WHERE DOES A PREDICTIVE GUI VERIFIER LOSE TO A DISCRIMINATIVE ONE? A CONTROLLED DIAGNOSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Step-level verification asks whether an observed GUI screen is compatible with the action claimed to have produced it, the goal-free check that retry and recovery rely on. A predictive verifier forecasts the successor with a latent world model and compares the forecast with the screen; a discriminative verifier reads the action and both screens jointly. We compare the two at matched data, negatives, backbone and training budget on a pre-registered benchmark of recorded transitions. The predictive verifier is worse by **6.53 AUC points**, by 6.30 on a test set built and scored once after every model was frozen, and by a wider margin at low false-positive rates. The gap is not a labelling artefact: two independent annotators find a minority of constructed negatives still compatible, and on the negatives both confirm the gap is essentially unchanged. A ladder of controlled variants between the two verifiers locates it. The four losses that align the forecast with the observed screen move aggregate AUC by only 0.19 points. Comparing the screens after encoding them apart costs 4.93; in-distribution, neither keeping every patch nor matching the comparison head recovers more than a point of it, which leaves fusion timing. Gains that narrow the gap do not survive a change of distribution, and both verifiers refuse genuine interface anomalies more often than ordinary transitions.

1 INTRODUCTION

GUI agents act between screenshots, and an action can fail without raising an error: a tap lands beside its target, a page is still loading, a dialog swallows the input. Whether the observed successor s_{t+1} is compatible with the action a claimed in state s_t is therefore the check that retry, recovery and process-reward methods consume. We call it **step-level effect verification** (L1). It is goal-free: it asks neither whether the transition advanced a task (L2) nor whether the task succeeded (L3), the questions that existing GUI critics and trajectory evaluators answer with the goal as input (Qian, 2026; Hu et al., 2025; Xiong et al., 2025; Chen et al., 2025a; Chae et al., 2025b; Chen et al., 2025b). A progress judgement conditioned on a transition that did not occur is unreliable however well it judges progress.

Two designs answer L1 (Figure 1). A **predictive** verifier forecasts what the action should produce and compares the forecast with what appeared: VeriGUI does so with a linguistic expectation of the next screen (Zhang et al., 2026b), and latent world models of the JEPA family (Assran et al., 2023; Zhou et al., 2025; Assran et al., 2025) offer the same predict-then-compare pattern in representation space. A **discriminative** verifier reads (s_t, a, s_{t+1}) jointly (Qian, 2026). The predictive design is attractive because the same model can also score an action before its successor exists, but what it costs as a post-hoc verifier has not been measured. Comparisons across papers change backbone, data and negatives together, and the constructed negatives that GUI verifiers are trained and tested on are rarely checked for whether they are negative at all.

We price the choice under a pre-registered evaluation that fixes backbone family, data, negative construction, sample budget and adapter rank. On recorded GUI transitions the predictive verifier is worse by **6.53 AUC points** (95% CI $[+6.08, +6.99]$) and by **43.7 points** of TPR at 1% FPR. To locate the gap we build a ladder of variants that walks from one verifier to the other one design

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.

field at a time (Table 2). **Our central finding is that the cost lies in the late comparison that predict-then-compare builds in, not in the losses that train the forecast.** Both verifiers see s_{t+1}, so the 6.53 points price a post-hoc design, not the ability to score before execution, which a direct state–action scorer also provides (§6).

Contributions.

- **A matched comparison on an audited benchmark** (§3–4). The gap holds across three seeds per verifier, every negative tier, an action-string shortcut control, a one-shot test built after every model was frozen ($+6.30$), and blind relabelling by two independent annotators ($+6.60$ on the negatives both confirm; Table 1).
- **A ladder of controlled variants that locates the gap** (§5). Switching off the four alignment losses moves aggregate AUC by 0.19 points. Encoding the frames apart costs 4.93: keeping both patch grids buys back 0.79 in-distribution, and matching the comparison head moves AUC by less than a point on each of four evaluation sets, which leaves fusion timing ($+4.23$).
- **Boundaries on what the benchmark and the gap license** (§5–7). None of seven replacements for the frozen target closes the gap; interventions that narrow it do not survive a change of corpus, tier or candidate source; and both verifiers refuse genuine interface anomalies, consistent with scores that encode typicality rather than compatibility alone.

Scope. This is a controlled diagnosis of two designs on one backbone family, not a claim about prediction in general. The predictive objective is one latent, JEPA-style family (Assran et al., 2023; Bardes et al., 2022); Disc is a budget-matched control, not a deployed critic; Pred-verify-only removes the alignment losses but still trains the predictor, so nothing here shows that prediction is unnecessary; and everything is offline (§8).

2 Related Work

GUI verification and process evaluation. Existing GUI critics mainly judge whether an action advances a stated goal (L2) (Hu et al., 2025; Xiong et al., 2025; Qian, 2026; Chen et al., 2025a; Chae et al., 2025b); trajectory benchmarks such as SPA-Bench judge task success (L3) (Chen et al., 2025b), and recovery benchmarks such as ROTS test escape from an error state, downstream of the L1 decision (Bu et al., 2026). The levels describe what a judgement is conditioned on, not a partition of systems: VisCritic's action-aware head jointly scores action success, progress and error type with the goal as input (Qian, 2026). Closest to our question, VeriGUI checks each step inside a goal-conditioned loop against a *linguistic* expectation of the next screen and returns NO_CHANGE when the screen does not move (Zhang et al., 2026b). Fine-grained interface-difference models (Zhang et al., 2026a) supply a capability verification needs without deciding whether a transition is valid. **To our knowledge no prior work compares predictive and discriminative L1 verifiers at matched data, negatives, model family and budget.**

Predictive representations and world models. Latent predictive methods keep the target non-degenerate through a frozen encoder or an EMA target (Assran et al., 2025; Garrido et al., 2024; Zhou et al., 2025; Grill et al., 2020; Tarvainen & Valpola, 2017), or end to end under an embedding regulariser (Maes et al., 2026); we treat that choice as an experimental variable. UI-JEPA applies the family to interface videos to read user intent (Fu et al., 2024), without conditioning on actions. Action-conditioned GUI world models predict the next observation as language (Gu et al., 2025; Chae et al., 2025a; Yu et al., 2026), as pixels or sketches (Luo et al., 2025; Cao et al., 2026; Guan et al., 2026), as renderable code (Zheng et al., 2026), or as an image-to-image map for policy pretraining (Liu et al., 2026); value-environment models score $Q(s, a)$ directly (Zheng et al., 2025). These models are built for planning and pretraining. We study a latent predict-then-compare *verifier*, the representation-space counterpart of VeriGUI's linguistic check, and ask what its comparison costs.

Early and late comparison. In retrieval and sentence scoring, cross-encoders that read both inputs jointly outrank bi-encoders that embed them separately, which are kept for their pre-computable

indices (Nogueira & Cho, 2019; Reimers & Gurevych, 2019; Humeau et al., 2020); late-interaction models keep token-level representations to recover part of the difference (Khattab & Zaharia, 2020). A predict-then-compare verifier encodes separately for a different reason: its successor does not exist when the forecast is made. We measure what that late comparison costs where it is not a speed trade-off, and find that keeping token grids recovers little of it in-distribution (§5).

Benchmark validity. Constructed evaluation data can reward shortcuts (Gururangan et al., 2018; Geirhos et al., 2020), and label errors in test sets can reorder models (Northcutt et al., 2021). Constructed GUI negatives carry both risks: a perturbed action string can be guessable on its own, and a displaced click can land on the same control. We neutralise the action-string prior directly, have constructed negatives relabelled blind by independent annotators, and report the headline under adversarial relabelling and on a one-shot test.

3 Task, Benchmark and Controlled Variants

Two screenshots and an action label support only the weakest of three questions that are easily conflated: **compatibility** (is the successor consistent with the claimed action?), **intent** (did it achieve what the policy meant?), and **interventional causation** (would the successor have differed without the action?). **We score compatibility throughout.** The constructed labels nevertheless mix observational compatibility with the effect expected under successful execution: a no-op negative asserts that the action should have changed the screen, which holds only if it executed, and no label measures execution success itself. A substituted action is also occasionally one the environment would have answered the same way, so a constructed negative can fail to be negative; the audits below measure how often.

The benchmark is built from two mobile GUI corpora (Li et al., 2024; Lu et al., 2025). Positives are recorded transitions and negatives are generated from them in six tiers: cross-episode, same-episode offset, no-op, action mismatch, action-type mismatch and action near-miss (Figure 2; Appendix A). This tests compatibility against a synthetic counterfactual rather than failure detection, a gap §7 revisits with transitions that genuinely occurred. The judgement set joins `unseen_task` (instructions held out within seen applications) and `unseen_app` (whole applications held out): 2,898 source triplets and 15,465 samples after audit exclusions.

Two routes, and a ladder between them. The **discriminative** route (Disc) packs the action and both frames into one forward pass and reads a logit from an MLP head (Figure 1c); it is the simplest verifier in which all three inputs attend to each other. The **predictive** route (Pred) is a latent world model in the JEPA sense: a predictor $f_\theta(s_t, a)$ produces a patch-grid forecast $\hat{z}$ of the successor, a target encoder $g_\phi(s_{t+1})$ produces z, and verification reads fixed functions of the per-patch cosines between them (Figure 1b). Four alignment losses train the forecast: per-patch cosine regression, a pooled contrastive term, an action-contrastive term and a no-op term. The target encoder is frozen, as in latent world models that plan against a pretrained encoder (Zhou et al., 2025), and §5 tests seven alternatives. Because Pred emits a cosine profile rather than a score, it selects among eight fixed readouts on a held-back split (Appendix J).

The two routes differ in several fields at once—when the frames interact, whether a forecast intervenes, whether the comparison is fixed or learned, and whether a target encoder exists—so their gap alone cannot say which field costs what. We therefore walk from Disc to Pred one field at a time (Table 2). **Disc-grid** reads Disc's joint pass with a cross-attention comparator over the two frames' patch grids in place of the MLP head, and differs from Figure 1c only in that head, so the figure does not draw it; **Disc-late-grid** uses that comparator but encodes the frames in separate passes; **Disc-late** also pools each frame to a vector before the two meet; **Pred+readout** is the predictive route with a comparator of the same family reading $\hat{z}$ against z; and Pred replaces that comparator with the fixed readout. Side branches move one further field: **Pred-verify-only** switches off the four alignment losses behind Pred+readout, **Pred-EMA** replaces the frozen target with an EMA copy, **Pred+near-miss** changes the training negatives, and 4B/8B backbones change scale (Table 21).

Models, budget and statistics. All primary models are LoRA adapters (Hu et al., 2022) on Qwen3-VL (Bai et al., 2025), 2B by default and 4B/8B for scale checks, with a shared budget of 1.28M samples (80,000 steps at effective batch 16; 160,000 at 8 for 4B and 8B), adapter rank

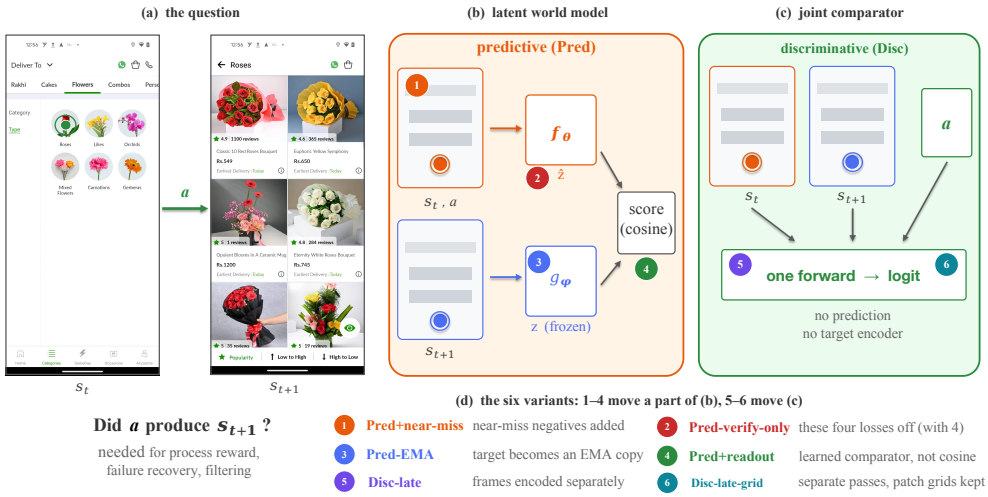

Figure 1: Task and the two routes. **(a)** The motivating question on a real benchmark step (the circle marks the recorded click); what the labels support, and what we score, is the weaker compatibility form (§3), the goal-free L1 primitive that recovery and process reward consume. **(b)** Pred compares a prediction $\hat{z}$ with a *frozen* encoding of the observed successor. **(c)** Disc reads the pair and the action in one forward pass and has no target encoder. **(d)** The six variants. Badges ①–④ sit on the part of (b) each one moves; ② sits on $\hat{z}$ because the alignment losses constrain it against z. Badges ⑤–⑥ move where the frames meet in (c): Disc-late encodes them separately and pools each, Disc-late-grid keeps both patch grids. Pred-verify-only is built on Pred+readout (④), and Disc-late-grid on Disc-late. Disc-grid, which replaces only the head of (c), is not drawn (§3).

16 and learning rate 1×10^{-4} (Appendix J). Training uses Android Control and GUI Odyssey; CAGUI, GUI-360 and GUI-Robust are evaluation-only. Each source triplet contributes one positive and several negatives, so every interval is a paired bootstrap over source triplets ($B = 2{,}000$, seed 0); episode- and row-level resampling change no conclusion (Table 13). Those intervals resample data, not training runs, so both verifiers were also trained at three seeds, giving $+6.53$, $+6.48$ and $+6.45$ points (SD 0.040; single-model SD 0.06–0.09). Three further variants trained twice differ by 0.12, 0.32 and 0.35: below the effects we interpret, but not below the 0.19 we report as an equality, which is why that contrast is a two-seed mean.

Are the constructed negatives negative? The pre-registered *pixel* audit excludes 398 of 23,899 near-identical rows; a broader post hoc scope flags 933, and the results hold across both (Table 14). Pixels cannot tell two clicks on the same control apart, so four blinded passes labelled the same 210 rows, 30 per stratum, seeing only the two frames and the action string: the pre-registered *author* audit, the strictest of the four, a language model, and two independent human annotators from outside the project. They find 16.1%, 13.3%, 11.1% **and** 7.2% **of constructed negatives still compatible** (human annotators' agreement 87%, κ=.69; author and first annotator κ=.86), with the same ordering of tiers in every pass: highest on near-miss (23–33%), whose generator never consults element bounds, then no-op, and zero on cross-episode (Appendix G). The gap stays positive under every relabelling in Table 1: with the author's sampling error propagated and the errors placed adversarially it is 3.73 $[2.96, 4.43]$, and 1.77 in the harshest case, every author rate at its Wilson upper bound. 6.53 is the operational number on the pixel-audited set, not a constant under relabelling. Judgement-set identifiers are excluded from training by a fail-closed whitelist, every experiment is pre-registered before its results exist, and an earlier leak of the benchmark into training worked *against* our conclusion (Appendix I).

