state under intervention setting λ, and D is a task-valid executor. We consider an executor family $\mathcal{D}$ determined by the task and evaluation protocol. Its members preserve the task semantics, metric, and execution-time information, but use different state-to-output rules. Examples include fixed thresholding versus hysteresis in SED, alternative candidate-scoring rules in multiple-choice reasoning, and greedy versus beam decoding in CTC (Cances et al., 2019; Meister et al., 2020; Graves et al., 2006). The central object is therefore not s_λ alone, but the response curve produced by the composition $D(s_\lambda)$ as λ varies.

We write task utility as

$$U_D(\lambda) = u(D(s_\lambda), y), \tag{3}$$

where higher values of u indicate better performance. For lower-is-better metrics, we use negative utility unless reporting the raw error. As defined in Eq. 1, $\Delta_D^{i\to j}$ denotes the measured transition between settings λ_i and λ_j under executor D. Across two executors, the interaction contrast is

$$\mathcal{J}_{D_2-D_1}^{i\to j} = \Delta_{D_2}^{i\to j} - \Delta_{D_1}^{i\to j}. \tag{4}$$

A nonzero $\mathcal{J}$ means that the same intervention transition is measured differently under the two executors. This can be a purely quantitative difference. Ordering relativity is stronger: the two

executors disagree on which intervention setting is better,

$$\Delta_{D_1}^{i\to j}\Delta_{D_2}^{i\to j} < 0. \tag{5}$$

The same pair of intervention states can favor different settings under different executors. This is a change in ordering, not just effect magnitude. Ordering is defined within each executor: the two can prefer opposite settings for a transition even when one achieves higher absolute utility throughout the family.

Execution can also change the setting selected from the full intervention family. Using a deterministic tie-breaking rule τ shared across executors, define

$$\lambda_D^* = \tau\left(\arg\max_{\lambda\in\Lambda} U_D(\lambda)\right). \tag{6}$$

Selection is execution-dependent when $\lambda_{D_1}^* \neq \lambda_{D_2}^*$. The three levels therefore form a natural hierarchy of consequences: effect-size relativity changes how large an intervention effect appears; ordering relativity changes which of two settings is preferred; selection relativity changes the setting carried forward from the full family. A numerical difference need not change an ordering, and a local ordering difference need not change the global optimum. This hierarchy lets us distinguish sensitivity of a metric from sensitivity of the scientific or practical conclusion.

For a task-valid executor family $\mathcal{D}$, pairwise ordering is invariant when

$$\operatorname{sign}\left(\Delta_D^{i\to j}\right) = \operatorname{sign}\left(\Delta_{D'}^{i\to j}\right), \qquad D, D' \in \mathcal{D}, \tag{7}$$

for every compared pair, and selection is invariant when $\lambda_D^* = \lambda_{D'}^*$. An intervention family can exhibit substantial executor-dependent effect sizes while preserving both ordering and selection. Matched-state evaluation reveals where these invariances break while holding the upstream intervention states, model, input, task, data, and metric fixed.

Stability of ordering, formation, and scoring. Ordering and selection are preserved whenever the utilities under two executors are related by a strictly increasing transform,

$$U_{D_2}(\lambda) = g(U_{D_1}(\lambda)), \qquad \lambda \in \Lambda, \tag{8}$$

because g preserves every pairwise comparison over Λ. Thus, an executor can change absolute utility without changing any comparative conclusion. Rankings can change only when the executor affects different intervention settings differently enough to distort the response curve rather than merely transform it monotonically.

The two mechanisms studied below make this non-uniformity concrete. For candidate instantiation, suppose membership of a task object under executor D is determined by a continuous formation score $f_D(s_\lambda)$,

$$z_D(\lambda) = \mathbf{1}[f_D(s_\lambda) \geq 0]. \tag{9}$$

If $\lambda \mapsto f_D(s_\lambda)$ is continuous on an interval $I \subseteq \Lambda$ and

$$\inf_{\lambda\in I} |f_D(s_\lambda)| > 0, \tag{10}$$

then $z_D(\lambda)$ is constant on I. The relevant quantity is therefore not perturbation size alone, but distance to the executor's formation boundary. When intervention-induced states remain well inside one region, object membership is stable. Near a boundary, however, a small continuous state change can create or remove an object and, in structured outputs, merge or fragment discrete task objects. Candidate-instantiation sensitivity should therefore concentrate at these boundary crossings.

For candidate scoring, let a^* be the highest-scoring candidate under executor D at λ_i, with margin

$$m_D = S_D(a^*; \lambda_i) - \max_{b\neq a^*} S_D(b; \lambda_i) > 0. \tag{11}$$

For the transition $\lambda_i \to \lambda_j$, define

$$\delta_D(a) = S_D(a; \lambda_j) - S_D(a; \lambda_i). \tag{12}$$

If

$$\max_{b\neq a^*} [\delta_D(b) - \delta_D(a^*)] < m_D, \tag{13}$$

then a^* remains the top-ranked candidate. The inequality has a simple interpretation: the original winner is protected as long as no competitor gains enough relative score to close the starting margin. Large margins therefore make the ranking stable, whereas small margins make it easier for executor-specific scoring rules to change the comparison. Candidate instantiation and candidate scoring act at different stages: the former changes which task objects exist or survive, while the latter changes how already-fixed alternatives compete. Together they provide two concrete routes by which identical intervention states can produce different task-level responses.

4 Execution Can Reverse Intervention Ordering and Selection

4.1 Matched States Reveal an Ordering Reversal

We first test the strongest comparative consequence: can changing only the executor reverse the ordering of two matched intervention states? Sound event detection (SED) is useful for this test because the model produces continuous frame-level posteriors, whereas evaluation is performed on discrete temporal events. The boundary between a continuous state and a task object is therefore explicit. We intervene on the prelogit hidden states of a frozen DCASE-2024 DESED CRNN (Cornell et al., 2024; Turpault et al., 2019). Intervention magnitude and location are fixed, while only the direction varies:

$$d_\theta = \cos\theta\, d_{\mathrm{harm}} + \sin\theta\, d_{\mathrm{null}}, \tag{14}$$

where $\theta \in \{0, 30, 60, 90, 120, 150, 180\}^\circ$. Here, d_{harm} is a unit prelogit direction that decreases the clean decision margin, and d_{null} is orthogonal to this local decision direction. Each posterior state is computed and cached once, and every executor receives exactly the same cached state. We compare fixed thresholding, minimum-duration filtering, gap merging, and hysteresis. Fixed thresholding uses a threshold of 0.5; the remaining executors are registered on clean validation data to match the reference in clean pooled-count Event-F1. The registered parameters are held fixed across the intervention sweep, and each executor is applied to the shared cached states. Full executor parameters and registration details are provided in Appendix A.

