TOPORERANK: BOUNDED STRUCTURAL RERANKING FOR FIXED-CANDIDATE GUI ACTION SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graphical user interface (GUI) agents may choose the wrong control even when the target is present, as semantically similar candidates receive close scores. End-to-end accuracy conflates candidate coverage, ranking, target assignment, and final selection. To isolate this decision, we propose TopoRerank under a protocol that fixes the candidate set, Stage–1 scores, target matcher, split, and evaluator. The method adds a bounded, instruction-independent residual from candidate-local accessibility-tree features, changing only close competitors. On package-disjoint AndroidControl, it improves a text-retrieval baseline by 3.62 percentage points across three seeds, with a positive package-equal interval; the gain falls to 0.53 percentage points under a stronger learned text-only ranker. An area-free nearest-center control is positive under the primary retrieval baseline but inconclusive under lexical scoring; app-disjoint GUIrilla results are positive after target-domain refitting. Overall, local hierarchy offers a conditional correction to frozen rankings, not a replacement for semantic ranking or candidate recall.

1 INTRODUCTION

Graphical user interface (GUI) agents turn natural-language requests into actions on web, mobile, and desktop applications. Before acting, an agent must identify the control that realizes a request. Repeated labels, nested elements, and similar results make this difficult even when the target is present: a semantic ranker may place a plausible alternative first. This final choice is distinct from proposing more controls or improving the broader interaction policy.

GUI grounding uses complementary representations. Screenshot-based methods align instructions with visual regions or coordinates (Cheng et al., 2024; Gou et al., 2025); structured web and mobile pipelines expose Document Object Model (DOM) or accessibility elements as candidate controls (Deng et al., 2023; Li et al., 2024b). Relation-aware methods further model links among interface elements (Ma et al., 2026a). These advances enrich target evidence, but end-to-end scores combine candidate generation, target matching, ranking, and action execution. They do not reveal whether hierarchy improves selection when the candidate set and upstream scores are unchanged.

Two difficulties remain when measuring this final decision. A reranker cannot recover a missing target; it can only change the order of available candidates, so candidate coverage must be held fixed. Its value may also depend on the upstream ranker: local structure can resolve near-tied candidates but has less room to act once a stronger ranker separates them. A single operating point cannot reveal this dependence.

Target assignment makes the distinction especially important on AndroidControl. The released manifest provides target points rather than target node identifiers, and the evaluator assigns each point to the smallest candidate box containing it. Geometry therefore affects both the reranker input and which candidate is labeled correct. A change in target assignment can therefore change the measured gain even when the candidate scores stay fixed.

To address these challenges, we propose TopoRerank, a structural reranker for fixed-candidate GUI selection. Specifically, it adds a bounded, instruction-independent residual from candidate-local accessibility-tree position and geometry to frozen Stage–1 scores, so only close competitors can change order (Figure 1).

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.

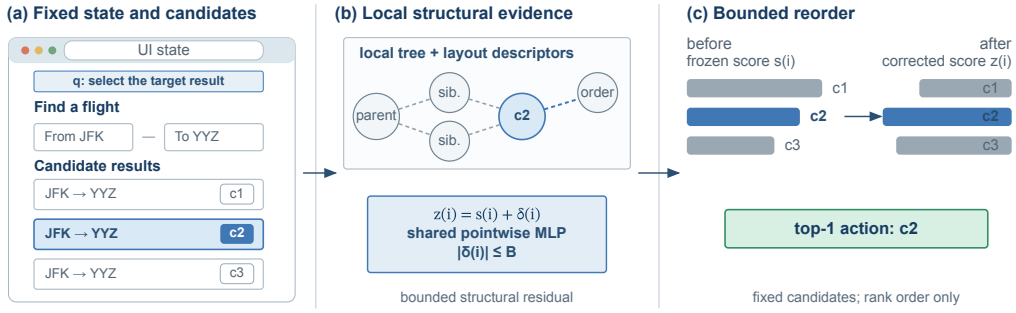


Figure 1: Fixed-candidate protocol. Frozen Stage–1 scores order the unchanged candidate set; local hierarchy and geometry add a bounded residual to close pairs. The target matcher and evaluator remain fixed, and panel (c) shows only the resulting rank change.

Each comparison holds the candidate set, Stage–1 scores, target matcher, split, and evaluator fixed; a separate nearest-center condition tests sensitivity to target assignment. This protocol isolates final selection without changing candidate coverage.

On package-disjoint AndroidControl, we test three frozen upstream settings: (i) lexical token-overlap scoring, (ii) query-local BM25, a retrieval baseline that combines inverse-document-frequency weighting, term-frequency saturation, and candidate-length normalization (Robertson & Zaragoza, 2009), and (iii) an in-domain learned text-only ranker. The residual adds 3.62 percentage points over BM25 across three seeds and 0.53 points over the learned ranker. An area-free matcher control tests sensitivity to target assignment, while an app-disjoint GUIrilla refit examines the correction under a macOS accessibility-tree schema. The results characterize local hierarchy as a measurable late-stage signal whose value depends on the ranking and matcher it acts on.

2 Related Work

GUI grounding and action selection. GUI grounding ranges from language-conditioned coordinate predictors to methods emphasizing layout, hierarchy, region selection, or screen parsing (Cheng et al., 2024; Gou et al., 2025; Wu et al., 2025; Tang et al., 2025; Yang et al., 2025; Park et al., 2025; Hsieh et al., 2026; Wang et al., 2025; Qian et al., 2024; Fan et al., 2024; Singh et al., 2025; Lee et al., 2023; Baechler et al., 2024; Lu et al., 2024; Ma et al., 2026b). Pure-vision and vision-language-action systems instead emphasize interaction and action generation (Wang et al., 2024; Lin et al., 2025; Hong et al., 2024; Zhang et al., 2025; You et al., 2024). Benchmarks expose difficulty with high-resolution or ambiguous interfaces (Li et al., 2025; Jandial et al., 2026), while recent audits warn that visible-label recovery can be mistaken for semantic grounding (Chen & Jacucci, 2026). We ask whether hierarchy changes final selection when candidates and upstream scores are fixed.

Structured interfaces. Mind2Web provides candidate-based web action selection, while WebArena and OSWorld evaluate execution in interactive environments rather than offline reranking (Deng et al., 2023; Zhou et al., 2024; Xie et al., 2024). RICO supplies mobile interface hierarchies and MUG studies interactive multimodal mobile grounding (Deka et al., 2017; Li et al., 2024a). AndroidControl provides demonstrations and accessibility trees; AndroidControl-Curated is a separate annotation variant, whereas our experiments use the original released manifest (Li et al., 2024b; Leung et al., 2025). AMEX adds multi-level mobile annotations, and GUIrilla provides macOS desktop accessibility data (Chai et al., 2025; Garkot et al., 2025). We use these resources for held-out ranking replication after refitting under their respective schemas.
RelationGUI/RelationAgent models spatial, functional, and hierarchical relations through a cross-platform dataset and relation-aware MLLM pipeline (Ma et al., 2026a); its target is end-to-end relational understanding. Wang et al.'s proposal-and-critic framework changes the proposal and critic loop, while RankGround reranks visual crops before a VLM call (Wang et al., 2026; Fan et al., 2026). In contrast, we freeze the candidate set, matcher, and upstream semantic scores to isolate the effect of accessibility-tree position on final selection.

Ranking and interaction. Learning-to-rank distinguishes pointwise, pairwise, and listwise objectives (Cao et al., 2007). Deep Sets and Set Transformer provide permutation-aware alternatives for unordered candidates (Zaheer et al., 2017; Lee et al., 2019), while graph attention networks (GAT) and GraphSAGE propagate information through explicit neighborhoods (Veličković et al., 2018; Hamilton et al., 2017). These models can use other candidates or graph neighborhoods, but their value depends on whether those relations are available and reliable. Raw model scores also need not be calibrated confidence estimates (Guo et al., 2017); our fixed upstream score distribution is only a control signal, not target-derived supervision. We therefore test candidate interaction rather than assume it is beneficial.

3 Problem Formulation

For an instruction and interface state q, let $\mathcal{C}_q = \{c_{q1}, \ldots, c_{qN_q}\}$ be the fixed candidate set and let s_{qi} be the frozen Stage–1 score. A predefined target matcher produces $\mathcal{T}_q \subseteq \mathcal{C}_q$, and a final score z_{qi} selects

$$\hat{i}_q = \arg\max_i z_{qi}, \qquad \mathrm{Hit}(q) = \mathbf{1}[c_{q\hat{i}_q} \in \mathcal{T}_q]. \tag{1}$$

The candidate set, target matcher, and evaluator are frozen. A change in top-1 accuracy comes only from reordering the same candidates.

When a target node identifier is available, the matcher includes every candidate whose `node_id` equals that identifier. Otherwise, it selects the smallest candidate box containing the recorded target point. The matcher does not use the instruction, semantic score, learned residual, or test outcome. A row is correct when the selected candidate belongs to $\mathcal{T}_q$; target multiplicity is an evaluation property, not a reranker input.

In the canonical AndroidControl manifest, target identifiers are unavailable for all 4,381 valid rows, so every row uses the smallest-containing-box fallback. We call a feature family F matcher-aligned under matcher M when, given the fixed candidates and target annotation, it suffices to determine target membership without the instruction or Stage–1 score. Here, containment and area partly encode the fallback assignment, although the target point is not a model input. Let $\Delta_{M,F}(h)$ denote the paired accuracy change for residual head h using F and evaluated under M. It measures utility for that matcher-feature pair, not matcher-invariant semantic disambiguation; matcher and feature changes are sensitivity analyses, not causal decompositions. The raw candidate cohort is unavailable, so proposal recall is outside this estimand.

For a test set $\mathcal{Q}$ partitioned into package groups $\{\mathcal{Q}_p\}_{p\in\mathcal{P}}$, we report both row-weighted and package-equal paired changes:

$$\Delta_{\mathrm{row}} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\big(\mathrm{Hit}_q^{\mathrm{ours}} - \mathrm{Hit}_q^{(0)}\big), \qquad \Delta_{\mathrm{pkg}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\frac{1}{|\mathcal{Q}_p|}\sum_{q\in\mathcal{Q}_p}\big(\mathrm{Hit}_q^{\mathrm{ours}} - \mathrm{Hit}_q^{(0)}\big). \tag{2}$$

The main table uses Δ_{row}; uncertainty is estimated from Δ_{pkg} with package-cluster bootstrap resampling.

The frozen lexical baseline is

$$z_{qi}^{(0)} = s_{qi}. \tag{3}$$

Concretely, s_{qi} is the size of the token intersection between the instruction and the candidate's `text`, `content_desc`, `resource_id`, `class_name`, and `role` fields, divided by the number of instruction tokens (with a unit denominator for an empty instruction). Equal scores use the same deterministic evaluator key before and after reranking. The tree-position reranker adds a bounded residual:

$$\bar{u}_{qi,k} = \frac{u_{qi,k} - \mu_k^{\mathrm{train}}}{\widetilde{\sigma}_k^{\mathrm{train}}}, \qquad \delta_{qi} = B\tanh(f_\theta(\bar{u}_{qi})), \qquad z_{qi} = s_{qi} + \delta_{qi}, \quad B = 0.10, \tag{4}$$

where μ_k^{train} and $\sigma_k^{\mathrm{train}}$ are computed from the training split only. The implementation sets $\widetilde{\sigma}_k^{\mathrm{train}} = \sigma_k^{\mathrm{train}}$ when $\sigma_k^{\mathrm{train}} \geq \epsilon$ and to 1 otherwise, with fixed $\epsilon = 10^{-6}$. The bound makes the model a local reranker rather than a replacement for Stage-1 semantics. For any pair of candidates, the residual difference is at most $2B$, so a frozen-score margin larger than $2B$ cannot be reversed. The only pairs whose order can change belong to the bounded flip set

$$\mathcal{F}_q(B) = \{(i,j) : |s_{qi} - s_{qj}| \leq 2B\}. \tag{5}$$

4 TREE-POSITION RERANKING

4.1 STRUCTURAL REPRESENTATION

The released configuration uses eleven candidate-local input slots, corresponding to ten distinct quantities plus one legacy duplicate:

$$u_{qi} = [d_i, r_i, n_i^{par}, n_i^{child}, n_i^{par\text{-}sib}, n_i^{dup}, x_i, y_i, \Delta x_i, \Delta y_i, a_i]. \tag{6}$$

d_i is normalized depth and r_i sibling order. n_i^{par} and n_i^{child} are the direct parent and child counts, and $n_i^{par\text{-}sib}$ is the sum of the child counts of the direct parents. The released schema also contains n_i^{dup}, a legacy slot whose value is identical to n_i^{child}. It is retained in the reported checkpoints for exact numerical reproducibility and is not counted as an independent structural factor. The remaining terms are normalized center coordinates, absolute candidate-to-parent coordinate differences, and normalized area. All count and coordinate normalizations and fallback behavior are specified in Appendix G. The feature constructor uses no target, instruction, semantic score, or test outcome.