4 The Cost at Equal Budget

Does the predictive route cost anything post hoc? Pred trails Disc by 6.53 AUC points (95% CI $[+6.08, +6.99]$; Table 2). All six tiers favour Disc, from 0.89 points on cross-episode to

Table 1: What the label audit leaves of the gap (Disc − Pred, AUC points). Four blinded passes labelled the same 210 rows (30 per stratum): the author (pre-registered), a language model, and two independent human annotators. Removals act on the full judgement set using each pass's per-tier error rates e_t; "adversarial" removes the rows most favourable to our conclusion, and the posterior row draws e_t from $\mathrm{Beta}(k+\frac{1}{2}, n-k+\frac{1}{2})$. The last rows score only negatives the named passes all judge incompatible, against all positives. Everything except the author's point-estimate removal and the tier drops is post hoc.

	author	model	human 1	human 2
six-tier error rate e	16.1%	13.3%	11.1%	7.2%
remove e_t adversarially	+3.86	+4.24	+4.46	+5.54
e_t from posterior	+3.73	+4.13	+4.35	+5.15
e_t at Wilson upper bound	+1.77	+2.14	+2.39	+3.15
all rows, pixel-audited	+6.53			
drop near-miss / drop tiers with author e_t>15%	+6.59 / +3.79			
remove author e_t at random (20 draws)	+6.26			
negatives both humans call incompatible (n=142)	+6.60 [+4.37, +9.09]			
negatives all four passes call incompatible (n=136)	+6.70			

Table 2: Main results: the ladder of variants from Disc to Pred. Each rung changes one design field from the rung above (second column), so adjacent rows price that field; Table 3 gives the differences with intervals. AUC is rank-AUC on the judgement set (15,465 rows) and on the one-shot test (3,830 rows, scored once after every model was frozen); TPR is the true-positive rate at 1% false positives on the judgement set. Two-seed variants list both seeds. Pred-verify-only branches off Pred+readout. **Bold**: best; underlined: second best.

variant	changes from the rung above	AUC	AUC, one-shot test	TPR@1%FPR
Disc	— (joint pass, MLP head)	.9835	.9853	.6263
Disc-grid	grid cross-attention head	**.9844**	**.9856**	**.6366**
Disc-late-grid	frames in separate passes	.9403 / .9438	.9412 / .9446	.1770 / .2419
Disc-late	each frame pooled first	.9336 / .9348	.9436 / .9438	.2139 / .2384
Pred+readout	s_t pass forecasts $\hat{z}$; frozen z	.9428	.9481	.2208
Pred	fixed cosine readout	.9182	.9222	.1898
Pred-verify-only	alignment losses off	.9431 / .9463	.9467 / .9496	.1881 / .1791

10.71 on same-episode offset, as do both held-out structures (+7.14 on `unseen_app`, +5.96 on `unseen_task`; Tables 12 and 6). At 4B the gap is 6.88 points (0.9182 against 0.9870); at 8B the discriminative pass does not fit in memory. Both verifiers still improve when training ends while the gap does not: the last 20,000 steps take 0.08 points off it (Appendix M). Where the design is asymmetric it favours Pred, which selects its readout on a held-back split, carries slightly more trainable parameters and re-estimates its readout statistics per corpus; the routes are matched in data and budget, not in parameters, compute or memory, and inference takes two passes for Pred against one for Disc (Appendix J).

Does it hold on data no one had analysed? Every model and readout was designed after the judgement set had been seen, so pre-registered checkpoints do not remove adaptive bias. After freezing all models, readouts and analyses we built a one-shot test from the 713 `unseen_task` sources the benchmark had never drawn, with the same generator (which reproduces all 23,899 original rows byte for byte) and the same pixel audit, and scored it once: 3,830 rows, Pred's readout statistics frozen at their selection-split values (Appendix A). Disc leads by $+\mathbf{6.30}$ $[+5.47, +7.18]$, and every row of Table 3 keeps its sign except granularity, which moves from +0.79 to −0.08 (seeds −0.24 and +0.08, both intervals spanning zero): on these rows keeping the patch grids buys nothing, and timing with the head matched is +4.27. The set holds out instructions, not applications.

Is it the action-string shortcut? Some tiers perturb only the action, creating an action-string prior. Stratifying AUC within cells of tier × action type × (scroll direction, click column or typed text) moves the gap from 6.53 to 6.45 points, a difference of −0.08 (95% CI $[-0.31, +0.15]$);

matching the negative action-type marginal agrees, and a coarser and a finer stratification give $+5.95$ $[+5.43, +6.51]$ and $+7.02$ $[+6.43, +7.62]$ (Table 9), all inside the registered one-point margin, and the action-side and frame-side tiers give 6.65 and 6.47 on their own (Table 7). Zero-shot `gpt-5.6-terra`, which never saw our negative construction, reaches .8822, below every trained route (Pred .9181 on the same rows), weakest on no-op (.7555 against .9221; Appendix H).

Does it reach a practical operating point? At 1% false positives Pred recovers 0.1898 of true transitions against 0.6263 for Disc, a **43.7-point** difference, because the score distributions differ most in the high-specificity regime a retry gate would use. These rates are read from each model's own ROC, so the threshold is oracle-chosen; whether one transfers is the harsher question of §6 (Appendix C).

5 Where the Cost Enters the Predictive Design

Table 2 gives each rung of the ladder and Table 3 differences adjacent rungs, with intervals, on the judgement set, the one-shot test and two external corpora. Each paragraph below asks about one rung, starting from Pred.

Is it the fixed readout, or the losses that train the forecast? Pred+readout replaces the fixed cosine readout with a learned comparator. A comparator has to be trained, so its 2.45 points belong to the comparator and its verification objective together; they split into 1.08 and 1.37 (Appendix F). Pred-verify-only keeps that predictor and comparator but switches off the four losses that align $\hat{z}$ with z; the predictor is still trained, through the change-localisation loss and the verification objective. It matches Pred+readout in aggregate AUC: $+0.19$ points as a two-seed mean, from $+0.03$ (95% CI $[-0.23, +0.27]$) and $+0.36$ ($[+0.09, +0.60]$), so neither seed favours keeping the losses. Equal AUC is not equal ranking: the two order 4.7% of positive–negative pairs oppositely, in directions that nearly cancel, and at 1% false positives removing the losses lowers the detection rate from 0.2208 to 0.1881 and 0.1791 (Appendix C). **The alignment losses are not free to drop at the operating point, but they are not where the gap comes from**: removing them widens it there.

Is it comparing the frames late? The remaining 4.08 points mix two properties, when the frames meet and whether a forecast intervenes, and the first can be isolated without a forecast. Disc-late encodes the frames in separate passes, pools each to a vector and compares them with a two-layer head; **Disc gives up** 4.93 **points to it**, 1.21 times the residual it was extracted from, and the part left over, Disc-late against Pred+readout, runs the other way (-0.85). Because the ratio exceeds one the two properties do not add, and the fourth cell, early fusion *with* a forecast, has no clean counterpart: feeding s_{t+1} into the pass that forecasts it turns the forecast into one more input of a discriminative model. What survives is an ordering rather than a causal decomposition, and late comparison is the larger term.

Is it pooling, the head, or timing? Separate encoding also pools each frame, so the 4.93 could be granularity. Disc-late-grid keeps both 23×23 patch grids and compares them by cross-attention, with neither detached, through a 1.5M-parameter head a fifth the size of Disc-late's and a third of Disc's. Keeping the grids is worth $+\mathbf{0.79}$ points, below the 1.0 line we registered for calling pooling a confound and below our predicted 2.0–3.5, and Disc still leads by $+\mathbf{4.14}$. That remainder differs from Disc in head form as well as timing, so Disc-grid reads Disc's joint pass, token for token, with Disc-late-grid's comparator: it differs from Disc only in the head and from Disc-late-grid only in timing. The head is worth $-\mathbf{0.09}$ $[-0.23, +0.04]$ and timing with the head matched $+\mathbf{4.41}$ $[+4.07, +4.76]$ against Disc-late-grid's seed 0 ($+4.23$ against its mean), clearing the 3.0 line registered before Disc-grid was trained. The head moves AUC by less than a point on the one-shot test and on both external corpora as well, and at 1% false positives Disc-grid matches Disc while both late variants stay near a fifth (Table 2). **With the comparison head held fixed, the late-comparison cost is fusion timing**, largest on action mismatch ($+9.04$) and under one point on no-op. This rests on one seed of Disc-grid, and it shows that these two heads are interchangeable under early fusion, not that head capacity never matters.

Is it the frozen target? Pred maps into a frozen target while Disc has none, so the gap could be a handicap we imposed. Replacing the frozen target with an EMA copy of the online encoder (Assran

Table 3: Differences between adjacent rungs of the ladder (Table 2), in AUC points, on the judgement set, the one-shot test (§4) and two corpora neither verifier trained on. Rows telescope to the measured total, so only the *split* is empirical; indented rows sum to their parent to within 0.01 of rounding, and the last row is a side branch and an equality in *aggregate* AUC only. Rows with Disc-late or Disc-late-grid are two-seed means; Disc-grid has one seed. In-distribution, per seed, Disc gives up $+4.99$ $[+4.59, +5.40]$ and $+4.87$ $[+4.49, +5.28]$ to Disc-late; Disc-late-grid gains $+0.67$ $[+0.36, +0.98]$ and $+0.90$ $[+0.60, +1.23]$ over Disc-late and trails Disc by $+4.32$ $[+3.98, +4.67]$ and $+3.97$ $[+3.62, +4.31]$ (paired bootstraps over source triplets). The remainder bundles the forecast with head form and objective. External columns re-estimate Pred's readout statistics per corpus and so measure ranking transfer; with the in-distribution statistics frozen the totals become $+12.41$ and $+21.72$ (per-seed external values: Appendix I).

component (AUC points)	judgement	one-shot test	CAGUI	GUI-360
late comparison: Disc − Disc-late	$\mathbf{+4.93}$	$+4.16$	$\mathbf{+7.20}$	$\mathbf{+19.10}$
timing + head form: Disc − Disc-late-grid	$\mathbf{+4.14}$	$+4.24$	$+6.16$	$+15.05$
head form: Disc − Disc-grid	-0.09	-0.04	$+0.19$	-0.95
timing, same head: Disc-grid − Disc-late-grid	$\mathbf{+4.23}$	$+4.27$	$+5.97$	$+15.99$
granularity: Disc-late-grid − Disc-late	$+0.79$	-0.08	$+1.04$	$+4.06$
remainder: Disc-late − Pred+readout	-0.85	-0.44	$+0.28$	$+2.02$
learned comparison stage: Pred+readout − Pred	$+2.45$	$+2.59$	$+4.47$	-1.90
comparison head (Table 18)	$+1.08$	—	$+2.78$	-2.12
added objective (Table 18)	$+1.37$	—	$+1.69$	$+0.23$
total: Disc − Pred	$\mathbf{+6.53}$	$\mathbf{+6.30}$	$\mathbf{+11.94}$	$\mathbf{+19.22}$
alignment losses: Pred-verify-only − Pred+readout	$\mathbf{+0.19}$	$+0.01$	—	—

et al., 2025; Grill et al., 2020; Tarvainen & Valpola, 2017) collapses the target instead: at momentum 0.996 Pred-EMA reaches AUC 0.7766 against 0.9182, the mean pairwise target cosine over 128 unrelated frames rises to 0.9405 and the effective rank falls to 74.0, while the forward loss becomes easier (0.0826 against 0.4461), as expected of a loss with no lower bound on the distance between unrelated states (Tables 10 and 11; Appendix N). Slower momenta and four registered repairs, two on the predictor and two on the target's own forward pass, fail the registered conjunction of cosine, rank and AUC as well (Appendix O): m=0.9999 returns the cosine to the frozen reference (0.8191) but not the rank (65.0 against 90.6), DINO-style centering scores below the collapsed model it was meant to repair (0.7572), and a covariance penalty leaves a geometry worse than the failure it targets (cosine 0.9881). **Across seven target rules none recovers the** 6.53 **points.** This bounds the substitutions, not the explanation: we identify no mechanism, and the centering variant diverged on l_{noop} over its last thousand steps, so its failure carries little evidence either way.

6 Gains That Do Not Survive a Change of Distribution

Does what we measured in-distribution survive a shift? We change the distribution three ways—the corpus, the negative tier and the source of candidate actions—and ask each time whether it is still there.

Corpus shift. Zero-shot on CAGUI and GUI-360 the gap grows from $+6.53$ to $+11.94$ and $+19.22$ points (Table 15, Figure 4), and to $+12.41$ and $+21.72$ if Pred's readout statistics are frozen at their in-distribution values rather than re-estimated per corpus. Pred degrades most on no-op and same-episode offset (Figure 5): a byte-identity rule reaches 0.9978 on no-op where Pred reaches 0.9138, and Pred's readout on an untrained adapter, at chance overall (0.4986, so training is worth $+42$ points), is nearly inverted there (0.0162), since an unchanged screen is what an untrained predictor matches (Table 16). The decomposition transfers (Table 3): the late-comparison row grows to 7.20 and 19.10, the part surviving the grids stays the larger one (6.16 and 15.05, against granularity's 1.04 and 4.06, whose desktop seeds differ by two points, $+5.05$ and $+3.06$), and the remainder stays an order of magnitude smaller. Disc-grid matches Disc off-distribution as well (head form $+0.19$ and -0.95), so with the head fixed timing takes 5.97 and 15.99 points. Two things do *not* transfer: the learned comparison stage, below, and Disc-late's advantage over Pred, since the two are indistinguishable on desktop (-0.08 $[-0.43, +0.26]$ and $+0.32$ for the two seeds); Disc-late-grid,

which keeps the grids, stays 5.0 and 3.4 points ahead of Pred there. A threshold fixed in-distribution at 5% false positives and moved unchanged leads Pred to reject almost everything, while Disc keeps most true transitions (Appendix C).

A learned comparison stage reverses on desktop. The comparison stage that adds 2.45 points in-distribution costs 1.90 on desktop. The comparator contributes -2.12 of that while the objective that trains it contributes $+0.23$ (Table 3), so the reversal sits with the learned comparison, not with the predictive alignment objective. The benchmark's action-string prior also feeds the readout's in-distribution advantage (Appendix I), another reason to treat this gain as distribution-dependent.