The complete trajectory contains a local ordering reversal. For the same cached $120^\circ \to 180^\circ$ transition, Event-F1 decreases by 0.0035 under fixed thresholding but increases by 0.0142 under hysteresis. Because the cached posterior states are identical, this sign change comes from the state-to-event mapping: fixed thresholding prefers 120°, whereas hysteresis prefers 180° for the same pair of states. The corresponding executor–intervention interaction is

$$\mathcal{J} = +0.0177, \tag{15}$$

with a simultaneous 95% confidence interval of $[0.0089,\ 0.0265]$ over all 21 angle pairs. Figure 2a separates absolute performance from the comparison of interest. Hysteresis has higher absolute Event-F1 throughout the sweep, but execution relativity concerns the direction of change within each executor, which is opposite for this transition. Minimum-duration filtering and gap merging likewise remove the local performance drop seen under fixed thresholding. Five-fold source-separated executor registration preserves the pattern, selecting hysteresis $0.3/0.7$ in every fold and yielding a mean held-out interaction of $\bar{\mathcal{J}} = +0.0170$. Complete trajectories, simultaneous contrasts, and per-shift results are reported in Appendix B (Tables 2–4).

The rotation sweep is a controlled representation-space family. We next test whether the same comparative structure appears when intervention settings arise from training. We train three otherwise matched SED checkpoints while varying only the evidence-weighted consistency coefficient $\lambda_{\mathrm{EW}} \in \{1, 2, 4\}$, with architecture, data, optimizer, schedule, and random seed fixed. For the endpoint transition $\lambda_{\mathrm{EW}} = 1 \to 4$, Event-F1 increases by 0.0124 under fixed thresholding but decreases by 0.0050 under hysteresis, giving $\mathcal{J} = -0.0174$ with a 95% confidence interval of $[-0.0276, -0.0067]$. Across the full three-setting family, fixed thresholding peaks at $\lambda_{\mathrm{EW}} = 2$, whereas hysteresis peaks at $\lambda_{\mathrm{EW}} = 1$. Both the controlled intervention trajectory and the separately trained checkpoints show executor-dependent ordering. The complete checkpoint family is reported in Table 5.

4.2 Candidate Instantiation Drives the Fixed-Threshold Performance Drop

We next ask why the fixed-threshold side of the reversal decreases. The change is concentrated at a sparse set of posterior values near the executor's formation boundary. For the $120^\circ \to 180^\circ$ transition,

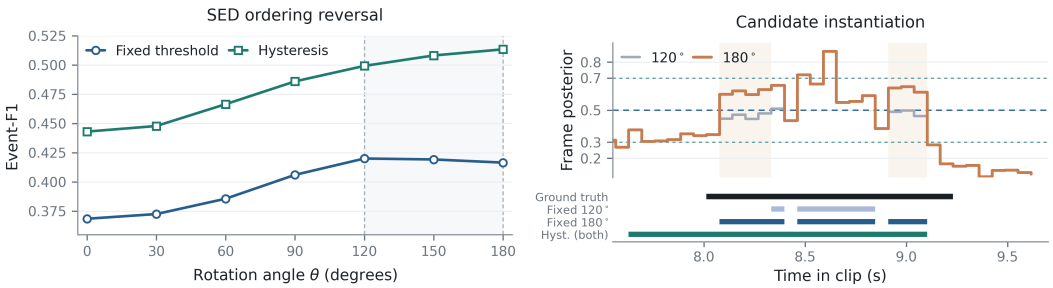


Figure 2: SED ordering reversal and candidate instantiation. (a) The same cached states give opposite $120° \rightarrow 180°$ orderings under fixed thresholding and hysteresis. (b) Sparse boundary crossings alter fixed-threshold event segments while hysteresis preserves the event structure.

only about 0.03%–0.05% of frame–class posterior locations newly cross the fixed threshold of 0.5, yet these crossings can create, connect, or split event segments. Figure 2b shows the mechanism on a held-out example: a small local posterior change alters the event structure produced by fixed thresholding while the corresponding hysteresis output remains unchanged. We refer to this conversion from continuous state changes to changes in discrete task objects as candidate instantiation.

To test whether these boundary crossings are sufficient to produce the fixed-threshold drop, we manipulate only the same crossing locations. Moving them from 0.499 to 0.501 decreases Event-F1 by 0.0039 and increases fragmentation by 0.0213. The intervention is deliberately local: all other posterior entries remain unchanged. Because both values lie between the low and high hysteresis thresholds, the manipulation crosses the fixed-threshold formation boundary without crossing a hysteresis start or continuation boundary. Complementary removal and injection tests give the same task-level effect in opposite directions. Removing the newly crossing locations from the $180°$ state restores 0.0039 Event-F1, while injecting them into the $120°$ state decreases Event-F1 by 0.0039. These results show that the sparse boundary-crossing set is sufficient to remove or reproduce the fixed-threshold performance drop. The mechanism is structural: crossing 0.5 changes event membership and connectivity, which increases fragmentation and changes Event-F1. Thus a tiny fraction of posterior entries can determine the sign of the local fixed-threshold transition because the executor converts those continuous changes into discrete event structure. Full manipulation results are reported in Table 6.

4.3 Execution Changes Intervention Selection

The ordering change also reaches intervention selection. We measure this consequence with five-fold cross-fitting, using sample splitting to separate selection from evaluation (Chernozhukov et al., 2018). For each held-out fold k, executor D selects an intervention setting using only the remaining four folds,

$$\hat{\lambda}_D^{(-k)} = \arg\max_{\lambda \in \Lambda} U_D^{(-k)}(\lambda), \tag{16}$$

with the same deterministic tie-breaking rule across executors. The selected settings are then evaluated on the held-out fold under the target executor. We define the cross-fitted transfer loss as

$$\mathcal{L}_{\mathrm{CF}}(D_1 \rightarrow D_2) = \frac{1}{K} \sum_{k=1}^{K} \left[U_{D_2}^{(k)} \left(\hat{\lambda}_{D_2}^{(-k)} \right) - U_{D_2}^{(k)} \left(\hat{\lambda}_{D_1}^{(-k)} \right) \right]. \tag{17}$$

Fixed thresholding and hysteresis select different intervention settings in all five folds. Transferring the setting selected under fixed thresholding to hysteresis incurs a mean loss of 0.0147 Event-F1 on held-out data, with a 95% confidence interval of $[0.0032,\ 0.0188]$; 99.0% of grouped bootstrap replicates yield a positive transfer loss. Both selected settings are evaluated under the same target executor on the same held-out data. Their utility difference measures the consequence of transferring a selected setting across execution rules. Execution therefore changes both the measured intervention response and the setting carried forward from the family.