The position head is a small multilayer perceptron (MLP):

$$h_i = \tanh(W_1 \bar{u}_i + b_1), \qquad \delta_i = B \tanh(w_2^\top h_i + b_2). \tag{7}$$

The released configuration uses an $11 \to 16 \to 1$ head (209 trainable parameters, excluding optimizer state and metadata). The head is instruction-independent: it estimates a structural residual on the frozen semantic ranking, not an instruction-conditioned grounding score.

4.2 CANDIDATE-INTERACTION CONTROLS

We keep candidate tokens and the bounded output fixed while changing only the interaction operator. Unrestricted multi-head self-attention (MHSA) is

$$H' = H + \mathrm{MHSA}(H), \qquad \alpha_{ij} = \mathrm{softmax}_j \left(\frac{(W_Q h_i)^\top (W_K h_j)}{\sqrt{d}} \right). \tag{8}$$

For a relation-masked control, j is restricted to a fixed neighborhood $\mathcal{N}(i)$:

$$\alpha_{ij} = 0 \text{ if } j \notin \mathcal{N}(i), \qquad |\mathcal{N}(i)| \le k. \tag{9}$$

On AndroidControl, the mask comes from the accessibility tree. On AMEX, flat element annotations require a deterministic undirected $k = 4$ nearest-center graph. The mask is geometry-derived on AMEX and tree-derived on AndroidControl. The mask audit reports mean allowed degree 7.60 including self on AndroidControl and 4.80 neighbors per node on AMEX.

Because the candidate list has no semantic order, the reranker should be equivariant to a joint permutation of candidates and their structural edges. For a permutation π, the desired property is

$$z_{q,\pi(i)}(\pi(\mathcal{C}_q), \pi(G_q)) = z_{qi}(\mathcal{C}_q, G_q). \tag{10}$$

The pointwise head satisfies this through shared weights; attention controls must additionally permute their mask with the candidate identities. The candidate-order audit in the supplement checks this implementation property separately from accuracy.

For this ladder, each candidate uses the same 12-dimensional input: the eleven standardized features plus the frozen lexical score. The pointwise and attention controls use width 32 and approximately 9K parameters; attention adds one four-head self-attention block and a $32 \to 64 \to 32$ feed-forward block. Parameter matching applies within this ladder, so the comparison isolates the interaction operator under a shared candidate budget, matcher, bound, and tie rule. This matching is local to the ladder and does not make its roughly 9K parameters capacity-matched to the 209-parameter headline head.

4.3 TRAINING OBJECTIVE

For each query, a listwise objective encourages target candidates to outrank non-target candidates:

$$\mathcal{L}_{list}(q) = -\log \frac{\sum_{i \in \mathcal{T}_q} \exp(z_{qi}/\tau)}{\sum_{j \in \mathcal{C}_q} \exp(z_{qj}/\tau)}. \tag{11}$$

All reported runs use $\tau = 1$. The headline checkpoint uses AdamW (Loshchilov & Hutter, 2019) (learning rate 2×10^{-3}, weight decay 10^{-4}), at most 20 epochs, four-epoch early stopping, and $B = 0.10$. The attention ladder uses the same candidate budget, matcher, bound, and tie rule, with a predeclared 12/3 epoch budget. Further training and gating variants are reported in the appendix.

Inference and invariants. At inference, each manifest is converted to the same candidate-local features, standardized with training-split statistics, and passed through the shared head. The frozen Stage-1 score plus bounded residual is sorted with the evaluator's fixed tie rule. Feature construction uses no target, instruction, or test-set statistic; within each comparison, the candidate budget, matcher, bound, and tie rule are fixed.

Exact-tie policy probe. The fixed lexical score also supplies a label-free ambiguity signal. Let

$$g_q^{\mathrm{tie}} = \mathbf{1}\left[\left|\left\{i : s_{qi} = \max_j s_{qj}\right\}\right| > 1\right], \qquad z_{qi}^{\mathrm{tie}} = s_{qi} + g_q^{\mathrm{tie}}\delta_{qi}. \tag{12}$$

This auxiliary policy activates the trained residual only when the top lexical score is shared; otherwise it preserves the frozen order. Near ties with a positive score gap remain eligible under the bounded model but are not activated by this strict rule.

5 Evaluation Protocol

5.1 Datasets and splits

We separate the evidence into one primary setting and boundary diagnostics. Package-disjoint AndroidControl supplies the primary estimate. MUG–RICO and AMEX provide schema-boundary audits; GUIrilla provides an app-disjoint accessibility-tree replication after target-domain refitting. These settings are not averaged into the AndroidControl headline.

Primary setting: AndroidControl. We use the AndroidControl benchmark (Li et al., 2024b) with a package-disjoint split of 43,355/3,260/4,381 train/dev/test states; the test set spans 78 held-out packages. Accessibility-tree fields provide position features and the one-hop relation mask.

The released manifest begins after candidate extraction. All 4,381 test rows have finite scores, rankable candidate lists, and an in-candidate target; the raw proposal cohort is unavailable, so candidate recall is not measured.

The lexical baseline is 27.231% on every seed. All three upstream operating points reuse these test candidates and target assignments. The structural head receives no screenshot pixels. BM25 scores each row's fixed candidate cohort and is scaled to $[0, 1]$ before fitting a structural head. The stronger text-only upstream is a hashed-token EmbeddingBag–MLP trained on instruction and candidate text; its scores are frozen before structural reranking.

Boundary settings. MUG (interactive multimodal grounding) instructions and target object records are joined with RICO (a mobile interface hierarchy dataset) (Li et al., 2024a; Deka et al., 2017). The primary evaluation is app-disjoint, with 9,659 test rows and 681 held-out apps. The lexical baseline is 16.523%.

We use a clean package-disjoint subset of AMEX (Chai et al., 2025) (22,132 rows; 17,348/2,830/1,954 train/dev/test over 73 packages) and a frozen $k = 4$ geometry graph for sparse attention.

GUIrilla-Task is a non-Android, app-disjoint macOS accessibility-tree diagnostic (Garkot et al., 2025). After filtering, it has 972 test clicks from 176 apps; images are unused, and retained target node IDs provide a complementary matcher check.

5.2 Statistical protocol

Every model is independently trained with seeds 17, 23, and 42. We report row-level top-1 accuracy and paired changes in percentage points (pp). For the primary estimate, we average paired package deltas and give 78 packages equal weight, then use 2,000 fixed-seed (53) bootstrap samples for the

empirical 95% confidence interval (CI). This conditions on the split and saved checkpoints rather than integrating training randomness. The pre-specified ladder compares the lexical reference, tree-position correction, candidate-local BM25 operating point, and interaction-ladder-matched controls. Every condition uses the same deterministic key to resolve equal scores. Test results do not select models, scales, or seeds; AndroidControl remains the primary claim and other settings delimit its interpretation.

Estimand. Our estimand is paired change in final top-1 selection conditional on a fixed candidate manifest. Candidate availability, target assignments, and Stage–1 scores stay constant, so each change comes from reordering candidates already present. This isolates a final-selection component; end-to-end success also depends on candidate recall and action execution. The package-disjoint split tests whether the correction transfers beyond the applications used to train it.

6 RESULTS

Table 1 compares lexical, BM25, learned text-only, smallest-area, and TopoRerank results on fixed AndroidControl candidates. Figure 2 shows the lexical and BM25 operating points under the released and nearest-center protocols. UGround's screenshot-based point predictions (Gou et al., 2025) are projected onto the same candidates as an external visual reference (Appendix Table A27); the input modality and prediction stage differ, so this is not a protocol-matched reranker comparison.

6.1 ANDROIDCONTROL: THREE UPSTREAM OPERATING POINTS

Table 1 compares three frozen Stage–1 settings under matched candidate, matcher, split, and evaluator conditions. BM25 is the primary retrieval point, lexical scoring tests ambiguity, and learned text-only scoring provides a stronger upstream. Reranking improves each seed at all three points, with a smaller mean gain over the learned ranker.

The 4,381-row manifest has 100% target coverage, a median of 91 candidates, and 77.88% of top-1 margins within $2B$. At the lexical point, accuracy is 32.92% (+5.69 pp row-weighted); Section 7 reports the package-equal estimate. The reachable set covers 3,656 rows (83.45%), including 2,463/3,188 baseline errors; exact ties cover 2,519 (57.50%; Appendices E–F). Detailed package stress tests are in Appendix H.2.

Matcher sensitivity. Nearest-center matching changes 1,046/4,381 target assignments (23.88%) with full coverage; mean gain remains $+3.43$ pp (Appendix E).

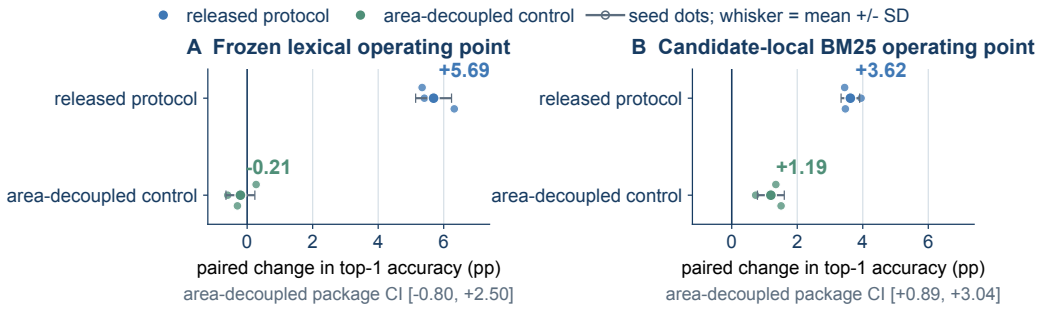


Figure 2: AndroidControl operating-point and area-free controls. Blue and teal denote the released and nearest-center protocols; dots show seed means and whiskers show SD.

The sensitivity controls locate where the lexical result depends on the interface representation. The deduplicated 10-slot refit gives $+6.277$ pp row-weighted and $+2.498$ pp package-equal (78-package bootstrap 95% CI $[+0.998, +4.121]$); we report it separately rather than replacing the original 11-slot primary result (Table A23). Joint removal of area, child count, and the alias gives -1.986 pp. Under the nearest-center, area-free matcher, lexical and tree-only changes are -0.205 and -0.571 pp. At BM25, the area-free control gives $+1.195$ pp row-weighted and $+1.844$ pp package-equal (95% CI $[+0.893, +3.037]$; Table 2).

Table 1: Fixed-candidate top-1 accuracy (%) on package-disjoint AndroidControl. Each upstream pair shares candidates, matcher, split, and evaluator; the final column gives row-weighted paired change from its frozen Stage–1 reference.

Model	Seed 17	Seed 23	Seed 42	Mean Δ (pp)
BM25 operating point (base 32.55)				
BM25 Stage–1 (protocol-matched control)	32.55	32.55	32.55	0.00 (BM25)
BM25 + TopoRerank (ours)	36.02	36.50	36.00	**+3.62 (BM25)**
Learned text-only operating point (base 39.44)				
Learned text-only Stage–1 (in-domain)	39.26	39.83	39.21	0.00 (text)
Learned text-only + TopoRerank	39.60	40.24	40.06	+0.53 (text)
Lexical operating point (base 27.23)				
Frozen lexical Stage–1 (internal reference)	27.23	27.23	27.23	0.00 (lex.)
Smallest-area rule (calibration)	30.24	30.24	30.24	+3.01 (lex.)
TopoRerank (ours)	33.55	32.64	32.57	**+5.69 (lex.)**

Table 2: Area-free sensitivity on AndroidControl with the nearest-center matcher. Package changes are package-equal with 95% bootstrap intervals; the BM25 reference differs from the released-matcher result in Table 1.

Operating point	Base	Area-free	Δ (pp)	Pkg. Δ [95% CI]
Lexical / nearest-center	26.30	26.09	-0.205	+0.860 $[-0.803, +2.503]$
BM25 / nearest-center	30.77	31.96	**+1.195**	**+1.844** $[+0.893, +3.037]$

The nearest-center controls vary matcher and feature set jointly, so they are sensitivity checks rather than a feature decomposition. The label-free exact-tie rule in Eq. 12 is a separate deployment option; the always-on model remains the headline.

With the released matcher, query-local BM25 (Robertson & Zaragoza, 2009) rises from 32.55% to 36.17% with the structural head (+3.62 pp across seeds; package-equal +1.263 pp, 78-package bootstrap 95% CI $[+0.260, +2.325]$). The label-free smallest-area rule gains +1.507 pp, 2.115 pp less than the learned head. Table 2 instead uses the nearest-center matcher (30.77% BM25 baseline); per-seed details are in Table A16.