Negative-distribution shift. Replacing half of the action negatives with near-misses improves the near-miss tier by 3.87 points and the benchmark overall by 0.32, while the registered endpoint, the far-displaced action-mismatch tier, moves -0.36 (interval containing zero) and -1.94 on desktop. It teaches the distinction it was built to teach and no more, and that distinction is the least trustworthy one we measure (Appendix G).

Candidate-source shift. Scoring before execution has no successor, so Disc cannot be applied, but a world model is not the only way to score (s_t, a). A goal-conditioned scorer on the same backbone ranks the expert first 62.2% of the time among fifteen constructed perturbations (6.25% chance), yet on a behaviour-cloned policy's own candidates its top-1 is 0.5048, below the policy's greedy decode at 0.6588, the one comparison stable across de-duplication rules (Figure 6; Appendices K and L). A control that removes only the instruction to predict ranks the expert 1.16, 2.49 and 3.13 points *higher* at $N = 4, 8, 16$ (one seed), so on this probe it was not prediction that delivered the capability.

Four instances, one pattern, and the fourth is ours: each intervention helps where it was built and fails under a shift, and so does Disc-late's advantage over Pred. What survives is the *attribution*. Narrowing a gap and locating it are different claims, and only the second transferred.

7 Deployment-Time Interface Anomalies

Do the verifiers accept transitions that really happened but went wrong? Every shift so far kept positives from sessions that went as intended. GUI-Robust records human operators meeting real interface anomalies—ad pop-ups, captchas, cookie banners, login walls, page loading and network loss (Yang et al., 2025). These are L1 positives too, so we compare them with ordinary successors from the same sessions, stratified by frame-change magnitude because anomalies change more pixels, against a registered target separation of 0.50.

Both verifiers score the genuine anomalies as *less* compatible than ordinary successors: change-stratified AUC 0.683 (95% CI $[.588, .777]$) for Pred and 0.759 ($[.662, .839]$) for Disc (Table 17). The separation becomes refusals. At a threshold set within each stratum to accept 90% of ordinary successors, the two verifiers accept 32 and 38 percentage points fewer anomalies ($[+10.6, +43.8]$ and $[+16.7, +54.4]$), that is 58% and 52% of them. These are refusals of transitions that occurred, but not a measured false-retry rate, since a network failure may still warrant a retry. Pooled across strata the predictive refusal disappears (-2.5 points, interval spanning zero) while the discriminative one does not ($+29.0$), because the predictive score rewards change. With 72 anomaly steps and thresholds read from about 37 ordinary rows per stratum, we read this as a failure-mode signal, not a per-type estimate.

The result also clarifies what the benchmark measures. Its positives come from sessions that went as intended, whereas an interface anomaly is a recorded transition that is undesirable and still dynamically real. **The results are consistent with both verifiers having learned transition *typicality*, narrower than compatibility itself**—the reading the candidate-source experiment also invites—so a high score against constructed negatives can reflect what an interface ordinarily does rather than what it did. This is not an identified mechanism: 72 steps cannot separate typicality from features correlated with it. Positives from GUI-360's released failed trajectories likewise separate weakly once stratified by frame change (Appendix E).

8 LIMITATIONS

What was held fixed. All models are LoRA fine-tunes of one backbone family at three sizes, with no matched discriminative verifier at 8B, and the predictive objective is one family, per-patch cosine regression. Adding a patch-level contrastive term whose negatives are other samples' target patches lowered AUC on the frozen target and barely moved the collapsed one (Appendix O), but we tested it only as an addition, at one weight and one seed; an objective built around such a term need not behave the same way. The target-encoder results cover three momenta and four repairs, not end-to-end training under an embedding regulariser (Maes et al., 2026), with no momentum beyond 0.9999 and no identified mechanism of collapse, and our covariance repair decorrelates each sample's own patches rather than taking the cross-sample VICReg form, which a per-device batch of eight cannot estimate.

How far the ladder isolates. Disc-late needs a wider head and pools; Disc-late-grid removes the pooling but not the head mismatch; Disc-grid removes the head mismatch on one seed, and no variant changes head size alone. The variants also ran at different per-device batch sizes (effective batch equal), and Disc-late-grid's seeds agree to within a point in-distribution and on CAGUI but not on desktop.

What the labels measure. Everything is offline and the primary negatives are constructed: a genuinely failed action still produces a real successor, and no label records whether an action executed. The author audit finds a third of near-misses compatible and 80% of no-op labels valid only if the action executed; the independent annotators put the near-miss rate lower (23%) and did not record the execution premise; every pass is an equal-allocation sample of 30 rows per tier, and the annotators agree only moderately on the negatives (κ=.49), mostly over what counts as undecidable. The one-shot test holds out instructions, not applications, and every design choice was made after the judgement set had been seen. The anomaly analysis rests on 72 steps and on thresholds read from about 37 ordinary rows per stratum, so we read its direction and not its magnitude.

What the gap does not measure. Disc is a budget-matched control rather than a deployed critic, so 6.53 is evidence about the predictive design we tested, not a general ordering of the paradigms and not the cost of a pre-execution score. Goal-conditioned and linguistic step critics already move agent metrics when wired into accept, retry or recovery loops (Qian, 2026; Hu et al., 2025; Xiong et al., 2025; Chen et al., 2025a; Chae et al., 2025b; Zhang et al., 2026b); we do not measure whether the 6.53-point gap, or the 43.7-point gap at 1% FPR, changes task success, invalid retries or recovery when Pred or Disc is the hook, which needs recovery labels and a frozen policy.

9 CONCLUSION

A predict-then-compare verifier and a joint discriminative one, trained on the same data and budget, differ by 6.53 AUC points on GUI step verification, and the difference survives the checks a reader would ask for first: seeds, a shortcut control, a one-shot test and independent relabelling. The ladder between them says where it comes from. The losses that make the forecast match the future are nearly free in aggregate AUC; what the predictive design pays for is comparing the two screens after encoding them apart, and within that late comparison, neither pooling (in-distribution) nor the form of the comparison head but the absence of joint attention across the action and both frames.

Three consequences follow for the designs we tested. **A verifier that will see the successor should let the action and both screens meet inside the encoder**: separate encoding cost more than any other rung of the ladder, and a richer late comparator did not buy it back. **Pre-execution scoring is not by itself a reason to verify with a world model**: on our one-seed probe a non-predictive state–action scorer ranked candidates better than the predictive one. **Scores on constructed negatives should be read next to real transitions**: both verifiers refuse genuine interface anomalies, so a high benchmark score can encode what an interface ordinarily does rather than what it did. Whether the same ordering holds for other backbone families, for pixel and language world models, and in closed-loop recovery is the natural next measurement.

REPRODUCIBILITY STATEMENT

Benchmark construction, both audit criteria with their thresholds, the negative generator's parameters, and every training configuration are specified in Appendices A–J, including the loss weights, the read-out candidates and the selection protocol, the comparator architecture, and the scope of the EMA update. Every judgement line quoted in the text, including each abort and recovery threshold, was fixed in a pre-registration written before any result it governs existed, and each is stated in the text or in the caption of the table it governs. The shortcut-neutralised re-analysis, the action-only prior, the operating points, the two zero-learning baselines, the label-audit bounds and both real-data analyses require no further training: each is computed from stored per-example scores and the source frames under the procedures and thresholds given above. We will release the following with the published version of the paper: the benchmark construction and audit code, with the judgement-set and one-shot-test sample identifiers and the negative-generation parameters that reproduce the cohort from the public source corpora, which we do not re-host; the training configuration of every model; the per-example scores behind every table and figure; all four label-audit passes; the dated pre-registrations, including every change we made to a judgement line or a measurement after seeing results, and why; and the scripts that compute every reported number from those scores.

AI USE STATEMENT

Throughout this work, the authors used a large language model as an assistant for coding, experimentation, and writing. Specifically, the model was used to implement and review experiment code, run and summarize evaluations, draft the pre-registration documents, and draft and revise several sections of the manuscript. It was also used to perform the language-model pass of the label audit (Appendix G). This pass is reported explicitly as a model-based audit and is not presented as human annotation. All experimental designs, judgement criteria and pre-registrations were reviewed and approved by the authors before execution, and all reported numbers were verified against the outputs of the experiment code. The authors take full responsibility for this paper.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Tianpeng Bu, Xin Liu, Qihua Chen, Hao Jiang, Shurui Li, Hongtao Duan, Lu Jiang, Lulu Hu, Bin Yang, and Minying Zhang. Recovering policy-induced errors: Benchmarking and trajectory

synthesis for robust GUI agents. In *International Conference on Machine Learning (ICML)*, 2026. Spotlight.

Yilin Cao, Yufeng Zhong, Zhixiong Zeng, Liming Zheng, Jing Huang, Haibo Qiu, Peng Shi, Wenji Mao, and Wan Guanglu. MobileDreamer: Generative sketch world model for GUI agent, 2026. URL https://arxiv.org/abs/2601.04035.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2410.13232.

Hyungjoo Chae, Sunghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, Dongwook Choi, Minseok Kang, Gwanhoon Im, ByeongUng Cho, Hyojun Kim, Jun Hee Han, Taeyoon Kwon, Minju Kim, Beongwoo Kwak, Dongjin Kang, and Jinyoung Yeo. Web-Shepherd: Advancing PRMs for reinforcing web agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b. URL https://arxiv.org/abs/2505.15277.

Cong Chen, Kaixiang Ji, Hao Zhong, Muzhi Zhu, Anzhou Li, Guo Gan, Ziyuan Huang, Cheng Zou, Jiajia Liu, Jingdong Chen, et al. GUI-Shepherd: Reliable process reward and verification for long-sequence GUI tasks. *arXiv preprint arXiv:2509.23738*, 2025a.

Jingxuan Chen, Derek Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Li Yixing, Xurui Zhou, Weiwen Liu, Shuai Wang, Kaiwen Zhou, Rui Shao, Liqiang Nie, Yasheng Wang, Jianye HAO, Jun Wang, and Kun Shao. SPA-Bench: A comprehensive benchmark for SmartPhone agent evaluation. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Yicheng Fu, Raviteja Anantha, Prabal Vashisht, Jianpeng Cheng, and Etai Littwin. UI-JEPA: Towards active perception of user intent through onscreen user activity. *arXiv preprint arXiv:2409.04081*, 2024. URL https://arxiv.org/abs/2409.04081.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning, 2024. URL https://arxiv.org/abs/2403.00504.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the Internet? Model-based planning for web agents. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2411.06559.

Yu Guan et al. Computer-using world model, 2026. URL https://arxiv.org/abs/2602.17365.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zhiyuan Hu, Shiyun Xiong, Yifan Zhang, See-Kiong Ng, Anh Tuan Luu, Bo An, Shuicheng Yan, and Bryan Hooi. Guiding VLM agents with process rewards at inference time for GUI navigation, 2025. URL https://arxiv.org/abs/2504.16073.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2020.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

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 (Datasets and Benchmarks Track)*, volume 37, 2024.

Xiangyan Liu, Kaixin Li, Haonan Wang, Biao Wu, Meng Fang, Longxu Dou, Chao Du, Michael Qizhe Shieh, and Tianyu Pang. Scaling GUI agents with visual state transitions, 2026. URL https://arxiv.org/abs/2607.24112.

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

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. GUIOdyssey: A comprehensive dataset for cross-app GUI navigation on mobile devices. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22404–22414, 2025.

Dezhao Luo, Bohan Tang, Kang Li, Georgios Papoudakis, Jifei Song, Shaogang Gong, Jianye Hao, Jun Wang, and Kun Shao. A generative visual GUI world model for app agents. *arXiv preprint arXiv:2504.13936*, 2025. URL https://arxiv.org/abs/2504.13936.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Jian Mu, Chaoyun Zhang, Chiming Ni, Lu Wang, Bo Qiao, Kartik Mathur, Qianhui Wu, Yuhang Xie, Xiaojun Ma, Mengyu Zhou, et al. GUI-360: A comprehensive dataset and benchmark for computer-using agents. *arXiv preprint arXiv:2511.04307*, 2025.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Jiachen Qian. VisCritic: Visual state comparison as process reward for GUI agents. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2606.24525.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

Tao Xiong, Xavier Hu, Yurun Chen, Yuhang Liu, Changqiao Wu, Pengzhi Gao, Wei Liu, Jian Luan, and Shengyu Zhang. GUI-PRA: Process reward agent for GUI tasks, 2025. URL https://arxiv.org/abs/2509.23263.

Jingqi Yang, Zhilong Song, Jiawei Chen, Mingli Song, Sheng Zhou, Linjun Sun, Xiaogang Ouyang, Chun Chen, and Can Wang. GUI-Robust: A comprehensive dataset for testing GUI agent robustness in real-world anomalies. *arXiv preprint arXiv:2506.14477*, 2025.

Xiao Yu, Baolin Peng, Michel Galley, Hao Cheng, Qianhui Wu, Janardhan Kulkarni, Suman Nath, Zhou Yu, and Jianfeng Gao. Dyna-Mind: Learning to simulate from experience for better AI agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=F848aPzCJy.

Linhao Zhang, Aiwei Liu, Yuan Liu, and Xiao Zhou. DiffSpot: Can VLMs spot fine-grained visual differences in web interfaces?, 2026a. URL https://arxiv.org/abs/2605.29615.

Yuzhe Zhang, Xianwei Xue, Xingyong Wu, Mengke Chen, Chen Liu, Xinran He, Run Shao, Feiran Liu, Huanmin Xu, Qiutong Pan, and Haiwei Wang. Don't act blindly: Robust GUI automation via action-effect verification and self-correction. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026b. URL https://aclanthology.org/2026.acl-long.1335/.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, Jie Xie, Wei Zhou, Wang Xu, Yuanheng Zhang, Zhou Su, Zhongwu Zhai, Xiaoming Liu, Yudong Mei, Jianming Xu, Hongyan Tian, Chongyi Wang, Chi Chen, Yuan Yao, Zhiyuan Liu, and Maosong Sun. AgentCPM-GUI: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*, 2025.

Jiani Zheng, Lu Wang, Fangkai Yang, Chaoyun Zhang, Lingrui Mei, Wenjie Yin, Qingwei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. VEM: Environment-free exploration for training GUI agent with value environment model, 2025. URL https://arxiv.org/abs/2502.18906.

Yuhao Zheng, Li'an Zhong, Yi Wang, Rui Dai, Kaikui Liu, Xiangxiang Chu, Linyuan Lv, Philip Torr, and Kevin Qinghong Lin. Code2World: A GUI world model via renderable code generation. *arXiv preprint arXiv:2602.09856*, 2026. URL https://arxiv.org/abs/2602.09856.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025.