5 CANDIDATE SCORING PRODUCES EFFECT-SIZE RELATIVITY

The SED mechanism acts while continuous states are being turned into discrete task objects. Execution dependence can also arise one stage later, after the candidate set is already fixed, through the rule used to compare those candidates. We study this complementary mechanism on frozen Qwen3-8B (Yang et al., 2025) under CAA (Rimsky et al., 2024), where $h' = h + \alpha v$. For each candidate answer a, we cache its token log-probability sum L_a and length T_a at every intervention setting and reuse exactly the same quantities across scoring executors. Only the deterministic scoring rule changes:

$$S_\gamma(a) = \frac{L_a}{T_a^\gamma}. \tag{18}$$

Here, $\gamma = 0$ ranks candidates by total sequence log-probability, while larger γ applies stronger length normalization. The candidate identities, lengths, and cached log-probabilities are fixed across executors; only their conversion into relative scores changes. For a fixed correct–competitor pair (y, b), the intervention-induced change in pairwise margin is

$$\Delta M_\gamma(y, b) = \frac{\Delta L_y}{T_y^\gamma} - \frac{\Delta L_b}{T_b^\gamma}.$$

Two scoring executors therefore apply different candidate-specific weights to exactly the same changes in cached evidence. If $T_y = T_b$, changing γ only rescales the pairwise margin by a positive constant and preserves its ordering. When candidate lengths differ, however, the same ΔL_a values can produce different margin movements under different scorers. Sensitivity is most likely when this reweighting acts on competitors with small initial margins, connecting the scoring rule directly to the stability condition in Eq. 13. This gives a concrete condition under which candidate scoring should matter: the scorer must weight competing candidates differently enough for the intervention-induced margin movement to approach a decision boundary.

At the aggregate level, the scoring rule reshapes the Qwen3-8B intervention response while all candidate statistics remain matched. On ARC (Clark et al., 2018), for the endpoint transition $\alpha : -1.5 \to 1.5$, changing the scoring executor from $\gamma = 0$ to $\gamma = 1.5$ yields an executor–intervention interaction of $+0.0529$ with a 95% confidence interval of $[0.0350, 0.0708]$. Both scoring executors select the same coefficient, $\alpha^* = 0.5$. This illustrates the first level of the hierarchy: execution dependence can be real and statistically detectable without changing the selected intervention setting. The local mechanism is candidate competition. Let $b_\gamma(\alpha)$ denote the highest-scoring incorrect candidate, and define the margin between the correct answer and its strongest competitor as

$$b_\gamma(\alpha) = \arg\max_{b \neq y} S_\gamma(b; \alpha), \qquad m_\gamma(\alpha) = S_\gamma(y; \alpha) - S_\gamma(b_\gamma(\alpha); \alpha). \tag{19}$$

For a local intervention transition $\alpha_i \to \alpha_j$, define

$$\Delta m_\gamma = m_\gamma(\alpha_j) - m_\gamma(\alpha_i), \qquad G = \left| \Delta m_{\gamma_2} - \Delta m_{\gamma_1} \right|. \tag{20}$$

The statistic G quantifies how differently two scoring executors move the correct-versus-competing margin under the same intervention transition. It is small when both scorers transform the intervention evidence similarly and large when the same cached evidence produces different changes in candidate competition. Figure 3 makes the mechanism concrete on a held-out MMLU example. Panel (a) shows the candidate-specific change in total log-probability, ΔL_a, for the transition $\alpha : 0 \to 0.5$, together with candidate token lengths. These quantities are shared by both scoring executors. Panel (b) applies the two scoring rules to the same cached statistics. Under $\gamma = 0.5$, the correct-versus-best-wrong margin moves from negative to positive, so the correct answer becomes preferred. Under $\gamma = 1.5$, the margin remains negative and a competing answer remains preferred. The executor dependence is created entirely at the comparison stage: the intervention-induced candidate evidence is identical, but the scoring rule turns that evidence into different decision margins.

The same margin view also produces an aggregate sensitivity signal. On ARC, G separates questions whose intervention-induced correctness change differs across the two scoring executors with an AUC of 0.8214. We then freeze the transition, layer, executor pair, and statistic identified on ARC and transfer them to MMLU (Hendrycks et al., 2021); only the MMLU-specific CAA direction is re-estimated from validation data. Among 1,000 held-out questions spanning 57 subjects, 31 are executor-sensitive under this fixed comparison. The transferred score achieves an AUC of 0.7395,

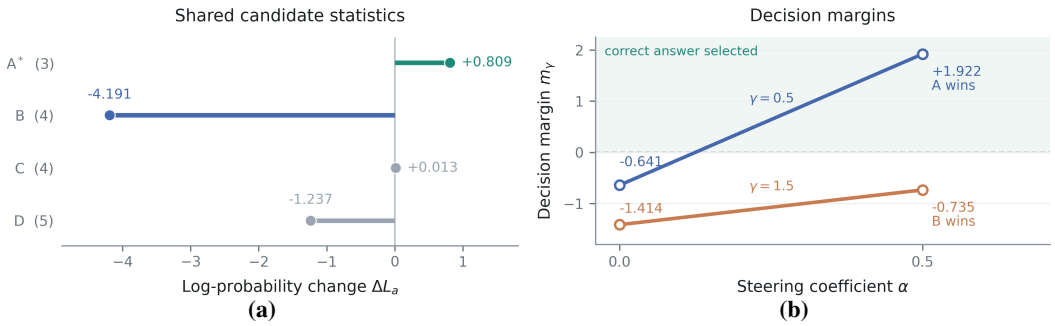


Figure 3: Candidate scoring with shared statistics. (a) Candidate-level changes ΔL_a and token lengths are shared across scoring executors. (b) Different length-normalized scoring rules map those same statistics to different correct-versus-competitor margins.

with a subject-cluster 95% confidence interval of $[0.6869, 0.8216]$. ARC discovery and MMLU transfer results are summarized in Table 7.

Candidate instantiation and candidate scoring act at two distinct stages of the state-to-output mapping. Candidate instantiation changes how continuous states become discrete task objects; candidate scoring changes how fixed alternatives are compared. Both alter $U_D(\lambda)$ without changing upstream intervention states, but they concentrate sensitivity in different regions: near formation boundaries for instantiation and near competitive margins for scoring. Qwen3-8B exhibits effect-size relativity while retaining the same selected setting, whereas SED propagates the dependence to ordering and selection.

6 Mechanism-Derived Predictions Across Output Structures

The two mechanisms suggest where executor sensitivity should appear in other output structures. Candidate instantiation predicts stronger effects when intervention-induced states lie near object-formation boundaries. Candidate scoring predicts stronger effects when competing hypotheses have similar evidence and alternative comparison rules can move their relative margins. We test the first prediction in spatial outputs and examine the comparison stage in sequence decoding, while keeping the upstream intervention states matched throughout.

Spatial outputs test the formation-boundary prediction. On VOC2012, we study DeepLabV3-ResNet50 segmentation and DETR-ResNet50 detection (Everingham et al., 2010; Chen et al., 2017; Carion et al., 2020), changing only the downstream formation or filtering rule applied to shared upstream states. In segmentation, a confidence-gated executor maps pixels below the gate to VOC class 0 (background). As the gate becomes more selective, the executor–intervention interaction increases from 0.0056 at threshold 0.70 to 0.0107 at threshold 0.95; the strongest gate has a 95% confidence interval of $[0.0029, 0.0176]$. In detection, the score threshold controls candidate admission and class-wise NMS controls retention among overlapping boxes. For the $\alpha = 1 \to 2$ transition, moving from low-score, loose-NMS to high-score, strong-NMS execution shifts the intervention response from -0.0017 to $+0.0053$, yielding an interaction of $+0.0070$ with a 95% confidence interval of $[0.0014, 0.0117]$. Across both tasks, more selective formation rules amplify the leverage of intervention-induced states near downstream boundaries. Temporal events, foreground pixel assignments, and retained boxes differ in semantics, yet the same mechanism recurs: small state movements near a formation boundary change which task objects survive execution.