6.2 Candidate-interaction ladder

The main interaction controls are summarized in Table 3; complete seed values and ladder-matched input qualification are in Table A15. The bounded tree-position head has the largest mean gain in this ladder under the released protocol.

6.3 AndroidControl stress tests

Tree-only retraining retains +4.54 pp, versus +0.32 pp for geometry-only; shuffling all position features reverses the gain (Table A31). As baseline top-1 rises from 27.23% (lexical) to 32.55% (BM25) and 39.44% (learned text-only), mean gains fall from +5.69 to +3.62 and +0.53 pp. Topology shuffle above the learned upstream is +0.03 pp (Table A20). Figure 3 summarizes tie and within-$2B$ mass at the first two points. Three points suggest attenuation, not a calibrated dose-response.

At BM25, the exact-tie-only policy gains +2.74 pp versus +3.62 pp for always-on correction (Table A17). The latter also gains in every non-exact margin bin below $2B$ (+2.58, +3.61, and +5.10 pp), with no changes beyond the bound (Figure 3). A frozen- checkpoint sweep over $B \in \{0.05, 0.10, 0.20, 0.50\}$ remains positive ($+5.09$–$+5.69$ pp); this varies the inference bound without refitting the head (Appendix J.1, Table A30).

Table 3: Candidate-interaction ladder on package-disjoint AndroidControl. Values are mean paired changes in top-1 accuracy over frozen lexical Stage–1, in percentage points across three seeds.

Control	mean Δ (pp)
Pointwise-deep	-0.02
Unrestricted attention	$+4.08$
One-hop sparse attention	$+0.14$
Linear listwise fusion	$+4.25$
Pairwise logistic residual (Burges et al., 2005)	$+3.71$
Tree-position head	$+5.69$

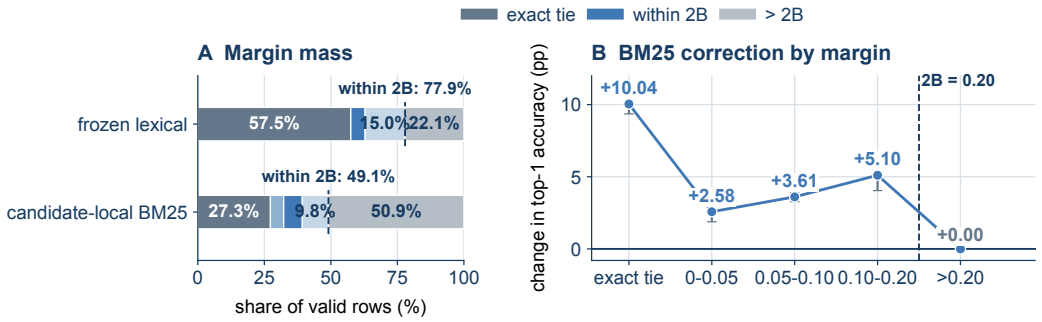


Figure 3: AndroidControl top-two margins and BM25 gains by margin bin. Panel A marks within-$2B$ mass; exact shares are in Table A4. Panel B's > 0.20 bin is outside the bound.

6.4 Cross-platform replication after refitting

On app-disjoint GUIrilla, target-domain refitting yields +6.036 pp over frozen lexical Stage–1 on 972 click rows from 176 held-out apps (app-equal +4.69 pp, 95% CI $[+2.11, +7.01]$). Table 4 gives shuffle controls; Appendix I reports the app-level audit.

Table 4: GUIrilla app-disjoint top-1 accuracy (%) after target-domain refitting on 972 click rows. The final column is paired change from frozen lexical Stage–1; shuffle controls preserve candidate/target rows.

Condition	Seed 17	Seed 23	Seed 42	Mean change
Frozen lexical Stage–1	24.07	24.07	24.07	0.000 pp
Original topology	30.66	29.84	29.84	+6.036 pp
Topology shuffle	24.69	26.85	27.26	+2.195 pp
Depth/sibling-only shuffle	28.81	29.42	28.50	+4.835 pp

Topology shuffle reduces the GUIrilla gain from +6.04 to +2.20 pp, while shuffling only depth and sibling order retains +4.84 pp. The first contrast shows sensitivity to the full topology assignment; the remaining gain indicates that retained features also matter.

MUG–RICO interaction controls produce no accuracy change across 9,659 rows; a tree-position refit was not run. In AMEX, sparse geometry is near zero, while unrestricted attention has a positive row mean but package intervals crossing zero (Tables A25 and A26).

7 Analysis

The package-equal gain is +1.82 pp (78-package bootstrap 95% CI $[+0.36, +3.27]$), below the row-weighted +5.69 pp because the estimates weight packages and rows differently. Leave-one-package-out means range from +1.57 to +2.19 pp. The interval supports a package-average gain on this split, not uniform gains; it conditions on the split and checkpoints, not training randomness. Figure 4 shows the package-level spread.

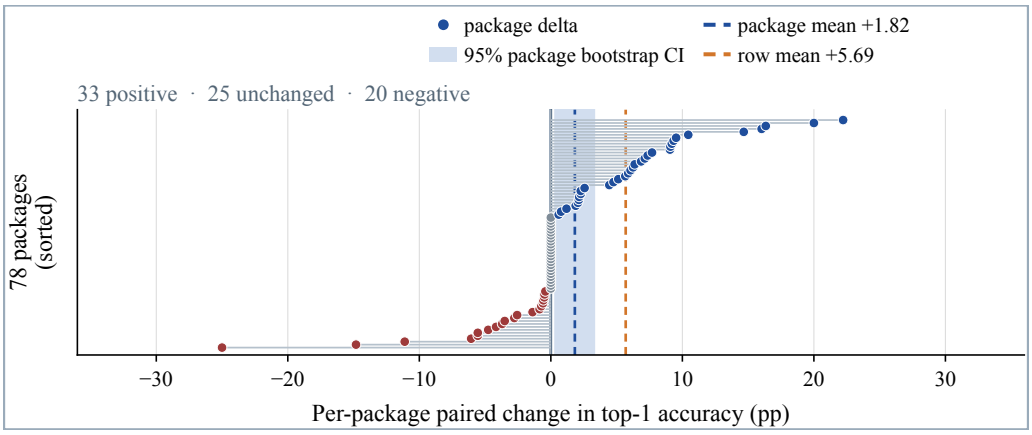


Figure 4: Paired changes for 78 held-out AndroidControl packages. The shaded band is the 95% package-cluster interval; dashed lines mark the package-equal and row-weighted means.

Bounded corrections resolve local ambiguity. For candidates i and j, the bounded score difference obeys

$$z_i - z_j = (s_i - s_j) + (\delta_i - \delta_j), \qquad |\delta_i - \delta_j| \le 2B. \tag{13}$$

Only candidates within $2B$ of the frozen winner can replace it; 77.88% of rows meet this margin. For winner $i_0(q)$, let $\mathcal{R}_B(q) = \{i : s_{qi_0} - s_{qi} \le 2B\}$. Then $\mathrm{Acc}_B^\star = N^{-1} \sum_q \mathbf{1}[\mathcal{T}_q \cap \mathcal{R}_B(q) \neq \varnothing]$ upper-bounds attainable accuracy. On the canonical replay, $\mathrm{Acc}_0 = 27.23\%$ and $\mathrm{Acc}_B^\star = 83.45\%$ ($H_B = 56.22$ pp); the +5.69 pp head gains 10.1% of this headroom. Test labels enter only this post-hoc bound. Of 3,188 baseline errors, 2,463 have a reachable target; the fitted head corrects 13.68–15.23% across seeds (Table A5).

Top-two margin mass within $2B$ falls from 77.9% under lexical Stage–1 to 49.1% under BM25, reducing but not fully explaining attenuation (Figure 3; Table A4). On 2,519 exact-tie rows (57.50%), random tie order gives +5.924 pp over 100 draws, versus +5.691 pp with the deterministic key (Table A10).

With the learned text-only EmbeddingBag–MLP, the residual adds +0.533 pp versus +0.030 pp after topology shuffle, but its package-equal interval crosses zero (Tables 1 and A20).

At the lexical point, a deeper pointwise control changes accuracy by -0.02 pp; unrestricted and one-hop sparse attention give $+4.08$ and $+0.14$ pp, versus $+5.69$ pp for the local head (Table 3).

Among 3,528 non-nested top-two rows, the row-weighted gain is +4.60 pp; among 1,028 exact-tie non-nested rows, it is +5.97 pp; both package-equal intervals cross zero (Table A8).

Limitations. Evaluation fixes candidates and matcher, not recall or live completion. Package changes remain heterogeneous (sign-test $p = 0.098$; Table A18). Across seeds, 86.3–87.5% of residuals lie within 1% of the cap ($|\delta_i| \ge 0.99B$); recalibration to new upstream scales is untested. GUIrilla requires target-domain refitting. Because the head excludes instruction text, an instruction-independent position prior remains an alternative explanation not separated by the current controls.

8 Conclusion

TopoRerank adds a bounded local-structure correction to frozen GUI rankings. Across the controlled AndroidControl settings, the gain is largest for lexical ranking and attenuates under a learned text-only ranker. The area-free matcher preserves a positive package-level interval at the BM25 point, while the lexical result is not robust to that matcher change. App-disjoint GUIrilla extends the finding to a macOS accessibility schema after target-domain refitting, rather than establishing zero-shot transfer. Together, these results position hierarchy as complementary evidence for final-choice ambiguity, not as a substitute for candidate recall or semantic grounding.

Reproducibility Statement

The anonymous supplement provides model and audit source, split summaries, manifest schemas, seed/configuration metadata, and aggregate result audits. The reported tables use canonical manifests, checkpoints, matcher, feature constructor, and evaluator identified by recorded hashes. Restricted manifests, raw screenshots, and the canonical tree-position and UGround model checkpoints are not included. The supplement does include three compact JSON state sidecars for replaying the BM25 operating-point audit. Exact numerical replay of the headline results therefore requires legally obtained inputs and the corresponding canonical checkpoint artifacts. The included synthetic smoke test checks the code path; it does not reproduce the paper results.

AI Use Statement

Generative AI tools assisted with language polishing, code debugging, and manuscript preparation under the authors' supervision. The authors set the research direction, designed the experimental protocol, ran the reported experiments, verified all numerical claims, and made the final analytical and presentation decisions. The authors take full responsibility for the content of this paper.

References

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Carbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. ScreenAI: A vision-language model for UI and infographics understanding. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3058–3068. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/339. URL https://www.ijcai.org/proceedings/2024/339.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96. ACM, 2005. doi: 10.1145/1102351.1102363. URL https://doi.org/10.1145/1102351.1102363.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pp. 129–136, 2007. doi: 10.1145/1273496.1273513.

Yuxiang Chai, Siyuan Huang, Yazhe Niu, Han Xiao, Liang Liu, Guozhi Wang, Dingyu Zhang, Shuai Ren, and Hongsheng Li. AMEX: Android multi-annotation expo dataset for mobile GUI agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2138–2156. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.110. URL https://aclanthology.org/2025.findings-acl.110/.

Qijia Chen and Giulio Jacucci. Lexical coupling in GUI element grounding: Sentence embeddings track labels across mobile and web, 2026. URL https://arxiv.org/abs/2608.21794. arXiv preprint.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 9313–9332. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.505. URL https://aclanthology.org/2024.acl-long.505/.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pp. 845–854, 2017. doi: 10.1145/3126594.3126651. URL https://doi.org/10.1145/3126594.3126651.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 28091–28114, 2023.

doi: 10.52202/075280-1220. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html. Datasets and Benchmarks Track.

Liyang Fan, Xinping Bi, Yitai Li, Shuaimin Li, Hui Li, and Min Yang. RankGround: Efficient high-resolution GUI grounding via lightweight reranker-guided crop selection, 2026. URL https://arxiv.org/abs/2609.18690. arXiv preprint; the arXiv record reports acceptance to ACM Multimedia 2026.

Yue Fan, Lei Ding, Ching-Chen Kuo, Shan Jiang, Yang Zhao, Xinze Guan, Jie Yang, Yi Zhang, and Xin Eric Wang. Read anywhere pointed: Layout-aware GUI screen reading with Tree-of-Lens grounding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9503–9522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.533. URL https://aclanthology.org/2024.emnlp-main.533/.

Sofiya Garkot, Maksym Shamrai, Ivan Synytsia, and Mariya Hirna. GUIrilla: A scalable framework for automated desktop UI exploration, 2025. URL https://arxiv.org/abs/2510.16051. arXiv preprint.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.05243. URL https://arxiv.org/abs/2410.05243. Oral.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. CogAgent: A visual language model for GUI agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290, 2024. doi: 10.1109/CVPR52733.2024.01354. URL https://openaccess.thecvf.com/content/CVPR2024/html/Hong_CogAgent_A_Visual_Language_Model_for_GUI_Agents_CVPR_2024_paper.html.