A Benchmark construction and audit

Corpora and splits. The benchmark is built from two mobile GUI corpora (Table 4). Android Control provides the held-out structure: `unseen_task` holds out instructions within seen applications and `unseen_app` holds out whole applications, with a third in-distribution split reserved for signal selection. GUI Odyssey contributes training rows only, so `unseen_app` is held out within Android Control rather than across both corpora. We quantified the resulting exposure: 26 of the 168 applications in the judgement set (15.5%, 16.0% of rows) also appear in GUI Odyssey, with no instruction shared verbatim. Splitting the judgement set on that boundary moves the gap the wrong way for a leakage account—$+6.64$ $[+6.11, +7.14]$ on genuinely unseen applications against $+5.98$ $[+5.03, +6.90]$ on the exposed ones—so the exposure that exists works against our claim rather than for it.

The one-shot test set. The original benchmark drew 1,500 source triplets per held-out split. `unseen_task` has 2,213, so 713 were never drawn, never trained on (the split is excluded from training as a whole) and never looked at. We regenerated the benchmark with the same code, prepending the original v0.1 rows so that the near-miss tier's sequential random draws are unchanged; all 23,899 original rows reproduce byte for byte. The 3,898 new rows (713 positives)

went through the same pixel audit, which flags 68. The protocol, models and readouts were registered before any model scored the set, and the set was scored once. It holds out instructions within seen applications, not applications.

What the no-op tier assumes. A constructed no-op pairs the recorded action with an unchanged screen and labels it incompatible. That label assumes the action *executed*: a click that left the screen unchanged because it failed, or because the interface was still loading, produces the same pair and is not incompatible in the sense §3 defines. We do not treat such cases as counterexamples to the tier—they are outside what an offline recorded corpus can distinguish—which is one reason §7 evaluates real interface failures against a separate corpus rather than reading them off this benchmark. The pixel audit defined below is not used to relabel these rows.

Table 4: Corpora. Training sees only the first two; every other row is evaluation-only. The last two carry *real* rather than constructed material: GUI-360's released failed trajectories (Appendix E) and GUI-Robust's human-recorded anomaly steps (§7).

corpus	role	domain	source
Android Control	train + held-out splits	English mobile	Li et al. (2024)
GUI Odyssey	train only	English mobile, cross-app	Lu et al. (2025)
CAGUI	zero-shot transfer	Chinese mobile	Zhang et al. (2025)
GUI-360	zero-shot transfer	Windows desktop	Mu et al. (2025)
GUI-360 (`fail`)	real failed-trajectory positives	Windows desktop	Mu et al. (2025)
GUI-Robust	real anomaly successors	web, zh + en	Yang et al. (2025)

Negative tiers. Six tiers isolate different failures a verifier should catch. *Cross-episode* draws a frame from a different episode. *Same-episode offset* draws another frame of the same episode at $|\Delta t| \geq 2$. *No-op* sets $s_{t+1} := s_t$. *Action mismatch* keeps the true successor and displaces the action far. *Action type mismatch* keeps the true successor and swaps the action type. *Action near-miss* displaces a click into the ring $r \in [0.09, 0.18]$ of normalised coordinates; realised radii run 0.081–0.180 after clipping to screen bounds, and **none of the 2,646 near-miss negatives falls inside the per-axis expert-equivalence tolerance of** 0.08. The median displacement is 0.133 of the screen diagonal, 351 px on a 1080×2400 screen, against 0.280 (736 px) for the far-displacement tier. **That tolerance is on coordinates, and the generator never consults element bounds**, so clearing it guarantees a different *point*, not a different *decision*: a displacement of this size stays inside a full-width list row, a large card or a tab-bar segment often enough that the blinded audits find a quarter to a third of sampled near-misses compatible after all (Appendix G). An element-aware generator would need layout bounds, which these corpora do not ship; we report the tier with that ceiling rather than dropping it.

Two prior revisions. A first revision removed an action-type marginal leak: perturbation turned clicks into scrolls half the time and could not perturb one action type at all, so the type distribution of negatives differed sharply from positives (scroll 47.7% of negatives against 12.5% of positives) and "guess false when you see a scroll" scored above chance. The fix relabels by pair type and splits the affected cases into the *action type mismatch* tier. A second revision added the near-miss tier, because the existing click negatives were displaced a median of 736 px—a different screen region, not a near-miss.

Audit criteria. Both rulers threshold a mean absolute pixel difference on images scaled to 256 px and converted to grayscale, with intensities in $[0, 1]$; **both thresholds are** 0.01, and criterion B's comparison window is a square of half-width 0.045 in normalised coordinates. **A (frame identity)** compares mean absolute pixel difference against the true successor at 256 px grayscale; it applies wherever s_{t+1} is asserted to differ (same-episode offset, no-op, cross-episode) and is meaningless on the `action_*` tiers, whose s_{t+1} *is* the true successor. **B (displaced click, same element)** compares local image patches of half-width 0.045 in normalised units around the original and displaced points in s_t; it applies wherever the action is perturbed and both points carry coordinates. B is a proxy: it over-flags flat backgrounds—measurably so, at 73% of the rows it flags (Figure 3)—and under-flags displaced points that straddle an icon, so we treat its count as a bound and report affected tiers under every exclusion regime. Neither count is used to relabel a row.

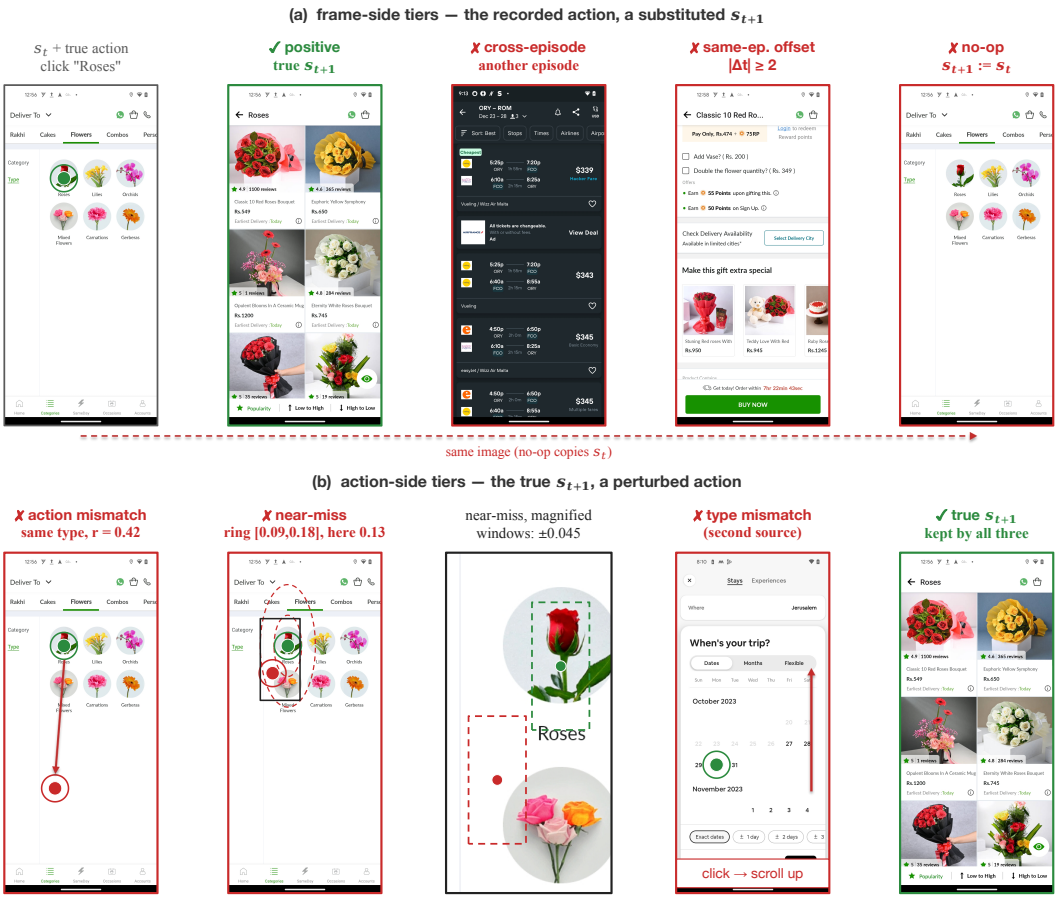

Figure 2: The six negative tiers as real benchmark rows, not schematics. **(a)** The action is the recorded one and s_{t+1} is replaced; the positive is the recorded transition. **(b)** s_{t+1} is the true successor and the action is perturbed, so all three share the frame on the right and what a verifier must read is the action. Green marks ground truth, red the corruption. The dashed ellipses are the registered sampling annulus $r \in [0.09, 0.18]$ in normalised coordinates—an ellipse in pixels because the screen is 1080×2400—and the magnified panel draws the two ± 0.045 windows of criterion B, the local patches around the original and the displaced click. Every row shown is from the judgement splits and none is audit-flagged.

Disclosed priors, measured on the benchmark. Some of the label is guessable from the action string alone, and we quantify how much rather than only naming it. We fit a non-parametric prior—smoothed empirical $P(\text{positive})$ keyed on action type, scroll direction and a 6×6 bin of the click coordinate—on the *in-distribution* split only, and score the judgement set with it (Table 5). It reads at most .83, and the structure of where it reads is the useful part:

Table 5: What an action-only prior earns per tier, fitted on the in-distribution split and evaluated on the judgement set. The three tiers that perturb the *frame* leave the action untouched, so the prior is at chance there by construction; the three that perturb the *action* are where the benchmark is guessable.

prior	act. mism.	near-miss	type mism.	cross-ep.	no-op	same-ep. off.
type + direction + coord. bin	.7054	.6866	**.8103**	.5000	.4967	.4910
type only	.6186	.5825	**.8340**	.5000	.4990	.4963

The worst tier is action *type* mismatch, at .81–.83, not action mismatch; and the coarser prior scores *higher* there, so what it exploits is the type marginal itself—true scroll directions run 439

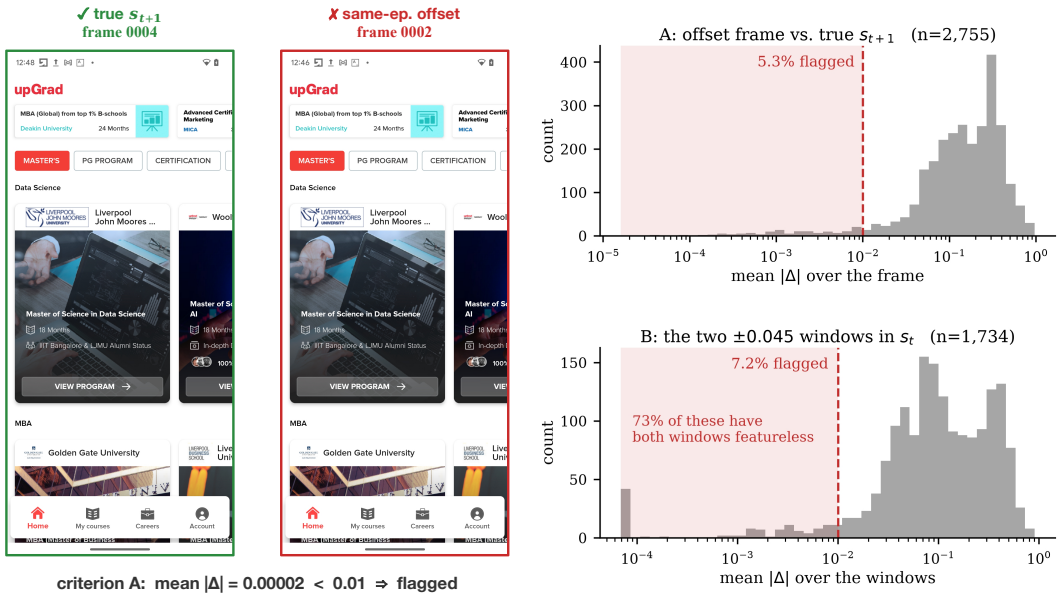

Figure 3: The two rulers on judgement-split rows. **Left:** a `same_episode_offset` negative whose frame *is* the true successor at mean $|\Delta| = 1.6 \times 10^{-5}$—the episode returned to a screen it had already visited two steps earlier, so the row asserts something false. **Right:** the statistics each ruler thresholds. Criterion A separates cleanly. Criterion B does not: 73% of the rows it flags have *both* comparison windows essentially featureless, which is the flat-background failure mode and the reason we report its count as a bound rather than a rate. Counts cover the two judgement splits only and therefore do not sum to the 398 rows excluded benchmark-wide.

down against 53 up while perturbation deterministically flips, and negatives of this tier are 75% scroll against a positive pool that is mostly click. **The other three tiers sit at** .49–.50**, exactly chance, and structurally must**: they replace s_{t+1} and leave the action alone, so no action-only rule can separate them. Half the benchmark is guessable-in-part and half cannot be, and we distinguish them rather than apologising for all six.

Disclosure is not control, so we also neutralise the prior and re-measure (§4, Table 9).

B PER-TIER RESULTS BY SCALE, AND THE INPUT ABLATION

Collapse is partial, and the tiers say where. The contrastive and no-op terms act on the *pooled* projection, not the patch grid, so the account we held before testing the repairs predicts that unfreezing should damage fine-grained patch geometry while leaving pooled-supervised tiers intact. Measured against Pred, Pred-EMA+readout's tiers do split that way: the tiers asking *which screen this is* degrade sharply (same-episode offset −8.47, cross-episode −4.49) while every tier a pooled term still supervises holds or improves (action type mismatch +0.80, action mismatch +0.20, no-op +1.36—no-op being exactly the tier the no-op term trains). Collapse destroys **episode-level frame identity**, and a learned comparator cannot recover information the representation discarded. Note that no-op *survives* here while collapsing under desktop shift (§6): the two failures are not the same failure, and only same-episode offset is common to both. **This prediction being borne out is what made the repair theory worth testing; the repairs still refuted it** (§5).

Table 6 is that per-tier grid, including the 4B and 8B rows cited in §4.

Table 6: Per-tier rank-AUC on the judgement set (union of the two held-out splits, 2,898 source triplets, 15,465 samples, 398 audited rows excluded). Every tier favours the discriminative verifier at both scales where both exist. All models see the same 1.28M samples; 4B and 8B reach it in 160,000 steps at half the per-device batch. The 8B row reports its own frozen signal (`combo`); its `x_2nce_mink` reads .9138, so the conclusion does not turn on that choice. **There is no discriminative 8B verifier**, so no gap can be quoted at 8B.

scale	verifier	act. mism.	near-miss	type mism.	cross-ep.	no-op	same-ep. off.	**all**
2B	pred.	.8888	.8722	.9531	.9876	.9138	.8742	**.9182**
2B	disc.	.9857	.9329	.9918	.9965	.9964	.9813	**.9835**
4B	pred.	.8906	.8805	.9498	.9819	.9093	.8807	**.9182**
4B	disc.	.9885	.9462	.9952	.9982	.9957	.9858	**.9870**
8B	pred.	.8655	.8461	.9653	.9755	.9220	.8791	**.9135**
8B	disc.	*not trained—the discriminative verifier already peaks at* 69.6 *GB at 4B*						

Table 7 is the input ablation behind the action-side and frame-side gaps in §4.