Sequence decoding probes the comparison side of the framework. On LibriSpeech, greedy decoding and language-model-free beam decoding operate on exactly the same cached Wav2Vec2 CTC emissions (Baevski et al., 2020; Panayotov et al., 2015), but resolve sequence hypotheses through different comparison rules. Across $\alpha \in \{-1, 0, 1, 2\}$, the two decoders produce different intervention responses. For the transition $\alpha = -1 \to 2$, the executor–intervention interaction is $+0.0015$ in raw CER units, with a 95% confidence interval of $[0.0011, 0.0019]$, while both retain the same preferred α. Sequence decoding exhibits effect-size relativity without changing selection; the full CER/WER sweep is reported in Table 8. Across these tasks, the same two downstream structures

recur. Formation-based executors become sensitive when a small state movement changes object membership; comparison-based executors become sensitive when different rules transform competitive evidence differently. Figure 2 localizes the first mechanism at event-formation boundaries, and Figure 3 localizes the second at decision margins. Segmentation and detection extend the formation pattern to spatial outputs, while CTC shows that an executor can change the measured intervention response without moving the selected setting.

The cross-task results also give a simple view of stability. An intervention can remain stable when its states stay well inside the same decision region: different executors may change the measured effect size, yet preserve the ordering of settings. Qualitative changes become more likely near boundaries or small competitive margins, where a modest state movement can change object membership or candidate preference. This geometric view connects the empirical hierarchy to the two mechanisms. Formation boundaries determine when continuous state movement becomes a discrete object change; competitive margins determine when the same evidence becomes a different ranking. In both cases, executor sensitivity is largest where the downstream decision is close to changing. For matched-state evaluation, a task-valid executor family should preserve task semantics, the evaluation metric, and the information available at execution time; each rule is chosen independently of the intervention sweep and then held fixed across settings. The family can naturally reflect application choices such as event-duration requirements, score thresholds, or decoding budgets, and its members need not have equal absolute utility. In deployment-oriented studies, the most informative family is the set of plausible output rules for the target pipeline, because this directly tests whether an intervention comparison or selected setting transfers across realistic implementations.

7 Discussion

Our results show that an intervention response is a property of the intervention together with the executor that turns its state into a task output. Under matched upstream states, changing only that final rule can produce different response curves. In SED, the difference reaches the strongest levels of the hierarchy: fixed thresholding and hysteresis reverse a local ordering and select different intervention settings. The boundary manipulations explain the fixed-threshold side of this reversal by showing that sparse crossings of an event-formation boundary are sufficient to reproduce the performance drop. In Qwen3-8B, the candidate set and candidate statistics remain fixed, yet alternative scoring rules change competitive margins and the measured intervention response while retaining the same selected coefficient. The spatial and CTC experiments show the same two downstream structures in other output types: sensitivity near formation boundaries and sensitivity in the comparison among competing hypotheses. More broadly, executor sensitivity is structured rather than arbitrary, concentrating near formation boundaries and small competitive margins. These quantities may therefore serve as practical diagnostics for identifying intervention comparisons that warrant evaluation under multiple executors.

Overall, an intervention setting should be selected together with the executor that will convert its states into task outputs, rather than treated as an executor-independent choice. When the downstream rule is fixed by the target pipeline, intervention tuning should use that rule. When several task-valid executors are plausible, the relevant question is whether the ranking and selected setting remain stable across them. Matched-state evaluation provides a direct way to answer this question because it reuses the same intervention states and changes only the final execution rule. This shows whether an intervention choice transfers across execution rules or is specific to one evaluation or deployment pipeline.

AI Use Statement

Generative AI tools were used for literature search, code debugging, translation, and language editing. The authors conducted all experiments and analyses, verified the reported results and conclusions, and take full responsibility for the submitted content.

REFERENCES

O. Antverg and Y. Belinkov. On the pitfalls of analyzing individual neurons in language models. *ICLR*, 2022.

S. Athey and G. Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS*, 2020.

Y. Bao, X. Zhang, J. Chen, G. Su, Y. Cai, H. Peng, B. Sun, H. Weng, Y. Liu, and J. Yin. Faithful Bi-Directional Model Steering via Distribution Matching and Distributed Interchange Interventions. *ICLR*, 2026.

C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic. A framework for the robust evaluation of sound event detection. *ICASSP*, 2020.

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS—Improving object detection with one line of code. *ICCV*, 2017.

L. Cances, P. Guyot, and T. Pellegrini. Evaluation of post-processing algorithms for polyphonic sound event detection. *WASPAA*, 2019.

N. Carion et al. End-to-end object detection with transformers. *ECCV*, 2020.

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587, 2017.

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457, 2018.

S. Cornell, J. Ebbers, C. Douwes, I. Martín-Morató, M. Harju, A. Mesaros, and R. Serizel. DCASE 2024 Task 4: Sound event detection with heterogeneous data and missing labels. *DCASE 2024 Challenge Technical Report*, 2024.

J. Ebbers, R. Haeb-Umbach, and R. Serizel. Post-processing independent evaluation of sound event detection systems. *DCASE Workshop*, 2023.

M. Everingham et al. The Pascal Visual Object Classes (VOC) challenge. *IJCV*, 2010.

Z. Feng, T. Li, Z. Zhu, H. Zhou, J. Qian, L. Zhang, C. Deryl, L. Mak, G. Ng, and K. Mao. Fine-Grained Activation Steering: Steering Less, Achieving More. *ICLR*, 2026.

Z. Gao, S. Peng, S. Wakamiya, and E. Aramaki. What Does Activation Steering Control? Attribution Across Answer Encodings and Output-Sensitive Subspaces. arXiv:2608.22985, 2026.

A. Geiger, D. Ibeling, A. Zur, M. Chaudhary, S. Chauhan, J. Huang, A. Arora, Z. Wu, N. Goodman, C. Potts, and T. Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *ICML*, 2006.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *ICLR*, 2021.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *ICLR*, 2020.

K. Imai and M. Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *Annals of Applied Statistics*, 7(1):443–470, 2013.

M. Kang and J. Kim. Enhancing instruction following of LLMs via activation steering with dynamic rejection. *ICLR*, 2026.

B. W. Lee, I. Padhi, K. N. Ramamurthy, E. Miehling, P. Dognin, M. Nagireddy, and A. Dhurandhar. Programming refusal with conditional activation steering. *ICLR*, 2025.

K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-Time Intervention: Eliciting truthful answers from a language model. *NeurIPS*, 2023.

X. L. Li, A. Holtzman, D. Fried, P. Liang, J. Eisner, T. Hashimoto, L. Zettlemoyer, and M. Lewis. Contrastive decoding: Open-ended text generation as optimization. *ACL*, 2023.

A. Makelov, G. Lange, A. Geiger, and N. Nanda. Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching. *ICLR*, 2024.

C. Meister, R. Cotterell, and T. Vieira. If beam search is the answer, what was the question? *EMNLP*, 2020.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. *NeurIPS*, 2022.