ZongHan Hsieh, ShengJing Yang, and Tzer-Jen Wei. ZonUI-3B: Competitive GUI grounding with a 3b VLM trained on a single consumer gpu. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 959–966, 2026. doi: 10.1109/WACV61042.2026.00100. URL https://openaccess.thecvf.com/content/WACV2026/html/Hsieh_ZonUI-3B_Competitive_GUI_Grounding_with_a_3B_VLM_Trained_on_WACV_2026_paper.html.

Surgan Jandial, Yinheng Li, Justin Wagle, and Kazuhito Koishida. Do GUI grounders truly understand UI elements? In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 2772–2785. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.144. URL https://aclanthology.org/2026.findings-eacl.144/.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3744–3753. PMLR, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 18893–18912. PMLR, 2023. doi: 10.48550/arXiv.2210.03347. URL https://proceedings.mlr.press/v202/lee23g.html.

Ho Fai Leung, Xiaoyan Xi, and Fei Zuo. AndroidControl-Curated: Revealing the true potential of GUI agents through benchmark purification, 2025. URL https://arxiv.org/abs/2510.18488. arXiv preprint.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*, pp. 8778–8786. ACM, 2025. doi: 10.1145/3746027.3755688. URL https://doi.org/10.1145/3746027.3755688.

Tao Li, Gang Li, Jingjie Zheng, Purple Wang, and Yang Li. MUG: Interactive multimodal grounding on user interfaces. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 231–251. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.findings-eacl.17. URL https://aclanthology.org/2024.findings-eacl.17/.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024b. doi: 10.52202/079017-2925. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a79f3ef3b445fd4659f44648f7ea8ffd-Abstract-Datasets_and_Benchmarks_Track.html.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. ShowUI: One vision-language-action model for GUI visual agent. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19498–19508, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lin_ShowUI_One_Vision-Language-Action_Model_for_GUI_Visual_Agent_CVPR_2025_paper.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. OmniParser for pure vision based GUI agent, 2024. URL https://arxiv.org/abs/2408.00203. arXiv preprint.

Longhui Ma, Di Zhao, Siwei Wang, Zhao Lv, and Miao Wang. Beyond element-level understanding: Explicit relational understanding for GUI agents. *Pattern Recognition*, 176:113262, 2026a. doi: 10.1016/j.patcog.2026.113262. URL https://www.sciencedirect.com/science/article/pii/S003132032600227X.

Longhui Ma, Di Zhao, Siwei Wang, Zhao Lv, and Miao Wang. Trifuse: Enhancing attention-based gui grounding via multimodal fusion, 2026b. URL https://arxiv.org/abs/2602.06351. arXiv preprint.

Joonhyung Park, Peng Tang, Sagnik Das, Srikar Appalaraju, Kunwar Yashraj Singh, R. Manmatha, and Shabnam Ghadar. R-VLM: Region-aware vision language model for precise GUI grounding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9669–9685. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.501. URL https://aclanthology.org/2025.findings-acl.501/.

Yijun Qian, Yujie Lu, Alexander Hauptmann, and Oriana Riva. Visual grounding for user interfaces. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Industry Track*, pp. 97–107. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-industry.9. URL https://aclanthology.org/2024.naacl-industry.9/.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Kunal Singh, Shreyas Singh, and Mukund Khanna. TRISHUL: Towards region identification and screen hierarchy understanding for large VLM-based GUI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 170–179, 2025. doi: 10.1109/CVPRW67362.2025.00022. URL https://openaccess.thecvf.com/content/CVPR2025W/MULA2025/html/Singh_TRISHUL_Towards_Region_Identification_and_Screen_Hierarchy_Understanding_for_Large_CVPRW_2025_paper.html.

Fei Tang, Yongliang Shen, Hang Zhang, Siqi Chen, Guiyang Hou, Wenqi Zhang, Wenqiao Zhang, Kaitao Song, Weiming Lu, and Yueting Zhuang. Think twice, click once: Enhancing gui grounding via fast and slow systems, 2025. URL https://arxiv.org/abs/2503.06470. arXiv preprint.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10903.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191. arXiv preprint.

Suyuchen Wang, Tianyu Zhang, Ahmed Masry, Christopher Pal, Spandana Gella, Bang Liu, and Perouz Taslakian. Improving GUI grounding with explicit position-to-coordinate mapping, 2025. URL https://arxiv.org/abs/2510.03230. arXiv preprint.

Wenkai Wang, Xiyun Li, Hongcan Guo, Wenhao Yu, Tianqing Fang, Haitao Mi, Dong Yu, and Shengyu Zhang. Measure twice, click once: Co-evolving proposer and visual critic via reinforcement learning for GUI grounding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20964–20984. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.960. URL https://aclanthology.org/2026.acl-long.960/.

Qianhui Wu, Kanzhi Cheng, Rui Yang, Chaoyun Zhang, Jianwei Yang, Huiqiang Jiang, Jian Mu, Baolin Peng, Bo Qiao, Reuben Tan, Si Qin, Lars Liden, Qingwei Lin, Huan Zhang, Tong Zhang, Jianbing Zhang, Dongmei Zhang, and Jianfeng Gao. GUI-Actor: Coordinate-free visual grounding for GUI agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.52202/085713-0508. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/16130af940e9dabb43c726119bd3b42e-Abstract-Conference.html. Main Conference Track.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems 37, Datasets and Benchmarks Track*, 2024. doi: 10.52202/079017-1650. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Yuhao Yang, Yue Wang, Dongxu Li, Ziyang Luo, Bei Chen, Chao Huang, and Junnan Li. Aria-UI: Visual grounding for GUI instructions. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22418–22433. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1152. URL https://aclanthology.org/2025.findings-acl.1152/.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-UI: Grounded mobile UI understanding with multimodal LLMs. In *Computer Vision – ECCV 2024*, volume 15122, pp. 240–255. Springer, 2024. doi: 10.1007/978-3-031-73039-9_14. URL https://doi.org/10.1007/978-3-031-73039-9_14.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. AppAgent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1–20. ACM, 2025. doi: 10.1145/3706598.3713600. URL https://doi.org/10.1145/3706598.3713600.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

This appendix records the complete fixed-candidate protocol, model configurations, seed-level results, replication boundaries, mechanism audits, and reproduction guidance for the accompanying manuscript. The main paper is self-contained; this document adds auditability rather than new headline claims. All reported comparisons use the same candidate manifest, target matcher, split, tie rule, and evaluator within a setting.

Every valid result used to support a scientific claim is listed here, including null results and intervals that cross zero. Operational failures and superseded source snapshots are documented separately in the artifact provenance record rather than treated as benchmark observations.

A Scope and Claim Boundary

The supported claim is that a small bounded tree-position residual can improve a frozen candidate ranking on package-disjoint AndroidControl, with positive but smaller evidence after target-domain refitting on a held-out macOS accessibility-tree subset. Candidate interaction is an experimental factor, not the headline method. MUG–RICO and AMEX provide schema-boundary evidence; the study evaluates offline candidate reranking, not live-agent task completion. Because the head excludes instruction text, a position prior remains an alternative explanation that the current controls do not fully separate from relational structure.

B Research Questions and Evidence Map

The appendix uses the same evidence hierarchy as the main paper. The primary claim is the package-disjoint AndroidControl estimate. Controls test alternative explanations, and external settings delimit replication; they are not pooled into a cross-dataset score.

Table A1 maps each research question to the corresponding evidence block and its intended interpretation.

Table A1: Mapping from research questions to evidence and interpretation.

Question	Evidence	Interpretation
Q1: Frozen ranking correction	AndroidControl main result, three seeds, package bootstrap	Primary estimand; row-weighted and package-equal values are both reported.
Q2: Alternative explanations	Pointwise, unrestricted, sparse, linear, shuffle, and feature-family controls	Mechanism evidence; not additional headline benchmarks.
Q3: Boundary conditions	Learned text-only upstream, MUG–RICO, AMEX, GUIrilla	Boundary evidence; null or uncertain intervals remain visible.
Q4: Artifact auditability	Manifest checks, fixed tie key, split summaries, seed outputs, and source-to-table mapping	Reproducibility evidence; not a raw-data redistribution claim.

This separation prevents a row-weighted AndroidControl effect, an external diagnostic, and a null cross-setting result from being interpreted as one undifferentiated average. It also identifies which results would change the main claim and which would only change its scope.

C Fixed-Candidate Protocol

For an instruction and interface state q, the immutable candidate set is $\mathcal{C}_q = \{c_{q1}, \ldots, c_{qN_q}\}$. A frozen Stage–1 system supplies semantic scores s_{qi}. A fixed target matcher defines $\mathcal{T}_q \subseteq \mathcal{C}_q$, and every model selects

$$\hat{i}_q = \arg\max_i z_{qi}, \qquad \mathrm{Hit}(q) = \mathbf{1}[c_{q\hat{i}_q} \in \mathcal{T}_q].$$

The main reranker is

$$\bar{u}_{qi,k} = \frac{u_{qi,k} - \mu_k^{\mathrm{train}}}{\widetilde{\sigma}_k^{\mathrm{train}}}, \qquad \delta_{qi} = B \tanh(f_\theta(\bar{u}_{qi})), \qquad z_{qi} = s_{qi} + \delta_{qi}, \quad B = 0.10.$$

Here μ_k^{train} and σ_k^{train} are computed from training rows only. The implementation sets $\widetilde{\sigma}_k^{\text{train}} = \sigma_k^{\text{train}}$ when $\sigma_k^{\text{train}} \geq \epsilon$ and to 1 otherwise, with fixed $\epsilon = 10^{-6}$. The eleven candidate-local structural and geometric features are standardized without using development or test statistics. The candidate set, target matcher, split, evaluator, and tie key are unchanged. Since $|\delta_i| \leq B$, a score margin larger than $2B$ cannot be reversed by the residual alone. The method is consequently a local ambiguity resolver rather than a replacement for semantic candidate retrieval.

No target identity, target box, evaluator-only field, or post-hoc test statistic is included in the model features. Test labels are used only for the final evaluation.

When a target node identifier is available, the matcher includes every candidate with the same `node_id`; otherwise it selects the smallest candidate box containing the recorded target point. Any candidate in $\mathcal{T}_q$ counts as a hit, so the manifest can encode multiple positives when identifiers are duplicated. The matcher does not use the instruction, semantic score, learned residual, or test outcome.

The canonical AndroidControl manifest has no usable target identifier in any of its 4,381 valid rows, so the point-based fallback is used throughout. Because that fallback selects the smallest containing box, area and relative position are partly coupled to the evaluation convention. The smallest-area rule is therefore a calibration control, not evidence of a causal feature fraction or target leakage: the feature constructor still does not read target metadata. The released candidate manifest does not include the raw candidate cohort, so a node-ID or IoU alternative would require a different evaluation artifact and is not claimed here.

Locality decomposition. Let $s_{q(1)}$ and $s_{q(2)}$ denote the two largest frozen scores and define $\mathcal{A} = \{q : s_{q(1)} - s_{q(2)} \leq 2B\}$. The bounded residual cannot change the top-ranked candidate outside this set. Therefore the row-weighted paired change satisfies

$$\Delta_{\text{row}} = \Pr(\mathcal{A}) \left[\text{Acc}_{\text{ours}}(\mathcal{A}) - \text{Acc}_0(\mathcal{A})\right]. \tag{14}$$

Since conditional accuracies lie in $[0, 1]$,

$$-\Pr(\mathcal{A})\,\text{Acc}_0(\mathcal{A}) \leq \Delta_{\text{row}} \leq \Pr(\mathcal{A}) \left[1 - \text{Acc}_0(\mathcal{A})\right]. \tag{15}$$

This is an operating-range certificate, not a causal decomposition or a guarantee of improvement. For the canonical AndroidControl replay, $\Pr(\mathcal{A}) = 77.88\%$.

The lexical score uses lowercased word-token sets: it counts the intersection between instruction tokens and the concatenated candidate fields `text`, `content_desc`, `resource_id`, `class_name`, and `role`, then divides by $\max(1, |\text{instruction tokens}|)$. Equal scores are ordered by the same deterministic evaluator key in every condition.

C.1 Computational complexity

For a query with N candidates and feature width d, the pointwise head computes one shared residual per candidate:

$$T_{pointwise} = O(Nd). \tag{16}$$

Unrestricted candidate self-attention forms all candidate pairs:

$$T_{full} = O(N^2 d). \tag{17}$$

If an implementation stores and computes only k relation neighbors per candidate, the corresponding interaction cost is

$$T_{sparse} = O(Nkd). \tag{18}$$

These are asymptotic comparison costs for the reranking head; the upstream semantic score and candidate construction are held fixed in all three cases. The released sparse control constructs a dense $N \times N$ mask and invokes standard multi-head attention, so it restricts connectivity but should not be read as an empirical $O(Nkd)$ runtime implementation.

D Datasets and Frozen Splits

Table A2 records the frozen split sizes, group structure, and role of each evaluated setting.