Table 7: What each verifier scores with one input removed, and the gap split by whether a tier needs the action. Blinding the action leaves the three action-side tiers at chance for both verifiers, so the +6.65 on that half is earned by reading the action. Intervals are paired bootstraps over source triplets.

	Pred	Disc	Disc − Pred
judgement-set AUC, all tiers			
normal	.9182	.9835	+6.53
action blinded (frames only)	.6485	.7786	+13.00
frames blinded (action only)	.5463	.5187	**−2.76**
normal condition, split by tier group			
action-side tiers	.9008	.9673	**+6.65** [+6.06, +7.28]
frame-side tiers	.9271	.9917	+6.47 [+6.01, +6.94]

C Operating points and threshold transfer

§4 reports that the gap at 1% false positives is 43.7 points against an AUC gap of 6.53. Two further details sit behind that.

A threshold fixed in-distribution does not travel. Fixing each verifier's threshold at 5% false positives on the judgement set and moving it unchanged, Disc realises 7.5% and 10.4% false positives on CAGUI and GUI-360 while retaining .897 and .659 of true transitions. Pred realises .000 and .021—it rejects almost everything, recovering .002 and .050 (Table 8).

Two different quantities appear in this table and they should not be read as one. TPR@FPR is read off each verifier's own ROC *on the set being scored*: it says what an oracle-chosen threshold would achieve there, and it involves no transfer. The threshold-transfer block is the deployable question—a threshold chosen once and then used elsewhere—and reports the *realised* false-positive rate, which is what degrades. **Pred also has no transferable absolute threshold at all** in its headline form, because that signal is a mixture of components whose z-statistics are re-estimated within each external corpus (they are fixed on the selection split for everything in-distribution, Appendix J); across corpora the identity of "score $> \tau$" therefore changes, and the relative weight of the components changes with it. The threshold-transfer rows accordingly use Pred's single-component signals, which do have corpus-independent units. So the transfer row and the AUC row are not the same scorer, and we do not read the transfer failure as a property of the headline read-out—it is a property of the predictive route's absolute scale, measured on the part of that route that has one. **An AUC tie is not one ranking.** Pred-verify-only and Pred+readout tie on aggregate AUC (§5), and over the 36.4M (positive, negative) pairs of the judgement set they disagree on 4.7%: 2.36% are ordered correctly only by Pred-verify-only against 2.34% only by Pred+readout, and that near-cancellation

Table 8: Operating points on the judgement set, and what happens to a threshold when the corpus changes. TPR@FPR is read off each verifier's own score distribution. Threshold transfer fixes the threshold at 5% FPR in-distribution and reports the *realised* rate elsewhere. Pred-verify-only is shown at seed 0 (.9431); its two-seed mean is .9447.

variant	AUC	TPR@1%FPR	TPR@5%FPR	TPR@10%FPR
Pred	.9182	.1898	.6049	.7781
Pred+readout	.9428	.2208	.6663	.8671
Pred-verify-only (seed 0)	.9431	.1881	.6580	.8613
Disc-late (seeds 0, 1)	.9336 / .9348	.2139 / .2384	.6094 / .6259	.8009 / .7947
Disc-late-grid (seeds 0, 1)	.9403 / .9438	.1770 / .2419	.6322 / .6587	.8406 / .8489
Disc-grid (seed 0)	.9844	.6366	.9438	.9793
Disc	**.9835**	**.6263**	**.9438**	**.9769**
threshold fixed at 5% FPR in-distribution, realised FPR / TPR elsewhere				
		ours	CAGUI	GUI-360
Pred (single-component)		.050 / .605	.000 / .002	.021 / .050
Disc		.050 / .944	.075 / .897	.104 / .659

is what the tie consists of (Spearman ρ=.90, Kendall τ_b=.73; both seeds behave the same). The consequence is visible at the operating point: the positives the two verifiers accept at 1% false positives overlap at a Jaccard index of only .35.

Calibration is not what separates the verifiers: a one-parameter Platt scaler brings both to an expected calibration error of .011–.017, so the difference is in ranking and in where the usable operating points lie, not in how well-calibrated the scores are.

D FURTHER TABLES BEHIND THE MAIN TEXT

Table 9 is the per-tier form of the shortcut control in §4.

Table 9: The gap with the action-type shortcut neutralised, two ways. Stratified: AUC computed within (tier × action type × scroll direction, click column or typed text) cells and pooled by pair count. Matched: negatives resampled to the positives' action-type marginal.

tier	unstratified	stratified	matched marginal
action mismatch	+9.70	+8.26	+8.11
near-miss	+6.08	+6.15	+6.08
action type mismatch	+3.87	+3.33	+3.18
cross-episode	+0.89	+1.05	+0.89
no-op	+8.26	+8.33	+8.27
same-episode offset	+10.71	+10.75	+10.63
pooled	**+6.53**	**+6.45**	—

Tables 10 and 11 are the numbers behind §5: removing the frozen target, and the geometry of each repair; which tiers the collapse damages is in Appendix B.

Table 10: Removing the frozen target.

	AUC	vs. Pred
Pred (frozen target)	.9182	—
Pred-EMA	**.7766**	**−14.16**
Pred-EMA+readout	.8069	−11.13
readout signal only	.8929	−2.53
Disc	.9835	+6.53

Table 11: Target-space geometry over 128 mutually unrelated held-out frames. Each frame is encoded by the model's target path, mean-pooled over the patch grid to one 2048-d vector and ℓ_2-normalised; the cosine column is the mean off-diagonal of the resulting Gram matrix. Effective rank is $\exp(-\sum_i p_i \log p_i)$ over $p_i = \sigma_i / \sum_j \sigma_j$, the singular values of the *mean-centred* matrix. **Its ceiling is therefore** 127**, not** 2048: centring 128 samples leaves at most $n-1$ non-zero singular values, so Pred's 90.6 is 71% of the achievable rank.

variant	pairwise cos.	eff. rank
Pred (frozen)	**.8173**	**90.6**
Pred+readout (frozen)	.8173	90.6
Pred-EMA	**.9405**	74.0
Pred-EMA+readout	.9219	**51.9**
+ patch InfoNCE	.3590	57.5
+ variance floor	.5449	**30.6**
+ centering	**.0021**	**23.7**
+ covariance	**.9881**	76.9
+ InfoNCE (frozen ctrl.)	.8173	90.6

Table 12 is the paired interval for every tier, the source of the claim that none contains zero.

Table 12: Per-tier difference (discriminative − predictive) at 2B with the intervals behind the claim that no tier contains zero. Paired bootstrap over 2,898 source triplets, B=2,000, seed 0; $P(a>b)$ is the fraction of resamples favouring the discriminative verifier. Each tier is estimated by its own bootstrap.

tier	difference	95% CI	$P(a>b)$
same-episode offset	+10.71	[+9.81, +11.58]	1.000
action mismatch	+9.70	[+8.74, +10.71]	1.000
no-op	+8.26	[+7.54, +8.95]	1.000
near-miss	+6.08	[+5.11, +7.02]	1.000
action type mismatch	+3.87	[+3.34, +4.44]	1.000
cross-episode	+0.89	[+0.65, +1.15]	1.000
all	**+6.53**	**[+6.08, +6.99]**	1.000

Table 13 is the resampling-unit check cited in §3.

Table 13: Three resampling units for the same paired bootstrap (B=2,000, seed 0). The point estimates are identical by construction; only the intervals move, and no conclusion turns on the choice. Clustering by source *narrows* slightly against per-sample resampling because it keeps a source's positive and its negatives in the same draw, and that pairing effect slightly outweighs the within-cluster correlation it introduces; the episode unit, with 4.8× fewer clusters, is the widest.

comparison	source (2,898)	episode (609)	sample (15,465)
Disc − Pred	+6.53 [+6.08, +6.99]	+6.53 [+5.94, +7.12]	+6.53 [+6.02, +7.03]
Disc − Pred+readout	+4.08 [+3.72, +4.44]	+4.08 [+3.63, +4.50]	+4.08 [+3.69, +4.47]
Pred-verify-only (seed 0) − Pred+readout	+0.03 [−0.23, +0.27]	+0.03 [−0.22, +0.29]	+0.03 [−0.26, +0.30]

Table 14 is the exclusion-regime sensitivity cited in §3.

Table 14: The main endpoint under all three exclusion regimes. The wider audit raises the predictive verifier and leaves the discriminative verifier flat, narrowing the gap by 0.36 points; the conclusion is unchanged across a 0.66-point spread. We keep the pre-registered regime as primary.

exclusion regime	predictive	discriminative	gap
none	.9144	.9827	+6.83
398 rows (pre-registered)	**.9182**	**.9835**	**+6.53**
933 rows (both rulers, every applicable tier)	.9216	.9833	+6.17

Figure 4 and Table 15 are the zero-shot transfer in §6.

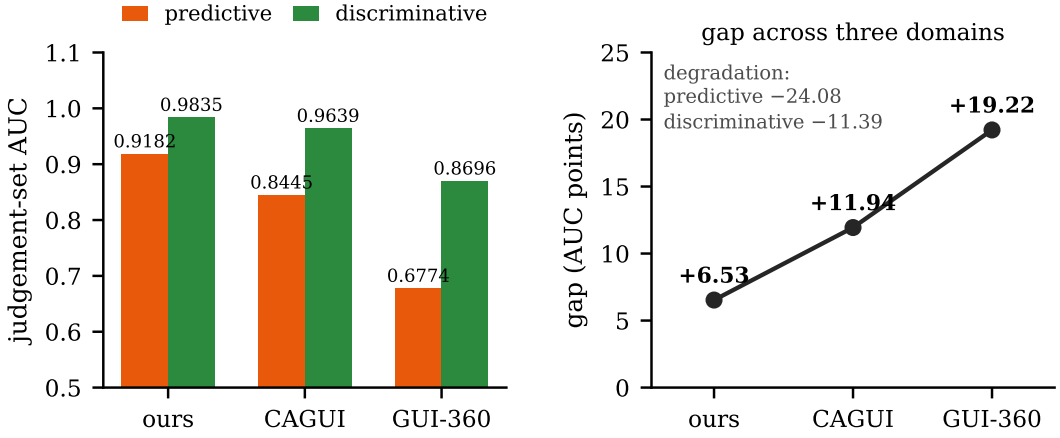

Figure 4: Zero-shot transfer to two external corpora, all rows under one audit regime. Left: both verifiers degrade, the predictive one far more. Right: the gap grows monotonically across the three domains, nearly threefold. Domains are mobile (in-distribution), Chinese mobile, and Windows desktop.

Table 15: Zero-shot transfer. All rows exclude the rows each corpus's own audit flags (398 / 278 / 18,832).

corpus	domain	predictive	discriminative	gap	degradation (pred / disc)
ours	mobile, in-distribution	.9182	.9835	**+6.53**	—
CAGUI	Chinese mobile, external	.8445	.9639	**+11.94**	−7.37 / −1.96
GUI-360	Windows desktop, external	.6774	.8696	**+19.22**	**−24.08 / −11.39**

Figure 5 breaks that transfer down by tier.

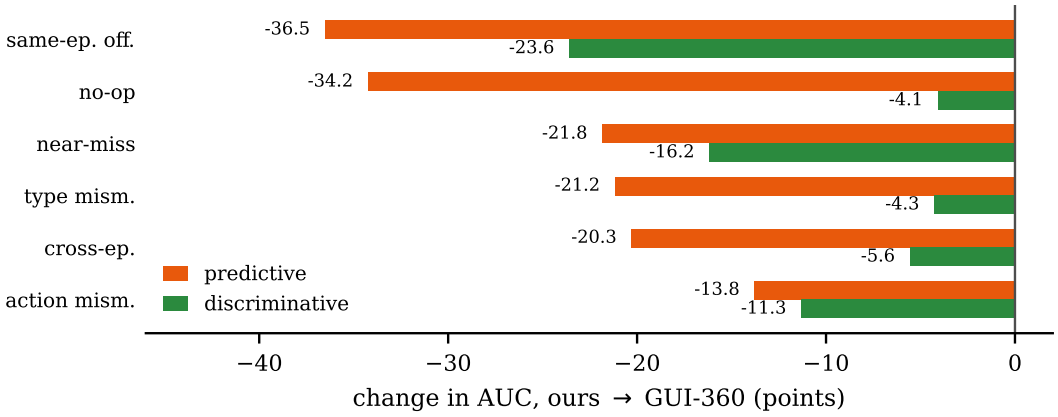

Figure 5: Per-tier change from our benchmark to GUI-360, both verifiers, 2B. The predictive verifier's two largest losses are no-op and same-episode offset; the third frame-side tier, cross-episode, sits with the action-side tiers. This is what bounds the claim in §6: the degradation is concentrated in two tiers, not in the frame-side group as a whole. Sources: Table 6 (2B rows) and the per-tier GUI-360 scores.

Table 16 is the zero-training reference cited for the no-op tier in §6.

Table 16: Two rules that train nothing, on the judgement set. `exact` scores 1 unless s_{t+1} is byte-identical to s_t; `pixdiff` is the mean absolute difference at 256 px grayscale—the same ruler as audit criterion A—oriented so that *more* change reads as positive. We keep that orientation rather than reporting $\max(a, 1-a)$, because a value below .5 is itself the finding: in the two tiers that replace the frame, negatives differ from s_t more than positives do. The last block runs *Pred's own scoring function* on an untrained adapter, which is the fairest zero-training reference available—identical read-out, no prompt chosen for it. Only `fwd` and `mink` are reportable there; the contrastive and verification heads are randomly initialised at that point, so their scores mean nothing. `exact` reads .9978 rather than 1.000 for a reason worth recording: 13 positives (0.45%) are themselves byte-identical—a real action that moved no pixel—and **no audit criterion covers positives**, only negatives.

rule	act. mism.	near-miss	type mism.	cross-ep.	no-op	same-ep. off.	all
`exact`	.4995	.4990	.5011	.4978	**.9978**	.4983	.6116
`pixdiff`	.5662	.4908	.4892	**.1844**	**.9978**	.4123	.5267
*Pred's own read-out on an **untrained** adapter (LoRA B is zero-init, so this is the frozen backbone)*							
`fwd`	.4445	.4965	.4933	.8719	**.0162**	.6422	.4986
`mink`	.4840	.4665	.5015	.7313	**.0170**	.6005	.4591
`exact`, no-op tier removed							.4988

Table 17 is the GUI-Robust comparison cited in §7.