D. V. Nguyen, N. Y. Pham, H. M. Vu, L. Zhang, and T. M. Nguyen. Activation steering with a feedback controller. *ICLR*, 2026.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. *ICASSP*, 2015.

L. Rauch, R. Heinrich, H. Ghaffari, L. Miklautz, I. Moummad, B. Sick, and C. Scholz. Unmute the Patch Tokens: Rethinking probing in multi-label audio classification. *ICLR*, 2026.

N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner. Steering Llama 2 via contrastive activation addition. *ACL*, 2024.

P. Rodriguez, A. Blaas, M. Klein, L. Zappella, N. Apostoloff, M. Cuturi, and X. Suau. Controlling language and diffusion models by transporting activations. *ICLR*, 2025.

J. Su, J. Chen, H. Li, Y. Chen, L. Qing, and Z. Zhang. Activation Steering Decoding: Mitigating Hallucination in Large Vision-Language Models through Bidirectional Hidden State Intervention. *ACL*, 2025.

A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation Addition: Steering language models without optimization. arXiv:2308.10248, 2023.

N. Turpault, R. Serizel, A. P. Shah, and J. Salamon. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. *DCASE Workshop*, pp. 253–257, 2019.

T. J. VanderWeele and M. J. Knol. A tutorial on interaction. *Epidemiologic Methods*, 3(1):33–72, 2014.

T. Wang, X. Jiao, Y. Zhu, Z. Chen, Y. He, X. Chu, J. Gao, Y. Wang, and L. Ma. Adaptive Activation Steering: A tuning-free LLM truthfulness improvement method for diverse hallucinations categories. *The ACM Web Conference*, pp. 2562–2578, 2025.

D. Xie, S. Shao, L. Bai, Z. Zhou, B. Cheng, S. Yang, J. Wu, and Z. Xie. Guidance matters: Rethinking the evaluation pitfall for text-to-image generation. *ICLR*, 2026.

Z. Xu, C. Wu, H. Sun, H. Hong, M. Wang, Y. Yao, L. Huang, H. Xue, S. Deng, Z. Chu, H. Chen, and N. Zhang. Why Steering Works: Toward a Unified View of Language Model Parameter Dynamics. *ACL*, pp. 31719–31736, 2026.

A. Yang et al. Qwen3 technical report. arXiv:2505.09388, 2025.

F. Zhang and N. Nanda. Towards best practices of activation patching in language models: Metrics and methods. *ICLR*, 2024.

A Experimental Details and Statistical Procedures

This appendix specifies the experimental choices underlying the main-text comparisons. Throughout, all conditions derived from the same original sample remain grouped during resampling, preserving the pairing across intervention settings and executors.

Table 1: Essential experimental configuration. Grouped resampling keeps all conditions derived from one original sample together.

Setting	Model / state	Intervention parameter and execution	Evaluation
SED rotation	DCASE-2024 DESED CRNN; prelogit hidden state	clean-decision, label-free d_{harm}; orthogonal d_{null}; $d_\theta = \cos\theta\, d_{\mathrm{harm}} + \sin\theta\, d_{\mathrm{null}}$; per-edited-frame $\|\delta h\|_2 = 1$; fixed 0.5 vs. hyst. $0.3/0.7$	940 validation files; three MicIRP shifts; collar Event-F1; filename groups
Qwen CAA	Frozen Qwen3-8B; final-token hidden state at layer 18	$h_l \leftarrow h_l + \alpha v_l$ broadcast over all token positions; ARC-developed $\alpha : 0 \rightarrow 0.5$; $\gamma = 0.5$ vs. 1.5	ARC development on 256 examples; protocol transfer to 1,000 held-out MMLU questions across 57 subjects
VOC seg.	DeepLabV3-ResNet50 backbone output; VOC2012 val (300)	$h_\alpha = h_{\mathrm{corrupt}} + \alpha(h_{\mathrm{clean}} - h_{\mathrm{corrupt}})$; 11 α settings; argmax vs. gates 0.70–0.95; sub-threshold pixels $\rightarrow$ background class 0	21-class mIoU; sub-threshold predictions scored as background; five severity-3 corruptions; 500 image-grouped bootstraps
VOC det.	DETR-ResNet50 final decoder state; VOC2012 val (200)	foreground-vs.-no-object direction; $\alpha = \{0, 0.5, 1, 1.5, 2\}$; score $0.20/0.50 \times$ class-wise NMS IoU $0.70/0.30$	exact mAP50; 500 image-grouped bootstraps
CTC	Wav2Vec2-base-960h emissions; LibriSpeech test-clean (1,000)	nonblank-vs.-blank direction; $\alpha = \{-1, 0, 1, 2\}$; greedy vs. LM-free beam (width 16, token top-k 12)	CER/WER; three corruptions; utterance groups
EW family	Frozen SED checkpoints	separately trained $\lambda_{\mathrm{EW}} = \{1, 2, 4\}$ variants; registered SED executors	exact Event-F1; held-out MicIRP; no current-sample clean counterpart

SED rotation and executor registration. The rotation is applied at the `prelogit_hidden` state of the frozen CRNN SED head. Let $p^{\mathrm{clean}}_{t,c}$ denote the posterior for class c at frame t on the paired clean view, and set $r_{t,c} = +1$ when $p^{\mathrm{clean}}_{t,c} \geq 0.5$ and -1 otherwise. The harmful direction d_{harm} is the local prelogit direction induced by the frozen decision head, oriented so that a positive step reduces the signed clean-side decision margin specified by $r_{t,c}$, then L2-normalized. A seed-controlled local null vector is orthogonalized to d_{harm} and L2-normalized to obtain d_{null} (main rotation seed 2026). We form

$$d_\theta = \cos\theta\, d_{\mathrm{harm}} + \sin\theta\, d_{\mathrm{null}}, \qquad \|d_\theta\|_2 = 1. \tag{21}$$

The 0° direction follows the harmful axis, 90° is locally decision-null, and 180° reverses the harmful axis. The construction uses the model's clean decision, with ground-truth events entering only at final evaluation. The same clean-decision procedure defines a fixed candidate-frame mask with `select_rate`$= 0.05$. This mask is chosen before the angular, corruption, and executor comparisons and reused for every θ, MicIRP shift, and executor; the observed edited fraction is approximately 0.0469. For each selected frame, we apply one unit-norm hidden perturbation,

$$h_t^{(\theta)} = h_t^{\mathrm{shift}} + d_\theta, \tag{22}$$

while unselected frames remain unchanged.

The rotation sweep uses $\theta \in \{0, 30, 60, 90, 120, 150, 180\}^\circ$. Fixed thresholding at 0.5 serves as the reference executor. The clean-only registration audit includes this reference and 49 candidate

settings spanning the auxiliary executor families. Within each family, one representative is selected on clean validation data by minimizing its pooled-count Event-F1 difference from the fixed-threshold reference. The resulting representatives are minimum duration (threshold 0.45, duration 0.1 s), merge gap (threshold 0.6, gap 0.3 s), and hysteresis (low/high $0.3/0.7$). A five-fold source-separated audit repeats registration using the four non-held-out folds; Appendix B.2 reports the results. Event matching uses a 0.2 s onset collar and an offset collar of $\max(0.2\,\text{s}, 0.2 \times \text{GT duration})$. For the primary fixed-versus-hysteresis analysis, exact `sed_scores` Event-F1 is recomputed inside every original-file grouped bootstrap replicate for all 21 angle pairs. Candidate-instantiation contrasts use the same source-level grouping with 500 grouped bootstrap replicates.