Table A2: Data and split audit. Counts are rows or states after the stated candidate-coverage filter.

Setting	Train	Dev	Test	Groups	Role
AndroidControl	43,355	3,260	4,381	78 packages	Main native hierarchy result
MUG–RICO	77,268	9,659	9,659	5,457/666/681 apps	App-disjoint null boundary
AMEX	17,348	2,830	1,954	51/11/11 packages	Flat metadata / geometry proxy
GUIrilla-Task	21,285	1,579	972	176 test applications	macOS app-disjoint structural diagnostic

Table A3: Candidate-space and bounded-flip audit on the fixed AndroidControl test manifest. Pair-flip rates are shown before and after removing exact lexical ties; bound violations exceed the $2B$ margin.

Seed	Cand. med.	Target rank med.	Top-1 margin $\leq 2B$	Pair flips (all/non-tie)	Bound violations	Top-1 help/harm
17	91	10	77.88%	39.31/11.00%	0	375 / 98
23	91	10	77.88%	39.60/12.89%	0	342 / 105
42	91	10	77.88%	37.79/10.31%	0	337 / 103

AndroidControl uses package-disjoint splits and exposes accessibility-tree parent/child relations, depth, sibling order, and normalized geometry. Web/DOM candidate-manifest diagnostics are outside the scope of this paper. MUG–RICO joins MUG target objects with RICO hierarchies and uses a deterministic app-disjoint split. AMEX exposes flat element records; its sparse relation control is therefore a four-nearest-center geometry graph, not an accessibility hierarchy. GUIrilla retains only visible, bounded click rows, uses application-disjoint splits, and does not use image features. The available test subset has 972 rows; all claims use this exact denominator rather than a larger nominal test count.

E Candidate Coverage and Bounded-Flip Audit

The primary AndroidControl manifest has 4,381 valid rows and 100% target-in- candidate coverage within the fixed evaluated manifest. This statistic is not proposal recall over the original raw interface states. Candidate counts have median 91 and IQR 50–154; the median initial target rank is 10, while the median lexical top-1 margin is 0 because lexical overlap is discrete. The following audit uses the same frozen checkpoints, candidate manifest, target matching, deterministic tie key, and bound $B = 0.10$ as the main evaluation. It is descriptive and does not select features, seeds, or thresholds.

Table A3 reports the candidate-space and bounded-flip quantities for each released checkpoint; Table A5 reports the corresponding target-aware baseline-error eligibility.

Each seed enumerates 116,637,295 candidate pairs. The maximum initial score gap among changed pairs is 0.20000002, attributable to floating-point representation of the $2B = 0.20$ boundary; no pair exceeds the bound after the stated numerical tolerance. The row-level top-1 changes are 1,732, 1,710, and 1,767 for seeds 17, 23, and 42, respectively. Full score-gap bins and aggregate pair counts are retained in the machine-readable audit outputs.

Table A4 gives the exact row counts behind Figure 3; the first four bins define the within-$2B$ mass.

The target-aware winner-scope audit additionally counts 3,656 of 4,381 rows (83.45%) as eligible, including rows that the lexical baseline already gets right. Among the 3,188 baseline errors, 2,463 are eligible, giving the 77.26% error-set rate shown above. This distinguishes target-aware error eligibility from the 77.88% all-row top-vs-second margin diagnostic in the preceding table.

Table A4: Exact top-two score-gap bins on the 4,381-row AndroidControl test set. Cells give count (share). Exact ties use a gap of at most 10^{-8}; subsequent bins use $(10^{-8}, 0.05]$, $(0.05, 0.10]$, $(0.10, 0.20]$, and > 0.20. The first four bins sum to the within-$2B$ mass reported in Figure 3.

Setting	$\leq 10^{-8}$	$(10^{-8}, .05]$	$(.05, .10]$	$(.10, .20]$	$> .20$
Frozen lexical	2,519 (57.50%)	1 (0.02%)	234 (5.34%)	658 (15.02%)	969 (22.12%)
Candidate-local BM25	1,195 (27.28%)	220 (5.02%)	305 (6.96%)	431 (9.84%)	2,230 (50.90%)

Table A5: Target-aware eligibility among lexical baseline errors on the fixed AndroidControl test manifest. Recovery uses the frozen tree-position checkpoint and is counted only within the $2B$ margin.

Seed	Baseline errors	Eligible errors	Eligible rate	Recovered	Recovery rate
17	3,188	2,463	77.26%	375	15.23%
23	3,188	2,463	77.26%	342	13.89%
42	3,188	2,463	77.26%	337	13.68%

For a target-aware reachable-set certificate, let $i_0(q)$ be the lexical winner and define $\mathcal{R}_B(q) = \{i : s_{qi_0} - s_{qi} \leq 2B\}$. Any B-bounded residual can change the winner only within this set. The corresponding oracle accuracy is $\mathrm{Acc}_B^\star = N^{-1} \sum_q \mathbf{1}[\mathcal{T}_q \cap \mathcal{R}_B(q) \neq \varnothing]$, and the score-reachability gap is $H_B = \mathrm{Acc}_B^\star - \mathrm{Acc}_0$. For the canonical AndroidControl replay, $\mathrm{Acc}_0 = 27.23\%$ and $\mathrm{Acc}_B^\star = 83.45\%$, giving $H_B = 56.22$ pp. The released tree-position head realizes $+5.69$ pp, or 10.1% of this score-reachability gap. This is an operating-range certificate under the released matcher; the oracle uses target labels for analysis only and is not used for training, threshold selection, or model selection.

At the predeclared score bound, 77.26% of baseline errors have a target in the reachable set. The three frozen checkpoints recover 13.68%–15.23% of those eligible errors. Eligibility thus measures an opportunity for correction on this manifest, whereas recovery measures the fitted head's decisions.

The 77.26% eligible-error rate is not an exact-tie rate; it includes positive score gaps up to $2B$. We therefore ran a separate tie-handling robustness audit without retraining. It reuses the same three checkpoints and fixed manifest, and randomizes only the order within equal-score groups using 100 reproducible repetitions per seed. An exact top lexical-score tie occurs in 2,519 of 4,381 valid rows (57.50%). In the current replay, the deterministic all-row gain has mean +5.691 pp, while randomized tie handling has mean +5.924 pp. On the exact-tie subset, the corresponding means are +10.229 pp and +10.611 pp; the per-seed randomized exact-tie gains are +11.537, +10.235, and +10.061 pp for seeds 17, 23, and 42. Because randomization changes only equal-score groups and the all-row correction remains positive, the correction does not collapse under this tie-handling perturbation. The exact-tie subset is nevertheless a different operating slice, so these numbers are a robustness audit rather than a causal decomposition of topology from all other local score gaps. The machine-readable outputs record the manifest, checkpoint, and current audit source hashes. For the complementary non-exact-top-score slice, the deterministic changes for seeds 17, 23, and 42 are -0.161, -0.644, and -0.537 pp over 1,862 rows (42.50%), with mean -0.448 pp. Since this slice has a unique lexical winner by construction, the result makes the operating boundary explicit: the observed improvement is concentrated in local lexical ambiguity rather than being a uniform gain on every row. The negative value is a slice-level aggregate, not a claim that every unique-score row is harmed. These values are deterministic-slice diagnostics; the randomized audit above remains the tie-handling robustness test. The current replay is the canonical estimate for the released source tree: the manifest, matcher, feature constructor, checkpoints, and evaluator are hashed in the machine-readable output. It supersedes an earlier historical export whose source snapshot was not preserved. Consequently, the precise historical-to-canonical cause cannot be reconstructed from the released artifacts, and we do not attribute it to a particular numerical, sorting, or dependency change. The historical values (+6.414, +5.410, and +5.204 pp for seeds 17, 23, and 42) remain in the artifact archive only as a provenance boundary; they are not mixed with the canonical replay or used in the main tables. The evidence therefore narrows the discrepancy to a provenance or protocol boundary, but the released artifacts do not identify whether the historical manifest, evaluator, or preprocessing differed.

E.1 Matcher protocol audit

The released AndroidControl manifest contains one target point per valid row; it does not contain a raw-candidate cohort from which proposal recall could be estimated. We therefore audit only the conditional fixed-manifest contract: candidate boxes are valid, the smallest containing candidate is selected by the target-point fallback, and no target is added or removed by the reranker.

The resulting matcher contract is summarized in Table A6.

Table A6: Target matcher audit on the released AndroidControl manifest.

Metric	Value
Rows in manifest	4,381
Rows with rankable candidates	4,381
Target coverage within manifest	4,381 (100.00%)
Target-node-ID matches	0
Smallest-containing-box fallbacks	4,381
Unmatched targets	0
Target multiplicity	1 in all 4,381 rows
Candidate count (median; IQR)	91; 50–154
Candidate count (min–max)	5–2,194

Table A7: Area-decoupled target-matcher sensitivity on the same 4,381 test rows. The model and evaluator are frozen; only target assignment changes.

Seed	Lexical	Corrected	Δ pp
17	26.30	30.45	+4.15
23	26.30	29.42	+3.13
42	26.30	29.31	+3.01
Mean	26.30	29.73	+3.43

The audit is intentionally narrower than raw proposal recall. Since all rows use the point fallback, the result is best read as a conditional estimate under the released matcher convention. The machine-readable result records the manifest and source hashes and explicitly marks the raw cohort as unavailable; this prevents conditional candidate coverage from being overstated as end-to-end grounding recall.

E.2 Area-decoupled target-matcher sensitivity

The released AndroidControl rows expose target points but no usable target node identifiers. To test whether the primary direction depends only on the smallest-area convention, we keep the candidate list, frozen checkpoints, feature values, lexical scores, tie key, and evaluator unchanged. The audit changes only the point-to-candidate assignment: among candidates containing the point, it selects the candidate whose box center is nearest to the point. This rule is independent of candidate area, while preserving the requirement that the selected target contains the recorded point.

The area-free rule has valid targets for all 4,381 rows and changes the target mapping on 1,046 rows (23.88%). Since no row has a usable target node ID, this is a sensitivity check on the released point annotation rather than a claim of recovered semantic node identity. Table A7 reports the frozen-checkpoint comparison. The positive direction remains, but the gain is smaller than under the released smallest-area matcher. This is consistent with a matcher-dependent operating point and does not remove the area feature from the model input.

The audit therefore weakens, but does not reverse, the concern that the headline direction is an artifact of the smallest-containing-box convention. It is not a causal feature ablation: the reranker still receives area, and the nearest-center rule is itself a geometric evaluation convention. We retain the released matcher as the canonical protocol and report this result as matcher-sensitivity evidence.

E.3 Non-nested local-candidate audit

The matcher audit above is complemented by a local geometric control. We retain the same manifest, lexical scores, checkpoints, tie key, and evaluator, but define two cohorts without changing any labels or candidate boxes: (i) rows whose two lexical top-ranked candidates have no strict box-containment relation, and (ii) exact lexical-tie rows whose tied top-score candidates have no such relation. The first cohort contains 3,528 rows and the second contains 1,028 rows. The cohort membership is determined from frozen candidate geometry and lexical scores before reading the corrected output.

Table A8 reports the two non-nested cohorts.

The replay therefore retains a sizeable row-weighted correction when the locally competing candidates are not nested. This reduces, but does not eliminate, concern that the result is explained only

Table A8: Local non-nested candidate audit. Values are means over the three released checkpoints; the row-weighted subset gain remains positive, while the smaller package-equal intervals are not used as a new headline estimate.

Cohort	Rows	Lexical	Corrected	Δ pp
Top-two non-nested	3,528	27.24	31.84	+4.60
Exact-tie non-nested	1,028	15.66	21.63	+5.97

Table A9: Tree-only nearest-center replay on the top-two non-nested cohort.

Seed	Rows	Row Δ (pp)	Package Δ (95% CI)
17	3,528	-0.312	-1.210 $[-2.639, +0.090]$
23	3,528	-0.283	-0.711 $[-1.999, +0.457]$
42	3,528	+0.737	-0.271 $[-1.634, +1.061]$

by the smallest-containing box convention. Because the cohorts are smaller and package composition is uneven, their package-equal bootstrap intervals cross zero; we treat this as matcher-sensitivity evidence rather than as a universal matcher-independent claim.

As a stricter joint boundary check, we replayed the same 3,528-row top-two non-nested cohort with a tree-only head and nearest-center target assignment. Table A9 shows the seed-level result. The row-weighted changes vary in sign and every package-equal interval includes zero, so this completed audit does not support promoting the subgroup to a headline claim; it instead bounds the claim when both the matcher and feature schema are changed.

F Tie-Handling Robustness Audit

For completeness, Table A10 reports the aggregate values from the frozen tie-handling audit. The deterministic column uses the current released ranking helper, and the randomized column averages 100 evaluations per checkpoint after shuffling only candidates with equal frozen lexical score. No scores, candidate identities, target matcher, structural checkpoint, or bound are changed. Figure A1 visualizes the seed-level changes.