Table 17: Real interface anomalies (GUI-Robust (Yang et al., 2025), Apache-2.0). Both pools are *recorded human transitions* from the same task sessions; they differ only in whether the successor screen is an anomaly, so we registered 0.50 as the target—a line stronger than correctness requires, since a verifier can separate two pools it accepts equally, which is why the accept-rate block is the deployment-relevant half of the table. Anomaly successors change the screen more (median .273 against .104), so the pre-registered primary endpoint is stratified by change magnitude—every stratum separates the pools, and the pooled value is *lower* than every stratum (Simpson's paradox), which is why we report both. Intervals resample *tasks*, not steps. Per-type rows are descriptive: n is 7–20 and no interval is quoted. One further pair is excluded because its two steps are not consecutive ($3.0 \rightarrow 4.1$: the clean frame of step 4 is absent, so whether the action at step 3 produced that anomaly screen cannot be established from the data). The interpretation thresholds were registered before these checkpoints were scored.

	Pred	Disc
pooled	.576 [.483, .666]	.712 [.627, .787]
change-stratified (primary)	**.683** [.588, .777]	**.759** [.662, .839]
accept-rate drop at a threshold accepting 90% of ordinary successors		
within change strata	**+.322** [+.106, +.438]	**+.377** [+.167, +.544]
pooled	−.025 [−.084, +.145]	+.290 [+.116, +.425]
by anomaly type (descriptive; n anomaly steps)		
no network (20)	.732	**.891**
login wall (20)	.501	.514
ad / pop-up (11)	.541	.789
captcha (7)	.766	**.855**
page loading (7)	.438	.676
cookie banner (7)	.345	.541

E Positives from failed trajectories

Every positive in the benchmark is a step that worked, so "positive" covaries with task success. GUI-360 releases 62,236 failed trajectories; a failed action still produces the successor it produced, so those steps are valid L1 *positives* whose trajectory did not succeed. Scoring 5,591 success-trajectory and 3,666 failure-trajectory positives from the same corpus and application in one pass (a single pass matters: our predictive signal is a per-file z-scored mixture, so scoring the pools separately would centre each and erase the contrast), a verifier of dynamics should read 0.50.

Failed steps change the screen far less (median .00186 against .00635), and that confound dominates: pooled, Disc reads .6154 [.6036, .6274], which would cross our pre-registered .60 line, but .669 **of it comes from the single lowest-change stratum and the other four sit at** .477**–**.558. Stratified by change magnitude—the pre-registered primary—both verifiers are weak: .537 (Pred) and .546 (Disc), which the registration reads as "report the numbers, make no qualitative claim". Matching scenario composition instead of change magnitude (the failure pool contains no `search` rows while the success pool is 96% `search`) *raises* the separation rather than explaining it (.543 / .683, n_{success}=198, intervals only), so the residual is not a scenario classifier; we have not run the two controls jointly. The interpretation thresholds were registered before these checkpoints were scored.

F Where the learned comparison splits

Table 18 is the in-distribution split of the learned comparison stage cited in §5, repeated on the two transfer corpora. It is independent of the failed-trajectory analysis above.

Table 18: Pred+readout decomposed, by reading two signals off two checkpoints. The *objective* column is Pred+readout's backbone read with Pred's frozen signal, minus Pred ($+1.37$ $[+1.14, +1.62]$ in-distribution); the *readout* column is the learned comparator against that same frozen signal *on the same checkpoint* ($+1.08$ $[+0.84, +1.31]$), so the two differ in exactly one field each. In-distribution they are worth 21.0% and 16.5% of the gap, together under two fifths. Each column is estimated by its own bootstrap, so rows need not sum exactly at two decimals.

corpus	verification objective	learned readout	total
ours	$+1.37$	$+1.08$	$+2.45$
CAGUI	$+1.69$	$+2.78$	$+4.47$
GUI-360	$\mathbf{+0.23}$	$\mathbf{-2.12}$	$\mathbf{-1.90}$

G BLINDED AUDITS OF THE LABELS

The automated audit (Appendix A) compares pixels and local patches. It cannot decide whether two click points fall on the same interactive element, which is the way a constructed negative is most likely to fail. So one author hand-labelled a blinded sample, pre-registered before any row was opened (Table 19).

Protocol. Thirty rows per stratum (six negative tiers plus positives, 210 rows), drawn deterministically by hash from the judgement set. The labeller saw only the two frames and the action string—not the tier, the label, the split, the goal or the instruction, and no model's score—with rows shuffled across strata. Each row was called *compatible*, *incompatible* or *undecidable*, conditional on the action having executed; a separate flag records whether the label rests on that premise. **This is an author audit, not independent annotation**; the independent passes follow below.

One blinding leak and one post-hoc correction, both disclosed. A constructed no-op sets $s_{t+1} := s_t$, so those rows are identifiable from the blind sheet by their identical frame paths; the no-op tier was therefore not blind. Separately, the blind labelling could not fix the sign convention of `scroll up`/`down`. After unblinding we fixed it from the four *positive* scroll rows, whose labels are true by definition—all four moved the viewport down the page under `scroll down`—and applied the rule mechanically to all 43 scroll rows. That lowered the aggregate error from 20.6% to 16.1%, so it favours us; it also created one new error, which we kept. Both label sets will be released with the paper. Table 19 records the calls.

Table 19: Blinded manual audit, 30 rows per stratum. e is the share of constructed negatives judged compatible (for positives, judged incompatible); u undecidable; x the share whose label holds only if the action executed. Wilson 95% intervals are for the author audit; the last column gives the two independent human annotators' e. Near-miss, whose generator checks a coordinate tolerance but never element bounds, has the highest rate in every pass.

tier	n	e	95% CI	u / x	e, humans 1 / 2
positives	30	3.3%	[0.6, 16.7]	0% / 0%	3.3% / 0.0%
cross-episode	30	**0.0%**	[0.0, 11.4]	0% / 0%	0.0% / 0.0%
action mismatch	30	13.3%	[5.3, 29.7]	3% / 0%	6.7% / 0.0%
action-type mismatch	30	13.3%	[5.3, 29.7]	7% / 0%	10.0% / 3.3%
same-episode offset	30	16.7%	[7.3, 33.6]	0% / 0%	6.7% / 3.3%
no-op	30	20.0%	[9.5, 37.3]	0% / **80%**	20.0% / 13.3%
action near-miss	30	**33.3%**	[19.2, 51.2]	0% / 0%	23.3% / 23.3%
six negative tiers	180	**16.1%**	[11.5, 22.2]	2%	11.1% / 7.2%

Where the failures come from, and why that was predictable. The near-miss generator displaces a click by a ring $r \in [0.09, 0.18]$ of the screen, chosen to clear an expert-equivalence tolerance of 0.08 per axis. That tolerance is on *coordinates*. Nothing in the construction consults element bounds, so a displacement that is a different coordinate is not necessarily a different decision: a third of the sampled near-misses land inside the same full-width row, card or tab-bar segment

as the recorded click. The no-op tier fails differently—six of its thirty rows pair an unchanged screen with an action for which *no change is the correct outcome*: three `wait` actions, one `open_app` for an app already in the foreground, one tap on an already-selected filter chip, one tap on empty space below a list. And 80% of no-op rows carry a label that holds only if the action executed, which two offline frames cannot establish. Cross-episode, whose negatives come from a different session entirely, has no errors at all.

What it does to the gap. A direct recomputation on the audited rows is hopeless—forty-odd source triplets give a paired interval near ± 3.8 points—so we bound instead. Removing from each tier, across the full $15{,}465$ rows, the measured fraction e_t of rows *most favourable to our conclusion* (the negatives Disc ranks well below Pred) drops the gap from 6.53 to $\mathbf{3.86}$ points. That is the worst case by construction. Removing as many rows at random instead leaves 6.26 (SD 0.09 over 20 draws), so the error *rate* alone does not shrink the gap—only an adversarial assignment of it does. And the rows actually judged mislabelled are not adversarially placed: their mean favourability is -0.081 against -0.026 for the rest, i.e. if anything they lean the other way. We report 3.86 as the floor at the measured per-tier rates, and take the ordering, not the magnitude 6.53, as what the audit leaves standing.

Two conditions on those numbers. The audit allocates 30 rows to every tier, so 16.1% weights the six tiers equally. Weighting each tier's rate by its share of the judgement set's negatives gives 15.0%, with a 95% stratified bootstrap interval of $[10.1, 20.2]$ (B=10,000); cross-episode, the one tier with no errors, is also the largest. The 3.86 floor conditions on the per-tier point estimates. Replacing every tier's rate, positives included, by its Wilson upper bound and still removing the most favourable rows removes $4{,}346$ rows and leaves the gap at $+\mathbf{1.77}$ (Pred .9588, Disc .9765): smaller, and still positive. Both analyses were added after the audit and were not pre-registered. Drawing each tier's rate, positives included, from its Jeffreys posterior $\mathrm{Beta}(k{+}\frac{1}{2},\, n{-}k{+}\frac{1}{2})$ and removing that many rows adversarially gives $+3.73$ $[+2.96, +4.43]$ over $2{,}000$ draws (minimum $+1.88$); removing them at random gives $+6.28$ $[+6.02, +6.50]$.

Two independent human annotators. Two annotators from outside the project labelled the same 210 rows from the same blind sheets and label definitions, seeing no tier, label, earlier pass or model score. Their six-tier rates are 11.1% $[7.3, 16.5]$ and 7.2% $[4.3, 12.0]$ (10.6% and 7.0% weighted by tier size); both put near-miss highest (23.3%), no-op next (20.0% and 13.3%) and cross-episode at zero, the same shape as the author audit (Table 19). They agree with each other on 87% of rows (κ=.69; κ=.49 on the 180 negatives) and with the author on 94% and 86% (κ=.86 and .66); most disagreements are over whether a row is undecidable. Only 5.6% $[3.0, 9.9]$ of negatives are judged compatible by both. On the 142 negatives both judge incompatible the gap is $+6.60$ $[+4.37, +9.09]$, and $+6.70$ on the 136 all four passes judge incompatible. Removing each annotator's error rates adversarially leaves $+4.46$ and $+5.54$, and $+4.35$ $[+3.56, +4.98]$ and $+5.15$ $[+4.40, +5.72]$ with each rate drawn from its posterior.

A language-model pass. The same 210 rows were labelled again by a language model (Claude), shown only the blind sheets, the three label definitions and the scroll-direction convention fixed in round one, and none of the first round's labels, tiers or model scores. It is a second pass, not independent human annotation. The two passes agree on 192 of 210 rows (91%, Cohen's κ=.80 over C/I/U; κ=.67 on the 180 negatives). The second pass puts the six-tier rate at 13.3% $[9.1, 19.1]$, with near-miss again the highest (23.3%), no-op identical (20.0%) and cross-episode again 0%; of its 18 disagreements with round one, 12 involve an undecidable call on one side. Removing the second pass's rates adversarially leaves $+4.24$. On the 143 negatives both passes judge incompatible, scored against all $2{,}898$ positives, the gap is $+7.08$ $[+4.73, +9.89]$; on the 29 negatives round one judged compatible it is $+7.62$ $[+1.73, +15.25]$, so Disc also ranks the mislabelled negatives lower, consistent with the typicality reading of §7.

The pre-registration also required, for any tier with $e_t{>}15\%$, the gap after dropping that tier whole. Dropping near-miss leaves $+\mathbf{6.59}$ (Pred .9250, Disc .9910); dropping no-op leaves $+6.02$; dropping same-episode offset, the tier where Disc's lead is largest, leaves $+5.43$. Dropping all three together leaves $+3.79$ on $8{,}406$ rows. The semantically worst construction is therefore not what holds the headline up; shrinking the gap to the 3.8–3.9 band takes either adversarial placement of the error rate or removing the two large frame-side tiers alongside near-miss.

H A PROMPTED MODEL AS A ZERO-SHOT REFERENCE

The zero-shot reference is `gpt-5.6-terra`. Table 20 scores it on the same 1,998 rows as the trained routes.

Table 20: Zero-shot `gpt-5.6-terra` against the trained routes, on the **identical** 1,998 rows it completed (every route restricted to them). The judge is zero-shot and the trained routes saw 1.28M training samples, so this is not a matched comparison; it answers what the benchmark looks like to a strong model that has never seen how its negatives were built. **It lands below every trained route**, and its weakest tier is no-op—the tier a byte-identity rule solves at .9978.

	ALL	act. mism.	near-miss	act. type	cross-ep.	no-op	same-ep. off.
gpt-5.6-terra (zero-shot)	.8822	.8486	.8312	.9641	.9914	**.7555**	.9111
Pred	.9181	.8767	.8609	.9411	.9876	.9221	.8840
Pred+readout	.9407	.9022	.8892	.9779	.9821	.9697	.8988
Disc-late (seed 0)	.9364	.8903	.8811	.9853	.9314	.9674	.9448
Disc-late-grid (seed 0)	.9409	.8923	.8685	.9857	.9370	.9881	.9441
Disc-grid	.9850	.9857	.9409	.9963	.9975	.9937	.9833
Disc	.9847	.9839	.9283	.9944	.9989	.9985	.9849

Protocol. Table 20 puts it beside the trained models. The judge sees the two frames and the action, with the action point drawn on the before-frame, and returns an integer 0–100; the episode goal is withheld, since the recorded instruction describes the true action and would hand a text-only judge the answer on the action tiers. The prompt, the scored subset (200 positive sources per held-out split and all their sibling negatives), and what counts as an acceptable run were registered before any call was made.

What the run actually was. Of 2,146 selected rows, 1,998 were scored: the endpoint terminated the run with HTTP 403 after 1,999 calls, and the loss is even across tiers (6.5%–7.7% each), so the missing rows are a content-independent tail rather than a selected subset. One further row was dropped because its reply was a scoring rubric for an unrelated task, which the audit of regex-fallback parses caught; it was the only fallback parse in the run. Calls were made through a third-party relay, so every response's self-reported identity was recorded: all 1,979 returned the requested model, no request field was rejected, and 44.2% of calls reported reasoning tokens, concentrated on the tiers that require reading the action (56.9% against 29.7% on cross-episode and no-op).

One hosted call is not a fixed reference point. The same model name, `gpt-5.6-terra`, the same prompt and the same reasoning effort, called two months earlier, scores 5.04 AUC points *higher* on the 1,789 rows the two runs share (.9402 against .8898), agreeing exactly on only 41% of them and differing by 50 points or more on 12%. The drift is not uniform: no-op moves by 22 points on average while cross-episode barely moves. We cannot separate model version drift from run-to-run variation, because the earlier rows cannot be rescored under the earlier endpoint. We therefore report one endpoint on one day and *do not* merge the two runs, even though the earlier one covers 131 of the 147 rows the later one lost. Anyone using a hosted frontier model as a baseline inherits this: the number is a property of a model, a prompt, and a date.