MicIRP distribution shifts. The three evaluation shifts are derived from real microphone impulse responses rather than synthetic equalization. We select three impulse responses from the MicIRP library using unlabeled frequency-response statistics only: `IR_FosterDynamicDF1.wav` (Colored), which has the largest frequency-response variation with a standard deviation of approximately 7.26 dB; `IR_Shure510C.wav` (HF-loss), which has the strongest high-frequency attenuation with a high-versus-mid difference of approximately -18.94 dB; and `Reslo_VMC2.wav` (Notch), which has the deepest narrow-band attenuation with a notch depth of approximately 41.28 dB. IR selection does not use SED labels or SED performance.

Each IR is resampled to 16 kHz, cropped from 5 ms before to 100 ms after its main impulse peak, mean-centered, and energy-normalized. Audio is transformed by FFT convolution. The IR peak is used to compensate convolutional delay, the output is cropped to the original waveform length, and its RMS is matched to the input RMS.

SED candidate-instantiation manipulations. For the $120° \to 180°$ transition, define the crossing mask

$$M = \big\{(t, c) : p_{120}(t, c) < 0.5,\ p_{180}(t, c) \geq 0.5\big\}. \tag{23}$$

The mask contains only 0.03–0.05% of frame–class entries. Same-location clamp, crossing-removal, and crossing-injection interventions modify only cached posterior entries in this mask; all remaining entries stay unchanged. The clamp sets selected values to 0.499 or 0.501. Appendix B.4 reports Event-F1 and fragmentation.

Dataset-level fragmentation ratio. We use a project-specific dataset-level fragmentation ratio to quantify whether one ground-truth event is split into multiple predicted segments. For class c, let G_c be the set of ground-truth events, and let $N_{\text{overlap}}(e)$ be the number of predicted events from the same file and class whose temporal IoU with ground-truth event e is greater than zero. We define

$$\text{Frag} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{|G_c|} \sum_{e \in G_c} \max(0,\ N_{\text{overlap}}(e) - 1)\,. \tag{24}$$

A ground-truth event covered by one predicted event contributes zero, whereas coverage by $k > 1$ predicted fragments contributes $k - 1$. The quantity is first averaged over ground-truth events within each class and then macro-averaged over the 10 event classes. This quantity is a project-specific structural diagnostic; it is not the per-file excess-count proxy or an official `sed_scores` fragmentation metric.

Training-derived SED checkpoint family. The EW family contains three separately trained frozen checkpoints indexed by $\lambda_{\text{EW}} \in \{1, 2, 4\}$. All variants share the same 10,000-clip synthetic strongly labeled training split, architecture, optimizer, schedule, and random seed (42); only λ_{EW} changes. Training runs for 10 epochs with Adam, learning rate 10^{-4}, batch size 16, and gradient clipping at 5.0. Every λ_{EW} variant uses the same checkpoint-selection rule: at each epoch, events are formed on clean validation data using the fixed-threshold executor with posterior threshold 0.5, and the checkpoint with the highest official `sed_scores` collar Event-F1 at threshold 0.5 is retained. Colored, HF-loss, and Notch results are never used for checkpoint selection. The paired EQ training view uses a smooth mel-frequency response perturbation of strength 0.20, eight random anchors, and gain clipping to $[0.5, 2.0]$. The objective is

$$\begin{aligned} \mathcal{L}_{\text{train}} = {} & \text{BCE}(p_{\text{raw}}, y) + \text{BCE}(p_{\text{shift}}, y) \\ & + 0.1\ \text{mean}\left[(1 + \lambda_{\text{EW}}\,\text{sg}(p_{\text{raw}}))(p_{\text{shift}} - \text{sg}(p_{\text{raw}}))^2\right], \end{aligned} \tag{25}$$

where sg denotes stop-gradient. All checkpoints are frozen before shift evaluation and are evaluated on shifted inputs without a current-sample clean counterpart.

Cross-task interventions and selection protocol. For segmentation, we use DeepLabV3-ResNet50 (Chen et al., 2017); for detection, we use DETR-ResNet50 (Carion et al., 2020), both on VOC2012 (Everingham et al., 2010). The 11 segmentation settings are

$$\alpha \in \{0, 0.25, 0.5, 0.75, 1, 1.125, 1.25, 1.375, 1.5, 1.75, 2\},$$

and the five corruptions are Gaussian noise, blur, brightness, contrast, and JPEG compression at severity 3. Segmentation uses the first 300 examples and detection the first 200 examples in the fixed VOC2012 validation ordering. These deterministic subsets are shared across every corruption, intervention setting, and executor.

For segmentation, let $q_r(c)$ be the predicted probability of VOC class $c \in \{0, \dots, 20\}$ at pixel r. The ungated executor predicts $\arg\max_c q_r(c)$. A gated executor with threshold τ instead uses

$$\hat{y}_r^{(\tau)} = \begin{cases} \arg\max_c q_r(c), & \max_c q_r(c) \geq \tau, \\ 0, & \max_c q_r(c) < \tau, \end{cases}$$

where class 0 is VOC background. Sub-threshold pixels are relabeled as background and remain in evaluation. We accumulate the standard 21×21 confusion matrix C, with ground-truth classes indexing rows and predicted classes indexing columns, and compute

$$\mathrm{IoU}_c = \frac{C_{cc}}{\sum_j C_{cj} + \sum_i C_{ic} - C_{cc}}, \qquad \mathrm{mIoU} = \frac{1}{21} \sum_{c=0}^{20} \mathrm{IoU}_c \,.$$

A gated pixel with ground-truth background contributes to C_{00} as a background true positive. A gated pixel with ground-truth foreground class $c > 0$ contributes to C_{c0}, acting simultaneously as a false negative for class c and a false positive for background. Every gated pixel therefore remains part of the 21-class mIoU computation. Only pixels with ground-truth label 255 are ignored according to the standard VOC evaluation convention.

For detection, we use the implemented inverse-squared-norm scaling

$$u = \frac{\bar{w}_{\mathrm{fg}} - w_{\mathrm{noobj}}}{\|\bar{w}_{\mathrm{fg}} - w_{\mathrm{noobj}}\|_2^2 + 10^{-12}}, \tag{26}$$

where $\bar{w}_{\mathrm{fg}}$ is the mean foreground-class classifier weight and w_{noobj} is the no-object classifier weight. Original DETR boxes are reused across α settings and only class logits change; the downstream score/NMS rules are then applied to these shared detection states.