The small increase under randomized handling is compatible with a change in which member of an equal-score group is selected; it is not interpreted as an additional method gain. In particular, the audit does not claim that all of the exact-tie subset is caused by topology, only that replacing the deterministic tie key does not remove the observed correction. The non-exact slice has mean deterministic change -0.448 pp across the three checkpoints.

To calibrate this result against simple alternatives, we also evaluated label-free secondary orders inside equal lexical-score groups. These rules use no training labels, target metadata, or learned parameters; a unique lexical winner is unchanged. Table A11 reports their paired changes from the deterministic lexical key. The strongest simple rule, choosing the smallest-area candidate, accounts for part of the paired correction (+3.013 pp over all rows and +5.240 pp on exact ties), but remains below the released tree-position checkpoints (+5.691 and +10.229 pp, respectively). The ratio between these changes is a descriptive calibration, not a causal fraction of the learned mechanism: the rule and the checkpoint are different ranking policies. The +2.336 pp change after permuting area is likewise a frozen- checkpoint sensitivity test, not a feature-level causal attribution or an estimate of the non-area contribution. Because the label-free rule and the permuted checkpoint differ in scoring rule and reorderable rows, their gains cannot be added into a feature decomposition; the difference from the full gain is not an identified Area contribution. Random tie breaking is slightly worse than the deterministic key. Thus the gain is not fully attributable to the evaluator's tie key, while

Table A10: Tie-handling robustness on the fixed AndroidControl test manifest. Entries are mean changes over the three checkpoints in percentage points.

Evaluated rows	Deterministic key	Randomized equal-score groups
All valid rows (4,381)	+5.691	+5.924
Exact top-score ties (2,519)	+10.229	+10.611

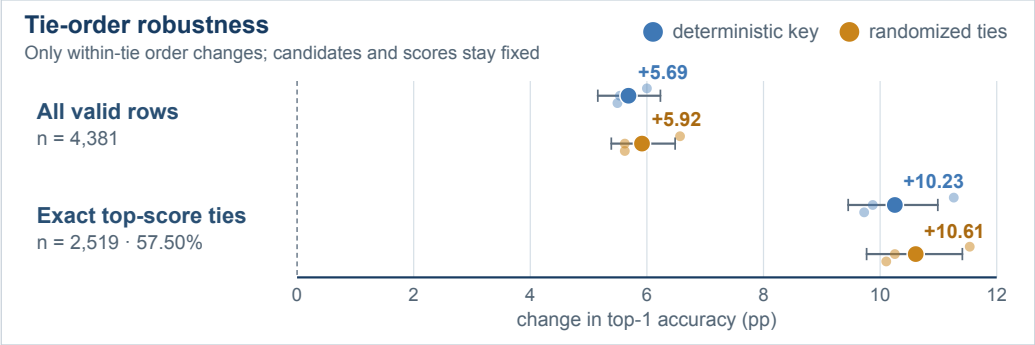


Figure A1: Tie handling on AndroidControl (4,381 rows; 2,519 exact top-score ties). Large dots show the three-checkpoint mean, small dots show individual seeds, and whiskers show ±1 SD. Randomization permutes only equal-score candidates.

Table A11: Label-free tie controls on the fixed AndroidControl candidate manifest. Lexical and BM25 columns use their respective deterministic keys; random tie breaking reports mean ± sample SD over 100 repetitions. The learned row averages the three frozen tree-position checkpoints.

Secondary order	Lexical all	Lexical ties	BM25 all	BM25 ties
Deterministic evaluator key	+0.000	+0.000	+0.000	+0.000
Random tie	−0.298 ± 0.296	−0.518 ± 0.515	–	–
Smallest area	+3.013	+5.240	+1.507	+5.523
Deepest node	+2.625	+4.565	–	–
Sibling order	−0.183	−0.318	–	–
Viewport center	−1.849	−3.216	–	–
DOM order	−2.191	−3.811	–	–
Largest area	−2.990	−5.200	–	–
Tree-position checkpoint	+5.691	+10.229	–	–

a position-prior explanation remains relevant. This audit does not include a train-fitted frequency lookup, so it does not claim to exhaust every position-only baseline.

We repeated the smallest-area control at the candidate-local BM25 operating point. BM25 has the same 1,195 exact top-score ties (27.277% of valid rows), and the smallest-area rule changes top-1 by +1.507 pp over all rows and +5.523 pp on those ties. Thus the area prior is not explained solely by the larger lexical tie rate; at the same time, its all-row change remains below the +3.622 pp gain of the learned tree-position residual above BM25. These are paired calibration comparisons under different ranking policies, not an additive decomposition or a causal estimate of an Area contribution. At this operating point, the learned all-row gain is 2.4 times the area heuristic's observed gain, which weakens an area-only explanation but does not identify a causal separation. On exact BM25 ties, the ordering reverses (+2.739 pp for the learned residual versus +5.523 pp for smallest area), so the learned advantage is concentrated beyond exact ties.

F.1 Exact-tie-only policy probe

Equation 12 defines a strict, label-free policy that activates the already-trained residual only when the maximum lexical score is shared by at least two candidates. It is not a new training run: the same three released checkpoints, fixed manifest, matcher, and bound are replayed under the policy. Table A12 reports the current audit replay. The gate covers 57.50% of valid rows and leaves the complementary unique-score rows exactly at the frozen lexical ordering by construction.

Table A12: Label-free exact-tie-only policy probe. The always-on column is the current canonical audit replay used for this probe; it is not substituted for the primary headline table.

Checkpoint	Valid rows	Exact ties	Always-on	Exact-tie-only
Seed 17	4,381	2,519 (57.50%)	+6.323	+6.391
Seed 23	4,381	2,519 (57.50%)	+5.410	+5.684
Seed 42	4,381	2,519 (57.50%)	+5.341	+5.570
Mean	4,381	2,519 (57.50%)	+5.691	+5.881

Table A13: Predeclared model ladder and shared configuration.

Model	Inputs / interaction	Training and reporting role
Lexical Stage–1	Frozen semantic score only	Reference; identical across seeds
Tree-position	11 input slots (10 unique quantities plus one legacy duplicate); $11 \to 16 \to 1$ MLP	Main bounded residual; $B = 0.10$
Pointwise-deep control	Interaction-ladder-matched local features; no cross-candidate mixing	Separates representation from generic capacity
Unrestricted attention	Candidate self-attention	Interaction control; seed-sensitive
Sparse attention	Fixed native-tree or geometry neighborhood	Relation control; not automatically transferable
Learned text-only Stage–1	Instruction and candidate text only	In-domain upstream pressure test; no topology input
Confidence-conditioned	Structural residual gated by normalized score entropy	Cross-setting replication diagnostic; gate is not strongly selective

The gate is best read as a conservative deployment option: when a frozen lexical upstream exposes an exact top-score ambiguity, the structural correction can be enabled; otherwise the upstream ranking is preserved. The mean difference from always-on is only +0.190 pp, so the probe is not evidence of a significant or universal superiority. Leaving unique-score rows unchanged avoids the aggregate non-exact-slice decrease by construction, but does not guarantee that every flip inside an exact-tie group is helpful. The always-on replay is generated by the same canonical source tree as the main table. The gate is reported as an auxiliary policy probe rather than promoted to a new headline model because it was introduced after checkpoint release and is not a new pre-registered test-set selection. Near ties with a positive score gap remain eligible under the general $2B$ bound but are intentionally outside this strict gate.

G Model Configurations and Training

Table A13 lists the predeclared model ladder and the role of each control in the evidence hierarchy.

Here MLP denotes a multilayer perceptron, and SD denotes sample standard deviation.

For the released $11 \to 16 \to 1$ architecture, the structural head has 209 trainable parameters, excluding optimizer state and checkpoint metadata. This is a module-size accounting, not an end-to-end latency measurement. The head is intentionally instruction-independent and should not be interpreted as an instruction-conditioned grounding model.

G.1 Exact structural feature specification

Table A14 makes the released feature constructor executable. All quantities are computed per candidate from the valid nodes in the same row and clipped to $[0, 1]$ after normalization. A candidate's direct parents and children are taken from the manifest's ID lists after invalid IDs are removed. The sibling index is zero-based and divided by $\max(n_{sib} - 1, 1)$. Depth is divided by 20; parent count by 4; child count and parent-sibling count by 20. Center coordinates are divided by the maximum valid node-center coordinate in the corresponding axis. Parent coordinate differences use the mean center of the direct parents and the same axis normalizer; if no valid parent exists, the candidate center is used, giving a zero difference. Area is divided by the product of the two maximum center coordinates. The legacy duplicate-child slot is exactly the child-count slot. No target, instruction, score, or evaluator output enters this constructor.

Unless otherwise stated, the main tree-position and non-attention controls use AdamW (Loshchilov & Hutter, 2019) with learning rate 0.002, weight decay 10^{-4}, at most 20 epochs, early stopping patience 4, listwise loss, gradient clipping at 1.0, and seeds 17, 23, and 42. The attention ladder uses the same candidate budget, matcher, residual bound, and tie rule, but a predeclared 12-epoch and patience-3 budget. Feature statistics are computed from training rows only. The confidence-conditioned variant uses temperature 0.10, gate penalty 0.001, and the same maximum residual.

For the interaction controls, each candidate uses the same 12-dimensional input: eleven standardized structural features and the frozen lexical score. The width-32 pointwise control has no cross-

Table A14: Executable definition of the released 11-slot structural input.

Slot	Definition after normalization and clipping
n^{par}	Number of valid direct parents divided by 4
n^{child}	Number of valid direct children divided by 20
$n^{par\text{-}sib}$	Sum of valid child counts over direct parents, divided by 20
d	Manifest depth divided by 20
r	Zero-based sibling index divided by $\max(n_{sib} - 1, 1)$
n^{dup}	Legacy alias of n^{child}; not independent evidence
x, y	Candidate center divided by the maximum valid center in each axis
$\Delta x, \Delta y$	Absolute difference from mean parent center divided by the corresponding axis maximum
a	Candidate box area divided by the product of the two axis maxima

Table A15: Package-disjoint AndroidControl top-1 accuracy and paired change. The canonical block uses the frozen lexical Stage–1; the BM25 block is a separate candidate-local upstream operating point; the final block contains protocol-matched classical ranking controls.

Model	Seed 17	Seed 23	Seed 42	Mean change
Canonical lexical operating point				
Lexical Stage–1	27.23	27.23	27.23	0.000 pp
Tree-position reranker	33.55	32.64	32.57	**+5.691 pp**
Candidate-local BM25 operating point				
BM25 Stage–1	32.55	32.55	32.55	0.000 pp
BM25 + tree-position residual	36.02	36.50	36.00	**+3.622 pp**
Interaction controls at the canonical lexical operating point				
Pointwise-deep (no interaction)	27.18	27.21	27.23	-0.023 pp
Unrestricted attention	33.32	27.23	33.37	+4.08 pp
One-hop sparse attention	27.23	27.50	27.37	+0.14 pp
Classical ranking controls				
Bounded linear fusion (listwise)	32.16	30.13	32.14	+4.246 pp
Bounded linear fusion (pairwise logistic)	30.45	31.11	31.25	+3.705 pp

candidate mixing; the attention controls add a four-head self-attention block and a $32 \to 64 \to 32$ feed-forward block. Each uses approximately 9K parameters. Capacity is matched within this interaction ladder, not to the 209-parameter headline head, so the shared-input comparison tests the interaction operator.

H Complete Seed-Level Results

H.1 AndroidControl Main Result

Table A15 gives the complete three-seed block for the canonical, BM25, interaction, and classical ranking operating points.

The package-equal-weight tree-position change is +1.82 pp with bootstrap 95% interval [+0.36,+3.27] pp. Leave-one-package-out means range from +1.57 to +2.19 pp. This package-equal estimand is consistent with the row-weighted direction but answers a different weighting question; its smaller value is specific to the package distribution rather than a universally more conservative estimator. The interval is obtained by averaging seed-level package means, sampling 78 packages with replacement for 2,000 repetitions using seed 53, and taking empirical 2.5th/97.5th percentiles. The same procedure and bootstrap seed are used for the current canonical replay. It conditions on the fixed checkpoints and does not include training randomness. The interaction controls do not meet the stronger stability gate. The BM25 (Robertson & Zaragoza, 2009) block uses the same train/dev/test manifests, candidate-local target matcher, deterministic tie key, and $B = 0.10$ bound as the canonical replay. BM25 is computed independently within each row's fixed candidate cohort and normalized to $[0, 1]$ only to make the residual scale comparable. For each seed, the tree-position residual head is independently fitted on the BM25 train/dev operating point; no lexical checkpoint or lexical train-only normalization statistics are reused. The paired gain is positive for all three seeds and averages $+3.622$ pp. Its all-row package-equal change is $+1.263$ pp with bootstrap 95% CI $[+0.260, +2.325]$ pp. This is an upstream operating-point control rather than a public external GUI baseline, so it is not pooled with the canonical lexical estimate.