I PER-SEED RESULTS AND FURTHER ANALYSES

The three-corpus decomposition, seed by seed. Table 3 reports two-seed means. The seeds behind them: on CAGUI, Disc gives up $+7.51$ $[+7.08, +7.95]$ and $+6.89$ $[+6.44, +7.36]$ to Disc-late, and the remainder against Pred+readout is -0.03 $[-0.49, +0.45]$ and $+0.59$ $[+0.10, +1.06]$—the two seeds fall either side of zero, which is why the main text reads the term as small rather than as absent. On GUI-360 the late-comparison term is $+19.30$ $[+18.96, +19.65]$ and $+18.90$ $[+18.56, +19.24]$, and the remainder $+1.82$ and $+2.22$. Every interval is the same paired bootstrap over source triplets, computed within each corpus's own audit exclusions.

The grid variant, seed by seed. Disc-late-grid against Disc-late, seed-paired: on CAGUI $+1.25$ $[+0.87, +1.61]$ and $+0.83$ $[+0.44, +1.21]$; on GUI-360 $+5.05$ $[+4.74, +5.35]$ and $+3.06$ $[+2.75, +3.39]$. Disc against it: on CAGUI $+6.26$ $[+5.88, +6.64]$ and $+6.06$ $[+5.64, +6.48]$; on GUI-360 $+14.25$ $[+13.98, +14.53]$ and $+15.84$ $[+15.55, +16.12]$. On desktop the grid gain is again frame-side, larger than in-distribution: no-op $+21.37$ and $+14.94$, cross-episode $+9.07$ and $+8.60$. Seed 1 also loses 7.26 points on action-type mismatch there while seed 0 does not move. In-distribution, Disc's lead over Disc-late-grid is 7.6–8.9 points on action mismatch, 5.5–5.8 on same-episode offset and on cross-episode, 3.9–4.7 on near-miss, and under one point on no-op and action-type mismatch; the grid gain is largest on no-op ($+2.23$, $+2.08$) and cross-episode ($+1.03$, $+1.13$), and on action mismatch (-0.73, $+0.25$) and near-miss (-0.17, $+1.35$) it changes sign between seeds. Disc-grid against Disc-late-grid's seed 0 has the same shape: $+9.04$ on action mismatch, $+5.64$ on near-miss, $+5.71$ on cross-episode, $+5.62$ on same-episode offset, and under one point on no-op ($+0.60$) and action-type mismatch ($+0.59$). Against Disc, Disc-grid differs by at most 0.90 points on any tier. Off distribution, Disc-grid against Disc-late-grid's two seeds is $+6.07$ and $+5.88$ on CAGUI and $+15.20$ and $+16.78$ on GUI-360; Disc leads Disc-grid by $+0.19$ $[+0.00, +0.36]$ on CAGUI and trails it by 0.95 $[-1.13, -0.77]$ on GUI-360, the one place where the grid comparator is better than Disc's MLP head. Disc-late-grid's aggregate AUC is .9014 and .9033 on CAGUI and .7270 and .7112 on GUI-360.

A leaked training pool, and what it was worth. An external review found the training set contained the benchmark: the dataset class never read the split field, so all 4,398 benchmark sources sat in the training pool, each seen roughly 6.8 times. Each negative tier also had a matching loss on the same triplets, including an action-contrastive loss that imported *the benchmark's own perturbation function*. One earlier observation must be read in reverse: the three splits scoring almost identically was evidence of leakage, not of generalisation. The fix is a split whitelist plus id-level exclusion through one fail-closed path with a contract test. Retraining addresses the model having seen the test set, not *our* having seen it, which is why every experiment carries a pre-registration written before any of its results exist (§3).

The leaked and clean trainings differ only in that whitelist and exclusion; data, negatives, budget, backbone, rank, seed and loss weights are field-identical. Subtracting them estimates what contamination was worth to each architecture: $+\mathbf{3.42}$ points $[+3.08, +3.78]$ to Pred and $+\mathbf{0.73}$ $[+0.58, +0.90]$ to Disc, on disjoint intervals. We pre-registered the *ordering* and predicted it backwards. On the contaminated split the gap measured $+3.84$; clean, $+6.53$. A leaked benchmark understates the cost of the predictive path, here by a factor of 1.7, but does not change the winner: Disc leads on both splits. This comparison is a before/after of a bugfix, not a designed experiment, which is why it is not in the main text.

The learned readout absorbs the action-string prior. Blinding the frames (§4) leaves only the action string. On the action-type-mismatch tier the two readout variants score .8228 and .8186 there, against .6583 for the hand-designed readout and .5876 for Disc—close to the .81–.83 ceiling of an action-only prior fitted in-distribution (Table 5). The learned readout leans on this benchmark's action-string prior harder than either variant without one, and on that tier its entire in-distribution gain over the hand-designed readout ($+2.80$) is available from the action text alone ($+16.45$ blinded). Both readout variants do this identically, so it is a property of adding a learned comparator, not of removing the alignment losses. Set against that, the readout's gain is otherwise action-free: $+2.48$ normally and $+2.39$ with the action blinded, so what it mostly buys is a better frame comparison rather than better action grounding.

J TRAINING CONFIGURATIONS

All models are LoRA fine-tunes (Hu et al., 2022) of Qwen3-VL (Bai et al., 2025) (Instruct checkpoints), share a sample budget of 1.28M (80,000 steps at effective batch 16; 160,000 at 8 for 4B and 8B), and differ only in the fields named below. Learning rate 1×10^{-4}, weight decay 0.05, 2,000 warmup steps, adapter rank 16, 512 tokens per frame on a 23×23 patch grid at 736 px. Loss weights for the predictive verifier: forward 1.0, contrastive 1.0, localisation 0.5, action 1.0, no-op 1.0, change-mask $\lambda = 3.0$, min-k fraction 0.05. Pred-verify-only is Pred+readout with forward, contrastive, action and no-op set to 0 and verification 1.0. Disc-late copies Disc field for

field and changes two things: the two frames are encoded in separate passes, and the head therefore reads a concatenation of two pooled encodings, $\mathrm{Linear}(2D, D)$-GELU-$\mathrm{Linear}(D, 1)$ against Disc's $\mathrm{Linear}(D, D)$-GELU-$\mathrm{Linear}(D, 1)$. Neither side is detached, so both frames pass through the trainable encoder as they do in Disc; image tokens per step and peak memory match Disc to within 0.5 GB. Disc-late-grid copies Disc-late and changes only what the head reads: both 529-token grids instead of one pooled vector per frame, fed to the comparator family of Pred+readout described below (the successor grid attends to the s_t grid, with mean and min-5% cosine as explicit features), with *neither* grid detached. The head has 1.51M parameters against Disc's 4.20M and Disc-late's 8.39M. It ran on four devices at per-device batch 4 rather than two at 8, keeping effective batch 16 and eight negatives per step; a 600-step check before the full run found the no-op, action and hard-negative proportions within sampling error of Disc-late's. Disc-grid keeps Disc's joint forward pass byte for byte (same prompt and packing) and replaces the MLP on the last pooled token with Disc-late-grid's comparator, applied to the two frames' 529-token runs inside the joint sequence; its head and trainable-parameter count equal Disc-late-grid's. It ran on eight devices at per-device batch 2 (effective batch 16), which leaves one negative per device per step; the logged mix on one device was .30 action, .44 no-op and .26 hard negatives, against .38, .33 and .29 for Disc-late-grid. Action negatives are drawn with probability 0.5 and same-episode hard negatives with probability 0.5. Optimiser AdamW (Loshchilov & Hutter, 2019).

The eight read-outs. A predictive variant emits a per-patch cosine profile, not a score, so a read-out has to be chosen. The eight candidates are three base signals—$\texttt{fwd}$, the mean cosine over the 23×23 grid; $\texttt{mink}$, the mean of the worst 5% of patches; and $\texttt{nce}$, the pooled contrastive head's similarity—and five fixed z-scored mixtures of them: $(\texttt{fwd} + \texttt{nce})/2$, $(\texttt{fwd} + \texttt{mink} + \texttt{nce})/3$, $(\texttt{mink} + \texttt{nce})/2$, $(\texttt{fwd} + \texttt{mink})/2$ and $(2\,\texttt{nce} + \texttt{mink})/3$. On the in-distribution selection split ($8{,}163$ rows) the last wins for every predictive variant we trained, and the z-statistics used to form it are estimated on that split and then frozen. Disc has no such choice: its head emits one logit.

Heads, temperature and EMA scope. The contrastive head projects the pooled prediction and target to 256 dimensions and uses temperature $\tau{=}0.07$ against the in-batch bank. The learned comparator of Pred+readout projects both patch grids to 256 dimensions, lets the candidate grid attend to the predicted one (4 heads, residual and LayerNorm), and feeds $[\,\text{attended mean},\ \text{predicted mean},\ \text{candidate mean},\ \text{mean cosine},\ \text{min-}5\%\ \text{cosine}\,]$ to a two-layer MLP of width 256; the two hand-designed cosines enter as explicit features, which is what makes the learned read-out contain them as special cases. The candidate grid is detached. Under Pred-EMA the moving average is taken over the *adapter* parameters only—the backbone is shared and frozen—updated once per optimisation step as $\theta_{\mathrm{ema}} \leftarrow m\theta_{\mathrm{ema}} + (1-m)\theta$, and the target pass runs with that adapter selected instead of with the adapter off. The two predictor-side anti-collapse terms of §5 enter at weight 1.0, the patch InfoNCE at temperature 0.1; of the two target-side repairs, the patch-covariance penalty enters at the same weight and DINO-style centering carries no weight, being an operation on the target's forward pass rather than a loss.

The predictive verifier carries 27.9M trainable parameters against the discriminative verifier's 21.6M, the difference being auxiliary contrastive and change-localisation heads. We report this asymmetry without asserting its sign: more trainable parameters and more objectives are not automatically an advantage. Both verifiers are field-identical in data, negatives, budget, backbone and seed.

Equal sample budget *is* equal wall-clock here, and the compute asymmetry runs against the verifier we favour. Equal samples need not mean equal compute, so we report the measured quantities rather than assuming. Over the $80{,}000$ logged steps of the five 2B runs trained on identical hardware ($2\times$A800-80GB, effective batch 16), median throughput is 5.59 examples/s for Pred, 5.61 for Disc, 5.63 for Pred-EMA, 5.88 for Pred+readout and 5.74 for Pred-verify-only—a spread of 5%, so these five consume the same wall-clock for the same 1.28M samples. **This also settles by measurement the claim that Pred+readout costs no extra per step**: it is the fastest of the five, because its verification head consumes tensors the predictive pass has already produced.

Counting image tokens per optimisation step at batch B and 512 tokens per frame, the two verifiers are *not* matched, and not in the direction one would guess. Disc runs $1.5B$ sequences (the positives plus $B/2$ negatives) of two frames each, $\approx 1536B$ image tokens, all with gradients. Pred runs $2B$

Table 21: Variants, named by what they change. The last column is each variant's run identifier, as used in the pre-registrations.

name	change from Pred	run id
Pred / Pred-4B	— (2B / 4B)	C1 / C3
Disc / Disc-4B	discriminative objective	C2 / C4
Pred-8B	8B backbone	C5
Pred+readout	+ learned verification head	C6
Pred-verify-only	Pred+readout, alignment losses off	C9
Pred+near-miss	near-miss negatives at p=0.5	C7
Pred-EMA	EMA target (m=0.996)	C8
Pred-EMA+readout	EMA + verification head	C8b
Pred-EMA (m=0.999 / 0.9999)	slower EMA momentum	C8_m999 / C8_m9999
Disc-late	Disc with the frames encoded separately	C12a / C12b
Disc-late-grid	Disc-late keeping both patch grids	C13a / C13b
Disc-grid	Disc read by Disc-late-grid's comparator	C14
Pred-EMA+patch InfoNCE	EMA + patch InfoNCE, target-side negatives	C10a
Pred-EMA+variance floor	EMA + per-patch variance floor	C10b
Pred+patch InfoNCE	frozen target + patch InfoNCE (control)	C10c
Pred-EMA+centering	EMA + centering of the target pass	C11a
Pred-EMA+covariance	EMA + patch covariance penalty	C11b

target encodings plus up to B distractor encodings without gradients and $1.5B$ predictor sequences with them, $\approx 2304B$ image tokens of which $768B$ carry gradients. **Pred therefore performs more forward work and less backward work per step**, which is why the wall-clock matches. Peak device memory differs substantially and favours neither interpretation cleanly: 71.3 GB for Disc against 41.1 GB for Pred, because Disc packs both frames into one sequence. Inference cost is the reverse of training in *pass count*: Disc needs one pass over the pair, Pred needs two (predict, then encode the successor). We did not measure deployment latency, memory or throughput, so we make no latency claim from this—and two considerations cut against the naive reading. The predict pass depends only on (s_t, a), so it can be issued while the action executes and need not sit on the critical path; and its compensating advantage, scoring *before* execution, is the subject of §6.

K Pre-execution scoring and decoding probes

The pre-execution scorer ranks candidate actions from (s_t, a, goal) alone. Against an ablation receiving no prediction it gains $+18.67$, $+25.31$ and $+28.03$ top-1 points at $N = 4, 8, 16$; but in that ablation the state pass never sees the action, which arrives as a frozen embedding at the head, and candidate ranking is precisely a test of action–image interaction. The isolating variant keeps the action in the backbone prompt and removes only the instruction to predict the next screen. Against it the predictive variant scores -1.16, -2.49, -3.13.

Four reasons keep the claim narrow. The two variants are architecturally identical, differing in one prompt sentence and in whether the scored tensor is also aligned to the encoded successor. The proposition "the predicted state carries value" is not falsifiable as posed. Supervision is asymmetric: the isolating variant carries only the ranking loss while the predictive variant carries four additional objectives, so multi-objective competition is as consistent with $\Delta < 0$ as our reading. And it is a single seed.

The decoding probes ask how much future information the constraint puts into the representation at all, comparing the same model's two internal pathways—so no seed or training confound—on three targets: how much of the screen changes, whether it changes, and where. Capture of the observable headroom is 14.8%, 11.9% and 10.6%, all short of the 20% line fixed in advance; the localisation interval excludes zero, so the increment is real but roughly one tenth of what observation provides. On "whether the screen changes" the untrained frozen backbone scores .8193 against our trained representation's .8082—on that target our training adds nothing over the backbone we started from.