The sequence experiment uses Wav2Vec2-base-960h (Baevski et al., 2020) on 1,000 LibriSpeech test-clean utterances (Panayotov et al., 2015) at 16 kHz. Corruptions are additive noise with standard deviation 0.08 times waveform RMS, gain multiplier 0.55, and clipping $\mathrm{clip}(1.8x, -0.55, 0.55)$. The intervention direction is the L2-normalized difference between the mean nonblank classifier weight and the blank classifier weight. Greedy decoding and language-model-free prefix-beam decoding operate on exactly the same cached emissions; the beam uses width 16 and token top-k 12.

Original samples are assigned to five fixed folds for cross-fitting (Chernozhukov et al., 2018); every corruption, intervention setting, and executor derived from a source sample remains in the same fold. For held-out fold k, each executor selects its intervention setting using the remaining four folds, and the selected settings are evaluated under the target executor on fold k. Positive $\mathcal{L}_{\mathrm{CF}}$ denotes target-executor utility lost by transferring the setting selected under the source executor.

Grouped bootstrap inference. For the seven-angle SED trajectory, we use 500 original-filename grouped bootstrap replicates. All three MicIRP shifts, seven intervention angles, and both executors derived from the same original file remain bound within each replicate, and exact `sed_scores` Event-F1 is recomputed. Simultaneous confidence intervals over all $\binom{7}{2} = 21$ angle comparisons use a studentized max-T procedure.

For the EW endpoint interaction, we use 500 original-filename grouped bootstrap replicates. The three MicIRP shifts, both $\lambda_{\mathrm{EW}} \in \{1, 4\}$ checkpoints, and both executors derived from one source file

remain grouped. Point estimates and confidence intervals are computed from exact `sed_scores` Event-F1.

For the cross-fitted transfer loss, the five folds define only the training/evaluation split and are not the bootstrap unit. Confidence intervals use 1,000 original-filename grouped bootstrap replicates. Within every replicate, the four-fold intervention selection is rerun and the selected settings are evaluated and aggregated over the held-out folds.

B SED Robustness and Mechanism Audits

B.1 Complete Trajectory and Simultaneous Inference

Table 2: Shift-average exact Event-F1 over the complete seven-angle SED trajectory for the headline executor pair.

Executor	0°	30°	60°	90°	120°	150°	180°
Fixed threshold	0.3686	0.3726	0.3857	0.4061	0.4201	0.4193	0.4166
Hysteresis	0.4431	0.4479	0.4666	0.4861	0.4993	0.5083	0.5135

The seven-angle sweep contains $\binom{7}{2} = 21$ pairwise transitions. We use 500 original-filename grouped bootstrap replicates, preserving all three shifts, seven angles, and both executors associated with each source file. Exact `sed_scores` Event-F1 is recomputed within every replicate. Studentized max-T intervals provide simultaneous 95% coverage over all 21 angle comparisons. Several interactions involving 180° remain positive after simultaneous correction, showing that executor dependence extends beyond a single angle pair.

Table 3: Trajectory-wide simultaneous inference for selected fixed-versus-hysteresis transitions. All intervals are simultaneous over the 21 pairwise angle transitions.

Transition	Δ_{hyst}	Δ_{fix}	$\mathcal{J}$ [95% simultaneous CI]
$0^\circ \to 180^\circ$	0.0704	0.0480	0.0224 [0.0081, 0.0366]
$30^\circ \to 180^\circ$	0.0656	0.0440	0.0216 [0.0073, 0.0359]
$60^\circ \to 180^\circ$	0.0469	0.0309	0.0160 [0.0015, 0.0306]
$90^\circ \to 180^\circ$	0.0274	0.0105	0.0170 [0.0051, 0.0288]
$120^\circ \to 180^\circ$	0.0142	-0.0035	0.0177 [0.0089, 0.0265]

B.2 Robustness Across Registration and Distribution Shifts

Five-fold source-separated registration reselects hysteresis $0.3/0.7$ in all folds. The held-out mean interaction remains $+0.0170$, with a source-file grouped-bootstrap 95% confidence interval of $[0.0099, 0.0229]$ while preserving fold assignment. The same opposite-sign ordering pattern appears under each of the three evaluated shifts (Table 4).

Table 4: $180^\circ - 120^\circ$ exact Event-F1 response by distribution shift.

Shift	Fixed threshold	Hysteresis
Colored	-0.0027	$+0.0137$
HF-loss	-0.0050	$+0.0122$
Notch	-0.0027	$+0.0167$

The executor-dependent ordering persists under source-separated re-registration and across all three evaluated shifts.

Table 5: Shift-average exact SED Event-F1 for the frozen EW checkpoint family under the two headline executors.

Executor	$\lambda_{\mathrm{EW}} = 1$	2	4
Fixed threshold	0.4044	0.4181	0.4168
Hysteresis	0.4901	0.4893	0.4851

B.3 Training-Derived Checkpoint Family

For $\lambda_{\mathrm{EW}} = 1 \to 4$, the paired fixed-versus-hysteresis interaction is -0.0174, with a 95% confidence interval of $[-0.0276, -0.0067]$. The interval uses 500 original-filename grouped bootstrap replicates, with all three shifts, both endpoint checkpoints, and both executors from a source file kept together and exact `sed_scores` Event-F1 recomputed in every replicate. The endpoint comparison favors $\lambda_{\mathrm{EW}} = 4$ under fixed thresholding and $\lambda_{\mathrm{EW}} = 1$ under hysteresis, while the full three-setting family peaks at $\lambda_{\mathrm{EW}} = 2$ and 1, respectively. This separately trained checkpoint family therefore reproduces the executor-dependent ordering seen along the controlled representation-space trajectory.

B.4 Candidate-Instantiation Audit

The clamp, crossing-removal, and crossing-injection interventions test the contribution of the sparse boundary crossings to the change in discrete event structure. All manipulations act only on cached posterior entries in the crossing mask defined in Appendix A; all remaining entries are unchanged.

Table 6: Exact `sed_scores` grouped-bootstrap results for the fixed-threshold candidate manipulations. Fragmentation is recomputed on the same bootstrap samples.

Comparison	Δ Event-F1 [95% CI]	Δ Frag. [95% CI]
$0.501 - 0.499$	-0.0039 $[-0.0062, -0.0016]$	$+0.0213$ $[0.0118, 0.0342]$
Remove @ 180°	$+0.0039$ $[0.0016, 0.0062]$	-0.0212 $[-0.0341, -0.0117]$
Inject @ 120°	-0.0039 $[-0.0062, -0.0016]$	$+0.0213$ $[0.0118, 0.0342]$

The three manipulations recover the same structural pattern. Crossing the boundary decreases Event-F1 and increases fragmentation; removing the new crossings reverses both changes; injecting the same boundary status reproduces them. The task-level change is concentrated at the sparse boundary-crossing locations.

C Candidate-Scoring Diagnostic: ARC Development and MMLU Protocol Transfer

This section specifies the construction and transfer protocol underlying the Qwen3-8B analysis in Section 5. The scoring rule is defined in Eq. 18, and the executor-disagreement statistic G is defined in Eq. 20. Figure 3 visualizes the candidate-level mechanism. ARC is used to develop and freeze the diagnostic protocol; MMLU is then used as a held-out protocol-transfer test.