Table A16: BM25 area-free sensitivity on the fixed AndroidControl test set with nearest-center matching. Values are top-1 accuracy (%) and paired changes; the 30.77% BM25 reference differs from the released-matcher result in Table 1.

Seed	BM25	Area-decoupled control	Δ pp	Helpful	Harmful
17	30.77	32.28	+1.507	155	89
23	30.77	31.50	+0.730	117	85
42	30.77	32.12	+1.347	145	86
Mean / total	30.77	31.96	**+1.195**	417	260

Table A17: BM25 ambiguity audit on the frozen BM25-conditioned reranker checkpoints. All values are paired top-1 changes over the BM25-only reference in percentage points.

Policy	Seed 17	Seed 23	Seed 42	Mean
Always-on residual	+3.470	+3.949	+3.447	+3.622
Exact-tie-only residual	+2.579	+2.945	+2.693	+2.739

Area-decoupled sensitivity control at BM25. We repeated the area-free matcher plus area-free retraining control at the candidate-local BM25 operating point. The target matcher selects the candidate whose center is nearest to the point, and the ten-slot schema removes area while retaining the other released inputs. Candidate rows, train/dev/test split, deterministic evaluator, and $B = 0.10$ remain fixed. All 4,381 test rows are valid. Table A16 gives the seed-level top-1 values and flip counts. Its 30.77% BM25 reference differs from the 32.55% value in the main operating-point table because the sensitivity control changes target assignment to nearest-center; the two values are not pooled.

The row-weighted bootstrap 95% interval is $[+0.563, +1.826]$ pp, while the package-equal estimate is $+1.844$ pp with a 78-package bootstrap 95% interval $[+0.893, +3.037]$ pp. Leave-one-package-out means range from $+1.579$ to $+2.012$ pp. Because both target assignment and representation change, this is boundary evidence rather than a causal decomposition of area or topology, and it is not combined with the released-matcher BM25 result above.

The BM25 ambiguity audit in Table A17 finds exact top-score ties in 1,195/4,381 rows (27.277%). Reusing the frozen checkpoints and applying the residual only on these ties gives the following row-weighted paired changes. The package-equal interval reported above belongs to the all-row always-on BM25 replay, not this exact-tie-only policy. This is a frozen, label-free policy audit; it does not refit the head or select a test-set operating point.

Unlike the lexical replay, the BM25 exact-tie-only policy is smaller than its always-on counterpart. The corresponding always-on changes in the three non-exact near-margin slices (gaps ≤ 0.20) are positive on average: $+2.576$, $+3.607$, and $+5.104$ pp for gaps in $[0, 0.05]$, $(0.05, 0.10]$, and $(0.10, 0.20]$, respectively; rows above 0.20 do not change. This observed concentration is consistent with the policy difference, but does not identify why BM25's near-ties differ from its exact ties and is not used as a causal explanation.

H.2 Package-level robustness audit

The package-cluster estimate is complemented by a direct sign and leave-group stress audit of the same 78 package-level deltas used for the primary bootstrap; no model is refit. Table A18 summarizes the audit.

Table A18: Package-level robustness audit for the canonical AndroidControl checkpoint set.

Audit	Result
Package count	78
Positive / unchanged / negative packages	33 / 25 / 20
Package mean and median	$+1.824$ pp / 0.000 pp
Package range	-25.000 to $+22.222$ pp
Two-sided sign test	$p = 0.098$
Remove five largest packages	$+2.954$ pp mean; 2,291 rows remain

Table A19: Residual sign and winner-change audit for the frozen canonical checkpoints. Near-bound is the fraction of all candidate residuals satisfying $|\delta_i| \geq 0.99B$. Helpful and harmful counts are target-aware top-1 flips.

Seed	Positive	Negative	Near-bound	Changed	Helpful	Harmful
17	31.62%	68.38%	87.49%	1,732	375	98
23	33.01%	66.99%	86.52%	1,710	342	105
42	30.52%	69.48%	86.33%	1,767	337	103

Thirty-three packages improve, 25 are unchanged, and 20 decrease; the exact two-sided package sign test gives $p = 0.098$. This is not a claim that every package benefits. It makes heterogeneity explicit and separates a directional row-level effect from uniform package-level success.

As a non-outcome-selected stress test, removing the five packages with the largest row counts leaves 2,291 rows and mean changes of $+2.663$, $+2.881$, and $+3.317$ pp for seeds 17, 23, and 42 (mean $+2.954$ pp). This check is descriptive: it is not a new estimand, and it does not justify removing any package from the primary result.

Seed-23 unrestricted-attention audit. The finite checkpoint stopped at epoch 1 after development changes of +3.006, +1.963, +0.337, and +0.061 pp at epochs 1–4. On test it changed zero top-1 rows and had mean and maximum absolute residuals of 0.1000. This is a valid seed-specific zero, retained as evidence of seed sensitivity; the available summary lacks residual signs, so it cannot establish one-sign corrections or explain optimization dynamics.

H.3 Residual sign and saturation audit

The three frozen checkpoints have both positive and negative residuals, while 86.3–87.5% of candidates lie within $0.01B$ of the output cap. This shows frequent near-cap outputs at the current operating point, not a bound violation or universal saturation; it also rules out a same-sign collapse.

Table A19 gives the seed-level residual signs, saturation rates, and target-aware winner changes.

'Changed' includes target-neutral winner swaps, so it is not additive with helpful or harmful flips and is not a new accuracy estimator.

H.4 Learned Text-Only Upstream and Confidence-Conditioned Controls

Table A20 summarizes the learned-upstream and confidence-conditioned controls on the shared candidate protocol.

Table A20: Robustness controls on the same fixed candidate protocol.

Condition	Seed 17	Seed 23	Seed 42	Mean
Learned text-only upstream top-1	39.260	39.831	39.215	39.435
Tree position above learned text-only upstream	+0.342	+0.411	+0.845	+0.533 pp
Topology shuffle above learned text-only upstream	+0.114	-0.183	+0.160	+0.030 pp
Confidence gate, AndroidControl	+5.729	+6.049	+5.821	+5.866 pp
Static control, AndroidControl	+6.094	+5.592	+6.254	+5.980 pp
Confidence gate, GUIrilla fitted diagnostic	+4.321	+2.881	+2.675	+3.292 pp
Static control, GUIrilla fitted diagnostic	+2.263	+2.984	+3.086	+2.778 pp

The confidence gate is not numerically identical to the static control. On AndroidControl it is 0.114 pp lower (+5.866 versus +5.980 pp), whereas on the GUIrilla diagnostic it is 0.514 pp higher (+3.292 versus +2.778 pp). This is a finite training difference rather than evidence of numerical instability: the gate adds a penalty to the training objective, so its development-selected parameters need not match the static head even when the final mean gate is near one. We do not tune the penalty post hoc; the gate is retained as a non-selective appendix diagnostic.

Here, the static control is the ordinary $11 \to 16 \to 1$ PositionGate fitted in the same run without a confidence multiplier. The gated control uses that head with a multiplier driven only by fixed upstream-score uncertainty; the rows are paired controls, not test-selected alternatives.

Table A21: Learned text-only upstream interaction ladder on the same fixed AndroidControl candidate protocol. Values are paired changes in percentage points over the frozen text-only Stage–1.

Condition	Seed 17	Seed 23	Seed 42	Mean
Learned text-only upstream top–1	39.260	39.831	39.215	39.435
Pointwise-deep	+0.548	+0.639	+0.616	+0.601
Unrestricted attention	+0.297	+0.479	+0.365	+0.380
Sparse tree attention	+1.118	0.000	+1.324	+0.814
Pointwise-deep shuffle	0.000	+0.114	-0.046	+0.023
Unrestricted shuffle	+0.205	+0.046	+0.274	+0.175
Sparse tree shuffle	+0.023	-0.068	+0.023	-0.008

Table A22: Classic structural controls on package-disjoint AndroidControl. Values are paired changes in percentage points over the frozen lexical Stage–1.

Control	Seed 17	Seed 23	Seed 42	Mean
Bounded linear fusion (listwise)	+4.930	+2.899	+4.908	+4.246
Bounded linear fusion (pairwise logistic)	+3.218	+3.880	+4.017	+3.705
Tree-position reranker (reference)	+6.323	+5.410	+5.341	**+5.691**

The learned text-only upstream is a compact in-domain hashed-token EmbeddingBag–MLP trained with a listwise objective on instruction and candidate text; it uses no visual, geometry, or tree inputs. Its +0.533 pp structural change is smaller than the +5.69 pp lexical-protocol result because the two are distinct upstream operating points. The confidence gate adds +0.514 pp over its same-run static control on the fitted GUIrilla diagnostic; its mean gate is 0.999 on AndroidControl and 0.983 on GUIrilla. These values show little selective suppression in this experiment.

This text-only stress test does not estimate performance with multimodal upstreams, which would change visual preprocessing, score calibration, and possibly candidate coverage. Such a comparison requires a separately released, protocol-matched scored manifest.

The finer interaction ladder above the learned text-only upstream is reported in Table A21.

The sparse tree attention variant gives the largest mean change, but only two of three seeds improve. The near-zero sparse shuffle mean is consistent with a relation-sensitive interpretation, while the residual unrestricted-shuffle signal shows that topology is not the only source of the correction. We retain this ladder as a robustness diagnostic rather than a new headline method.

H.5 Classic Structural Controls

We add two conventional learning-to-rank controls under exactly the same fixed-candidate protocol as the main AndroidControl result. Both use the same released 11-slot structural schema (ten unique quantities plus one legacy duplicate), frozen lexical score, target matcher, split, tie rule, and residual bound $B = 0.10$. The first is a bounded linear residual trained with a listwise objective; the second is a bounded linear residual trained with a pairwise logistic objective. Neither uses candidate interaction or target-derived features.

Table A22 reports the resulting protocol-matched linear controls beside the tree-position reference.

Both linear controls improve over lexical Stage–1, but remain below the bounded tree-position head under the shared protocol. This comparison covers these two fusions and the released feature schema; it does not rank all possible structural feature sets.

H.6 Duplicate-feature sensitivity

The released checkpoint uses eleven input slots because the historical manifest contains both `child_count` and its duplicate-child alias. We retrain the same bounded head with the alias removed, recompute standardization statistics from the training split, and keep the candidate set, target matcher, evaluator, bound, and seeds unchanged.

The first sensitivity check, removing only the aliased input slot, is reported in Table A23.

Table A23: Deduplicated ten-slot sensitivity on package-disjoint AndroidControl. Values are row-level top-1 accuracy in percent and paired change over the frozen lexical Stage–1.

Seed	Lexical Stage–1	Deduplicated head	Change
17	27.231	33.577	+6.346 pp
23	27.231	34.056	+6.825 pp
42	27.231	32.892	+5.661 pp
Mean	27.231	33.508	**+6.277 pp**

Table A24: Joint removal of area, direct child count, and the legacy duplicate child-count alias. Values are row-level top-1 accuracy in percent and paired change over the frozen lexical Stage–1.

Seed	Lexical Stage–1	Eight-slot head	Change
17	27.231	24.903	-2.328 pp
23	27.231	25.177	-2.054 pp
42	27.231	25.656	-1.575 pp
Mean	27.231	25.245	**-1.986 pp**

The three-seed row-weighted mean is +6.277 pp; its package-equal change is +2.498 pp (78-package bootstrap 95% CI $[+0.998, +4.121]$ pp), with leave-one-package-out means from +2.097 to +2.675 pp. This interval conditions on the fixed checkpoints and does not include training randomness. The row- weighted point estimate is 0.586 pp above the released eleven-slot mean of +5.691 pp, but this follow-up remains a sensitivity result rather than a replacement primary model. It shows that the observed gain does not depend on the aliased slot; it does not establish that every remaining feature is necessary.

To probe the remaining feature-coupling concern, we additionally retrained the same head after jointly removing area, direct child count, and the legacy duplicate-child alias. The resulting eight-slot schema retains parent count, parent-sibling count, depth, sibling order, center coordinates, and parent-relative coordinate differences. All other protocol choices remain unchanged. The joint ablation is negative on all three seeds, but its package-level interval crosses zero; it is therefore a boundary sensitivity result rather than a causal attribution to any one removed slot.

The joint removal result is shown in Table A24.

The package-equal change is -1.379 pp with bootstrap 95% CI $[-3.399, +0.593]$ pp; leave-one-package-out means range from -1.743 to -0.964 pp. The result is consistent with the full schema benefiting from information in at least one of the removed feature families, but it does not identify an individual causal feature. In particular, parent-sibling count is retained, and the joint removal changes both representation and model input dimension. We therefore use this result to qualify, rather than replace, the positive full and deduplicated-schema findings.