A related probe explains why frozen-feature comparators cannot settle the readout question of §6: a low-dimensional projection of frozen features cannot re-learn, from sixteen thousand rows, a con-

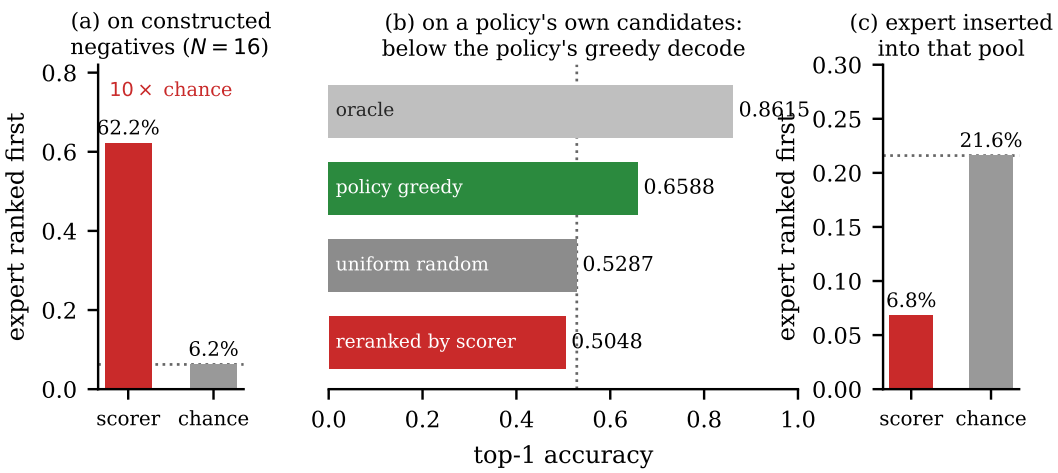

Figure 6: The same scorer, the same checkpoint, three candidate sources; bar heights are not comparable across panels because the pools differ in size. **(a)** On constructed negatives—the expert plus fifteen perturbations—it ranks the expert first ten times as often as chance. **(b)** On a behaviour-cloned policy's own samples it falls 15 points below the policy's greedy decode, although the pool contains a correct candidate 86% of the time. The comparison to uniform selection reverses under a different de-duplication rule and we do not rely on it (Appendix L). **(c)** With the expert action inserted into that pool it ranks first 6.8% of the time against 21.6% for chance.

trastive head that was trained end-to-end on 1.28M samples. Two such probes failed before we trained the comparator end-to-end instead.

L BEST-OF-N: DE-DUPLICATION AND POOL SIZE

The pre-execution scorer is evaluated by ranking a policy's sampled candidates, and two conventions in that setup move the numbers: how the pool is de-duplicated, and how the random baseline is computed. We report both sensitivities because one of them reverses a sign.

De-duplication. Our protocol de-duplicates by the action string. An alternative—arguably the more meaningful one—is to de-duplicate by *action equivalence* at the same coordinate tolerance the labels use, so that two samples that would be the same decision count once. Table 22 runs both. **The scorer's top-1 barely moves; the random baseline does**, because the policy's repeated samples of a decision it is confident about survive string de-duplication as distinct entries and are disproportionately correct. Under equivalence de-duplication the scorer is *above* random at every tolerance.

Table 22: Best-of-N under two de-duplication rules. Paired bootstrap over episodes, B=2,000, seed 0. The policy's greedy decode is .6588 throughout, and the scorer is 15 points below it under either rule—that comparison is unaffected. The random column and the interval beside it are the same estimator (the expected accuracy of a uniform pick, below).

tol.	rule	mean pool	scorer top-1	random	scorer − random
0.04	string	6.61	.4777	.4942	−1.65 [−2.83, −0.52]
0.04	equivalence	3.20	.4744	.4495	**+2.49** [+1.35, +3.63]
0.08	string	6.61	.5048	.5287	−2.38 [−3.63, −1.21]
0.08	equivalence	2.88	.5029	.4811	**+2.18** [+0.99, +3.40]
0.14	string	6.61	.5309	.5562	−2.54 [−3.75, −1.39]
0.14	equivalence	2.66	.5298	.5111	**+1.87** [+0.64, +3.03]

The random baseline. The metric is whether the *selected* candidate is correct, and a pool may contain several correct candidates, so chance is the per-step mean of $K/|\text{pool}|$ with K the number of pool entries within the label tolerance—not $1/|\text{pool}|$, which would assert that exactly one candidate is correct and would read 1.0 on the 637 steps whose pool has a single entry, where the true accuracy is $.9309$. Nor is it the reciprocal of the mean pool size, which exchanges a mean with a reciprocal and differs a great deal here ($.5287$ against $.1512$) because pool sizes are skewed.

Pool size. Stratifying by pool size under the string rule, the shortfall against random is concentrated in large pools (-5 to -9 points for pools of 11–16) and absent in small ones. **637 of the 3,611 steps** (18%) **have a pool of one**, where scorer and baseline are identical by construction at $.9309$; those steps raise both numbers and should be kept in mind when reading the pooled figures.

What we could not test. Reporting the scorer against a pool with *no* de-duplication would need the multiplicity of each sampled candidate, and the candidate pools we stored keep only the de-duplicated list. Without de-duplication the random baseline becomes the policy's own sampling distribution rather than a uniform pick, and we *expect* that to be the more accurate of the two, because the string-versus-equivalence rows above are exactly that effect measured: collapsing repeated samples lowers the random baseline by 4.8 points, which is only possible if the repeated samples are disproportionately correct. That is an expectation supported by an adjacent measurement, not a result—without the multiplicities we can neither compute the number nor guarantee its sign.

M THE BUDGET TRAJECTORY

Would twice the training budget close the gap? Because both verifiers save a checkpoint every 10,000 steps, this is measurable without training anything further. Scoring each checkpoint under the main-text protocol gives Figure 7. Both verifiers are still improving at the end of training—the predictive verifier from $.8915$ to $.9182$, the discriminative verifier from $.9706$ to $.9835$—but the *gap* between them is not: successive 20,000-step windows close it by 0.73, then 0.57, then 0.08 points. **What we measured is that deceleration, within the budget we ran**; carrying it forward arithmetically would put a doubled budget near 6.2–6.4, but that is an extrapolation, not a substitute for the experiment, and we did not run the experiment.

This is a limitation we measured rather than declined to test, and it carries three caveats: it rests on one pair of verifiers at one seed; the extrapolation follows the deceleration of the last three windows rather than a fitted curve; and a formal doubling would have to double *both* verifiers, since doubling only the predictive verifier would abandon the equal-budget comparison the paper depends on.

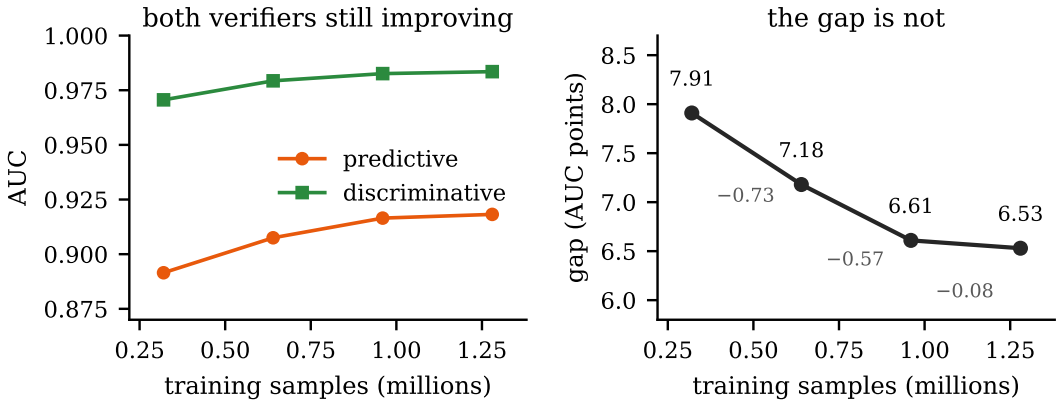

Figure 7: The gap is established early and then stops closing. Both verifiers improve monotonically with budget (left), but successive 20,000-step windows narrow the gap by 0.73, 0.57 and finally 0.08 points (right).

N THE FORWARD LOSS, AND WHY ITS GEOMETRY ARGUMENT IS NOT A JOINT OPTIMISER

Writing $\hat{z} = f_\theta(s_t, a)$ and $z = g_\phi(s_{t+1})$ over N patches,

$$\mathcal{L}_{\text{fwd}} = 1 - \frac{\sum_i w_i \cos(\hat{z}_i, z_i)}{\sum_i w_i}, \qquad w_i = 1 + \lambda m_i, \tag{1}$$

with m_i the change mask and $\lambda = 3$ throughout.

Two caveats on the argument in §5. First, under an EMA ϕ receives *no* gradient from Equation 1: the route is indirect, because the online encoder both produces $\hat{z}$ and, after averaging, supplies z. The defensible claim is therefore about the objective's geometry—with no target-side negatives, nothing in the objective keeps the similarity of two different states *away from* 1, equivalently there is no lower bound on the distance between them—and not about a joint optimiser over (θ, ϕ). Second, that same asymmetry is why the two *predictor-side* repairs of §5 act on $\hat{z}$ rather than on the target: there is no gradient path to the EMA copy, so the only available pressure is on the online weights the EMA will next copy. A non-gradient operation on the target's own forward pass is not ruled out by this argument, which is why §5 tests one—DINO-style centering—and finds that it scores below the collapsed model it was meant to repair.

O ALTERNATIVES TO THE FROZEN TARGET

Table 23: EMA momentum sweep, each variant field-identical to Pred-EMA except the momentum. The three statistics disagree: at m=0.999 the cosine recovers toward the frozen reference while effective rank and AUC are the worst measured. Half-lives are $\ln 2/-\ln m$ steps. Intervals are paired bootstraps over source triplets on the *unrounded* scores, so the last column can differ by 0.01 from the difference of the two rounded AUC columns (the m=0.9999 row).

momentum	half-life	pairwise cos.	eff. rank	judgement AUC	vs. Pred
m=1 (frozen, Pred)	—	.8173	**90.6**	**.9182**	—
0.996 (Pred-EMA)	173	**.9405**	74.0	.7766	$-14.16\ [-14.78, -13.56]$
0.999	693	.8386	**59.4**	**.7600**	$-15.82\ [-16.41, -15.22]$
0.9999	6,931	.8191	65.0	.8879	$-3.04\ [-3.54, -2.54]$

Two further variants, each field-identical to Pred-EMA except for the EMA momentum, trained for the same 80,000 steps. Together with Pred-EMA they give three momenta spanning a factor of forty in half-life. Table 23 carries the result; §5 reports it in prose and this section records its protocol and checks, followed by the four repairs.

Why the sweep had to run at full length. The cheap version—ten thousand steps per momentum—would have produced a false negative in our own favour. The adapter's second matrix is zero-initialised, so an EMA that has not yet tracked the online network is approximately the frozen base. At m=0.9999 ten thousand steps is 1.4 half-lives, leaving 37% of the initialisation weight in place, and that variant would have read as "did not collapse" because it had barely moved. Both variants therefore ran the full budget, which is also the only budget-matched comparison against Pred-EMA.

Validity check: the EMA really departed. If a variant's target encoder were indistinguishable from the disabled-adapter baseline, the variant would be void rather than uncollapsed. On the trained checkpoints, relative displacement between the two encodings is 1.75 (Pred-EMA), 1.65 (m=0.999) and 1.07 (m=0.9999), with cosine against the frozen base of 0.105, 0.037 and 0.161. All three moved far from initialisation, and the ordering matches the tracking speeds.

Trajectories. Measuring every saved checkpoint, Pred-EMA stays near cosine .94–.97 throughout. The two slower variants instead *descend* toward the frozen reference—m=0.999 from .9613 at step 10,000 to .8386 at 80,000, m=0.9999 from .9195 to .8191—while their effective rank falls over the same span, from 78.9 to 59.4 and from 88.7 to 65.0. The mean cosine and the rank move in opposite directions, which is why we report both.

What the pre-registration got wrong, and what caught it. The judgement lines were two-sided: collapsed if cosine $\geq .90$ *and* rank ≤ 85, not collapsed if cosine $\leq .85$ *and* rank ≥ 88 *and* AUC $\geq .85$. Both new variants satisfy neither, landing in the intermediate band, because the cosine criterion says they are fine and the rank criterion says they are worse than Pred-EMA. Requiring both is what stopped us reading the recovered cosine as recovery: the m=0.999 variant has the sweep's best cosine trend and its worst AUC. Our stated expectation—that both would collapse—was wrong, and so was the argument behind it (§5).

The two predictor-side repairs, and their frozen control. On the collapsed model the InfoNCE term reaches AUC 0.7839 and the variance floor 0.7468, against Pred-EMA's 0.7766 and the 0.8474 recovery line: $+0.73$ $[+0.13, +1.39]$ and -2.98 $[-3.59, -2.34]$ against the $+7.08$ registered as recovery. Added to the *frozen*-target model instead, the same InfoNCE term reads 0.9116, or -0.66 points $[-0.98, -0.34]$ in-distribution and -3.05 on CAGUI, so it is not a generally useful component that we had been missing: it helps the collapsed model slightly and costs the healthy one.

The two target-side repairs. Both were registered with the conjunction rank ≥ 85 and AUC $\geq .8474$, the latter being Pred-EMA's .7766 plus the $+7.08$ recovery line used for the predictor-side repairs. *Centering* (C11a) reaches rank 23.7—the lowest of any variant here—and AUC 0.7572, which is below Pred-EMA's own .7766: removing the common mean direction from the target costs the task rather than restoring it. Its registered validity condition, that the centering buffer be substantially non-zero, holds ($\|\cdot\|_2 = 1199.9$ over the 529×2048 buffer). *The pre-registration also recorded, before the variant existed, that this variant's pairwise cosine would fall for a reason that is not evidence*, since centering removes exactly the common direction that the collapsed .9405 cosine is mostly made of; it fell to 0.002, and we do not count it. *Covariance* (C11b) reaches cosine 0.9881 with a spread of 0.008—the highest cosine and the tightest spread of any variant, past the collapsed reference—rank 76.9, and AUC 0.7867, and its no-op tier falls to chance ($.4913$), the same tier Pred-EMA destroys. Its forward loss ends at 0.162 against Pred's $\approx .53$: the prediction problem became easy, which is the signature we already report for Pred-EMA.

An instability we disclose rather than route around. The centering variant's no-op term diverges over its last thousand steps, from 0.67 at step $79{,}000$ to 2.68 at $80{,}000$, while Pred under the same seed and data order stays at 0.07–0.17 and the covariance variant stays at 0.54–0.77. We report the final checkpoint because every model in this paper is read at its final checkpoint, and we also scored step $70{,}000$ to check that the reading does not depend on that choice. It does not, and the two conditions move in opposite directions: at $70{,}000$ the AUC is 0.8668, which would clear the line, while the rank is 15.9, worse than at the end. The conjunction fails at both checkpoints, and it fails on the geometry condition both times.