CAA direction construction. We use frozen Qwen3-8B (Yang et al., 2025) with CAA-style interventions (Rimsky et al., 2024). For each question and candidate answer, the model receives the concatenated prompt–candidate sequence, and we extract the final-token hidden state at the target layer. For sample i, the steering direction is

$$v_l^{(i)} = h_{l,\mathrm{correct}}^{(i)} - \frac{1}{|\mathcal{B}_i|} \sum_{b \in \mathcal{B}_i} h_{l,b}^{(i)}, \tag{27}$$

where $\mathcal{B}_i$ is the set of incorrect candidate answers. The layer-specific CAA direction v_l is obtained by averaging $v_l^{(i)}$ over 256 development examples. At inference time, the intervention

$$h_l \leftarrow h_l + \alpha v_l \tag{28}$$

is broadcast to all token positions at layer l during the candidate-sequence forward pass.

For every candidate answer a and intervention coefficient α, its token log-probability sum L_a and token length T_a are cached once. All scoring executors operate on these same cached candidate statistics. The dynamic correct-versus-competitor margin is

$$m_\gamma(\alpha) = S_\gamma(y;\alpha) - \max_{b \neq y} S_\gamma(b;\alpha), \tag{29}$$

so the competing incorrect answer is allowed to change with both α and γ.

ARC development and matched scoring. The headline analysis uses layer 18, selected during ARC development from the candidate set $\{12, 18, 24, 30\}$; this is a development choice rather than a preregistered layer. For the local diagnostic development, ARC considers $\alpha \in \{0, 0.1, \ldots, 1.0\}$ and scoring exponents $\gamma \in \{0, 0.5, 1, 1.5\}$. This analysis fixes layer 18, the local transition $\alpha : 0 \to 0.5$, the scorer pair $\gamma = 0.5$ versus $\gamma = 1.5$, and the geometry statistic G before MMLU evaluation. Separately, the aggregate ARC response reported in the main text includes the endpoint comparison $\alpha : -1.5 \to 1.5$: changing the scoring executor from $\gamma = 0$ to $\gamma = 1.5$ yields $\mathcal{J} = +0.0529$, with a 95% confidence interval of $[0.0350, 0.0708]$, while both scoring executors retain the same preferred coefficient, $\alpha^* = 0.5$.

For the frozen local diagnostic, define correctness as

$$c(\alpha, \gamma) = \mathbf{1}\left[\arg\max_a S_\gamma(a;\alpha) = y\right], \tag{30}$$

and the intervention-induced correctness change as

$$d_\gamma = c(\alpha_{\text{high}}, \gamma) - c(\alpha_{\text{low}}, \gamma). \tag{31}$$

A question is executor-sensitive when $d_{\gamma_{\text{left}}} \neq d_{\gamma_{\text{right}}}$. Under the ARC-developed configuration, G separates executor-sensitive ARC questions with AUC 0.8214.

Held-out MMLU protocol transfer. MMLU uses the same frozen layer, transition, scorer pair, and geometry statistic selected on ARC. None of these choices is reselected on the MMLU test set. A new MMLU-specific CAA direction is estimated from 256 MMLU validation examples using the same direction-construction procedure, and evaluation then uses 1,000 held-out MMLU test questions spanning 57 subjects with $\alpha \in \{0, 0.5\}$. Thus, the experiment transfers the diagnostic protocol rather than the ARC steering direction itself. Among the 1,000 held-out questions, 31 are executor-sensitive under this fixed comparison.

Table 7: ARC development and held-out MMLU protocol transfer for the Qwen3-8B candidate-competition diagnostic. The MMLU interval uses subject-cluster bootstrap.

Dataset	Role	α transition	Executors	AUC	95% CI
ARC	development	$0 \to 0.5$	$\gamma = 0.5$ vs. 1.5	0.8214	–
MMLU	held-out transfer	$0 \to 0.5$	$\gamma = 0.5$ vs. 1.5	0.7395	[0.6869,0.8216]

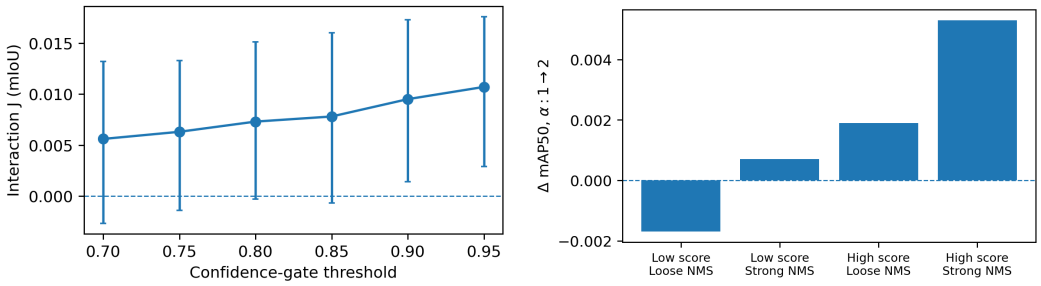
With the protocol frozen from ARC and only the MMLU-specific direction re-estimated from validation data, the transferred diagnostic reaches AUC 0.7395, with a subject-cluster 95% confidence interval of $[0.6869, 0.8216]$.

D Mechanism-Derived Cross-Task Checks

D.1 Spatial Candidate Formation

For segmentation, the confidence gate is an explicit foreground-formation boundary: pixels below the gate are reassigned to background class 0 and remain in the 21×21 confusion matrix used for mIoU. Executor sensitivity increases monotonically with gate strictness, from an interaction of 0.0056 at threshold 0.70 to 0.0107 at 0.95; the strongest gate has a 95% confidence interval of $[0.0029, 0.0176]$. Stricter gates give greater leverage to intervention-induced probability changes

near the foreground/background boundary. For detection, on the $\alpha = 1 \to 2$ transition, the response moves from -0.0017 under low-score/loose-NMS execution to $+0.0053$ under high-score/strong-NMS execution, yielding a loose-versus-strict interaction of $+0.0070$ with a 95% confidence interval of $[0.0014, 0.0117]$. Higher score thresholds remove marginal candidates, while stronger NMS suppresses competing boxes on the same shared box/logit state.



(a) Segmentation gate strictness

(b) Detection score/NMS strictness

Figure 4: Candidate-formation checks in spatial outputs. Segmentation interactions increase with gate strictness, while stricter score/NMS execution shifts the detection response on the same shared upstream state.

D.2 CTC Sequence Decoding

Table 8: CER and WER on 1,000 LibriSpeech test-clean utterances (Panayotov et al., 2015), aggregated over three corruptions.

	$\alpha = -1$	0	1	2
Greedy CER	0.0256	0.0133	0.0188	0.0522
Beam CER	0.0243	0.0132	0.0191	0.0524
Greedy WER	0.0909	0.0397	0.0620	0.1651
Beam WER	0.0835	0.0391	0.0650	0.1686

For $\alpha = -1 \to 2$, the interaction in raw CER units is $+0.0015$, with a 95% confidence interval of $[0.0011, 0.0019]$ (lower is better). Greedy and beam decoding retain the same preferred α. Sequence decoding changes effect size while preserving the selected setting.