I Cross-platform Replication and Boundary Audits

I.1 GUIrilla-Task

Figure A2 visualizes the three-seed identity and shuffle outcomes, which are also reported in Table 4 in the main text; this appendix records the control construction and its interpretation.

The 176 held-out application bundle IDs are the grouping units for the equal-weight audit. Across the fixed checkpoints, 67 apps improve, 94 are unchanged, and 15 decline. The app-equal mean change is +4.686 pp (95% app-bootstrap interval $[+2.108, +7.006]$ pp) on the same 972 click rows. The positive interval persists under equal app weighting, although most apps are unchanged; it does not establish zero-shot transfer.

The shuffle keeps node IDs, text, boxes, candidate flags, target IDs, and row order fixed, and permutes parent/child links and depth among non-root nodes. The depth/sibling-only row instead permutes only those two feature columns. Neither transformation consults target metadata. The joint shuffle attenuates the identity gain by 63.6% but does not remove it, so the result supports structural sensitivity without identifying graph connectivity as a causal source of every corrected row.

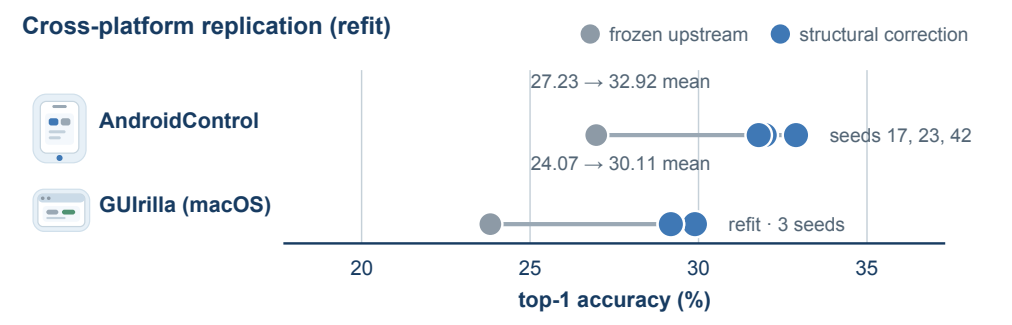


Figure A2: App-disjoint GUIrilla replication after refitting. The main text reports the aggregate result; this figure shows the paired seed replication and the structural shuffle control.

I.2 MUG–RICO

The retained interaction-control boundary is reported in Table A25; it is not a refit of the headline tree-position head.

Table A25: App-disjoint MUG–RICO boundary. The lexical baseline is 16.523% for all seeds and every reported interaction-control mode remains unchanged. A tree-position refit is not included in this boundary artifact.

Model	Seed 17	Seed 23	Seed 42	Mean change
Pointwise-deep control (9K; no interaction)	0.000	0.000	0.000	0.000 pp
Unrestricted attention	0.000	0.000	0.000	0.000 pp
Sparse attention	0.000	0.000	0.000	0.000 pp

The table reports the interaction-control boundary rather than a complete replication of the headline tree-position head; no tree-position MUG–RICO number is implied. The join, target coverage, app-disjoint split, row counts, finite checkpoints, and learned-score evaluator all pass audit. The app-equal bootstrap interval is [0.0,0.0] for each mode. The most defensible interpretation is an unseen-app cross-schema boundary: static RICO hierarchy is available, but the reported top-1 accuracy remains unchanged at this operating point. Because a zero accuracy delta does not by itself measure winner changes, this result does not establish that the residual never changes an individual ordering. It is also not evidence that hierarchy is universally useless.

The complementary winner-scope audit checks this distinction directly over the nine mode-by-seed checkpoints. All 9,659 rows retain the same 16.523% top-1 accuracy; only four winners change in total, all under pointwise-deep seed 23, and none is a helpful or harmful accuracy flip. Exact lexical ties cover 5,848 rows (60.545%). Thus the MUG–RICO boundary is a genuine no-measurable-gain result for this candidate/schema configuration, rather than an artifact of an uninspected tie count; it still does not imply that every individual score ordering is unchanged. The machine-readable audit is included with the reproducibility scripts.

I.3 AMEX

Table A26 records the flat-annotation geometry boundary and its seed-level interaction controls.

Table A26: Package-disjoint AMEX boundary. The sparse relation is a geometry proxy because the released element annotations are flat.

Model	Seed 17	Seed 23	Seed 42	Mean change
Pointwise-deep control (9K; no interaction)	-0.051	0.000	0.000	-0.017 pp
Unrestricted attention	+2.968	+1.228	+0.921	+1.706 pp
Sparse geometry attention	+0.051	0.000	0.000	+0.017 pp

The -0.017 pp entry belongs to the pointwise-deep control; the sparse geometry-attention control is $+0.017$ pp. Both are effectively null under the reported package uncertainty, so neither should be read as stable replication of the native accessibility-tree result.

The unrestricted package-cluster intervals are [+1.296,+5.068], [-0.208,+3.453], and [-0.499,+2.203] pp; sparse intervals include zero in every seed. Because AMEX has flat annotations, it tests a geometry proxy rather than native tree position. The fixed $k = 4$ graph may expose local geometry to unrestricted attention, but intervals crossing zero leave the isolated positive mean inconclusive.

I.4 External visual reference

We evaluate released UGround (Gou et al., 2025) on an image-bearing bridge manifest with the canonical 4,381 row IDs. Given only a screenshot and instruction, it predicts a point that we project to the smallest-area containing candidate; if none contains the point, we use the globally nearest box center, with candidate ID as the tie key. This differs from the area-free target-matcher audit, which selects among candidates containing the recorded target point.

Table A27 reports the resulting projected external reference without treating it as a matched structural reranker.

Table A27: UGround point predictions projected to fixed AndroidControl candidates. This reference uses screenshots rather than structural-reranker inputs.

Metric	UGround
Test states	4,381
Candidate top-1	80.347% (3,520)
Valid predictions	100.000% (4,381)
Inside candidate box	99.863% (4,375)
Nearest-center fallback	0.137% (6)

Table A28 gives the corresponding qualitative boundary between protocol-matched controls and external references.

Table A28: Inputs and comparison stage for external methods.

Method	Input	Candidate stage	Role
Frozen lexical Stage–1	Instruction and candidate text	Fixed candidates	Matched reference
TopoRerank	Frozen score and tree features	Fixed candidates	Bounded correction
Pairwise logistic (Burges et al., 2005)	Same tree features	Fixed candidates	Objective control
UGround (Gou et al., 2025)	Screenshot and instruction	Point projected to candidates	External visual reference
SeeClick / CogAgent / GUI-Actor (Cheng et al., 2024; Hong et al., 2024; Wu et al., 2025)	Screenshot	Method-specific	Literature context
RankGround (Fan et al., 2026)	Visual crops and instruction	Crop selection	Related visual method

UGround is the only external checkpoint projected to these candidates; its 80.347% top-1 is calibration, not a matched-reranker result, because it uses screenshots and predicts before projection. The bridge adds image paths and audits all 4,381 row IDs and targets, but is not byte-identical to the canonical manifest. Pairwise logistic is a matched objective reimplementation; other visual methods are context only. Direct comparison requires a public checkpoint with the same candidates and target contract. This reference complements accessibility-only services constrained by privacy, bandwidth, or runtime; it does not replace visual grounding. Predictions, hashes, and projection metadata are included in the artifact.

J MECHANISM AUDITS

Table A29 summarizes the fixed-checkpoint perturbations.

Table A29: Label-free frozen-checkpoint and control audits.

Setting / perturbation	Mean change	Rows	Interpretation
AndroidControl identity	+5.691 pp	4,381	Reproduces the main checkpoint result
Within-query structural shuffle	-3.203 pp	4,381	Structural input matters under fixed checkpoint
All-position shuffle	-5.813 pp	4,381	Position block is important
Geometry-only shuffle	+0.487 pp	4,381	Geometry alone is not the whole effect
GUIrilla original topology	+6.036 pp	972	External identity condition
GUIrilla topology shuffle	+2.195 pp	972	Gain attenuates by 63.6%, but not to zero
GUIrilla depth/sibling shuffle	+4.835 pp	972	Seeds +4.733, +5.350, +4.424; 80.1% of identity gain remains
GUIrilla count-descriptor shuffle	+5.075 pp	972	Counts permuted; other feature groups retained
Learned text-only upstream topology shuffle	+0.030 pp	4,381	Structural effect above learned text-only upstream is small

The converted GUIrilla manifest is unchanged. The joint shuffle permutes parent/child links and depth among non-root nodes; the narrow shuffle permutes only depth and sibling order. Both preserve node IDs, text, boxes, target IDs, candidate flags, and row order without target metadata. These are sensitivity tests, not causal proofs, since positional signal may survive. Of 972 rows, 664 are target-aware eligible (430/738 lexical errors); across seeds, helpful flips are 88/82/83 and harmful flips are 25/26/27. These counts bound correction scope, not live-agent performance.

At $B = 0.10$, at least one candidate reaches $0.99B$ on 4,381/4,381 AndroidControl rows and 971/972 GUIrilla rows. This maximum-over-candidates measure is not per-candidate saturation or a bound violation; see Table A30.

The identity-to-depth/sibling shuffle difference is 1.201 pp (19.9% of the identity gain), not a connectivity contribution: the narrow shuffle preserves links and perturbs depth/order, while the joint shuffle changes links and derived depth. These controls are therefore not factorial.

The count-descriptor shuffle permutes only count-derived inputs; its seed gains are +5.453, +4.835, and +4.938 pp (mean +5.075 pp). Because other feature groups remain, this is a sensitivity contrast, not a causal graph ablation.

J.1 RESIDUAL-BOUND SENSITIVITY

We replay the three frozen checkpoints on the same 4,381 AndroidControl test rows with varied residual bounds; no checkpoint is retrained or selected on the test split.

Table A30: Top-1 change (pp) over frozen lexical Stage–1 across residual bounds; entries show three seeds and mean (sample SD).

Bound B	Seed 17	Seed 23	Seed 42	Mean (SD)
0.05	+6.231	+5.455	+5.387	+5.691 (0.469)
0.10	+6.323	+5.410	+5.341	+5.691 (0.548)
0.20	+6.231	+4.748	+4.953	+5.311 (0.804)
0.50	+5.889	+4.565	+4.816	+5.090 (0.703)

The frozen-checkpoint ranking remains positive across the tested rescalings, while larger bounds are less stable. This audit does not test retraining or bound calibration and is therefore not evidence that any particular bound is optimal.

J.2 Feature-family Retraining Ablation

To isolate feature families, we retrain the same $11 \to 16 \to 1$ head after masking unused standardized slots. Manifest, matcher, split, bound, optimizer, stopping rule, and seeds remain fixed. The full refit uses all eleven slots (ten unique); tree-only uses six hierarchy/count slots, and geometry-only uses five center, relative-position, and area slots.

Table A31 reports the corresponding retraining ablation.

Table A31: Feature-family retraining ablation on package-disjoint AndroidControl. Values are paired changes in percentage points over the frozen lexical Stage–1.

Feature family	Seed 17	Seed 23	Seed 42	Mean
All slots (refit)	+6.414	+5.410	+5.181	**+5.668**
Tree-only	+4.428	+4.725	+4.474	**+4.542**
Geometry-only	+0.616	-0.160	+0.502	+0.320

Tree-only retains +4.542 pp of the +5.668 pp full-refit gain; geometry-only yields +0.320 pp and changes sign across seeds. Package-equal 95% CIs are $[-0.002, +3.161]$, $[-0.069, +2.419]$, and $[-2.575, +0.804]$ pp, all crossing zero. These are sensitivities, not primary significance claims; separate refits also explain the difference from the released +5.691 pp result.

J.3 Single-feature permutation audit

To refine the group controls, each structural or geometric feature was permuted across candidates within each query while preserving candidate text, boxes, target IDs, row order, and the evaluator. The checkpoint and bound were fixed. The mean changes below are therefore stress-test sensitivities, not feature deletion or causal attributions.

Child count and its legacy duplicate are exact aliases. Permuting either slot independently breaks that equality, so those two entries are schema stress tests and not evidence that the aliased quantities provide independent information.

Table A32 reports the resulting within-query permutation sensitivities.

Table A32: Mean change after within-query permutation of one feature.

Feature	Mean change	Feature	Mean change
Parent count	+5.235 pp	Child count (legacy alias also used)	+1.910 pp
Parent sibling count	+5.204 pp	Depth	+5.136 pp
Sibling order	+5.303 pp	Duplicate-child count	+1.826 pp
Center x	+4.999 pp	Center y	+4.946 pp
Parent Δx	+5.083 pp	Parent Δy	+5.296 pp
Area	+2.336 pp		

Single-feature permutations remain positive on average, whereas structural-block and all-position shuffles are negative. Together, these controls indicate a distributed sensitivity pattern, not a causal attribution to one feature.