ARIAD: ADAPTIVE TRAJECTORY MATCHING FOR PROMPT RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributed inference exposes intermediate states even when prompt text remains on the client. We introduce Adaptive Prompt Reconstruction from Internal Anchor Dynamics (ARIAD), which reconstructs a request by fitting and verifying a compact internal trajectory, without training an inverse model. An endpoint fit supplies provisional tokens; self-generated substitutions select internal anchors, and fixed target statistics define a shared metric for embedding optimization and causal token verification. We analyze the standardized cosine used by the attack: different depths can resolve different local ambiguities, while candidate omissions limit exact decoding. Controlled experiments separate conditioning, trajectory coverage, verification, and adaptive placement. Across five white-box initializations, ARIAD reaches .9806 accuracy and .6067 exact match, exceeding dual endpoints by .0069 and .0387. On 150 additional requests after independent validation selection, its accuracy is .9869 versus .9762 for a conditioned boundary; exact-match counts are 101 and 104. Shared-fit verification improves partial reconstruction without changing exact match, while larger proposal sets improve candidate coverage. Permitted-target black-box experiments further test adaptive placement under model mismatch. Experiments across domains, depths, model families, generated identifiers, and longer requests reveal both recovery gains and limits.

1 INTRODUCTION

Distributed inference lets several machines serve a large language model by passing hidden states between them (Borzunov et al., 2023a;b; Jiang et al., 2024). Although the prompt can remain on the client's device, a protocol-following worker can retain its received representation for offline reconstruction. The exposed request may contain medical descriptions, case facts, or identifiers that were never explicitly sent across the network.

Turning a hidden state back into text is difficult because nonlinear model layers separate it from the input tokens. Earlier attacks learn an inverse or optimize trial inputs to match the observation (Song & Raghunathan, 2020; Li et al., 2023; Morris et al., 2023; Dong et al., 2025). Prompt Inversion Attack (PIA) combines constrained embedding optimization with embedding and language-model proposals, then checks the proposed tokens against the observed activation (Qu et al., 2025). This makes reconstruction practical, but concentrates the comparison at one boundary. A trial prompt can match that boundary approximately while still containing wrong tokens, especially when feature scales are uneven or the attacker's model differs from the victim.

ARIAD starts from the computation that follows the boundary. During normal execution, a worker produces later states that can test the same trial prompt in different ways (Figure 1). A token difference that is difficult to distinguish at one depth may be visible at another. Matching every layer adds computation and can repeat nearly identical constraints; the states also have different coordinate scales. Reconstruction therefore requires deciding which states to retain, how to compare them, and how to reuse those comparisons when committing to discrete tokens.

We propose *Adaptive Prompt Reconstruction from Internal Anchor Dynamics* (ARIAD). A short endpoint fit produces provisional tokens from permitted observations. Substituting plausible alternatives into this sequence reveals how candidate differences propagate, allowing the attack to choose two internal anchors without private token labels. It then restarts embedding optimization from the original initialization and fits the selected trajectory using fixed, target-derived feature statistics. Finally,

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

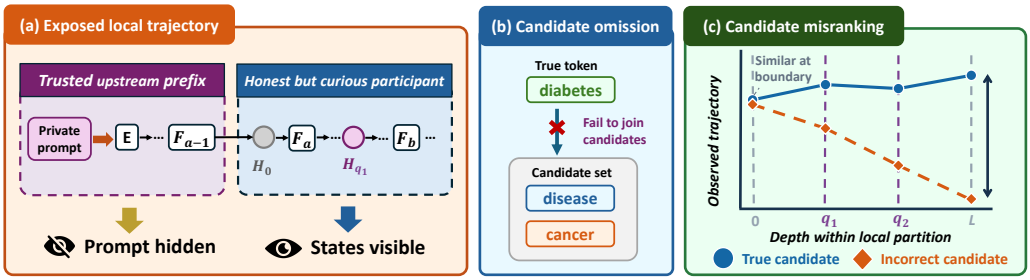


Figure 1: From exposed states to prompt reconstruction. (a) A worker observes the received boundary and states produced by its local blocks. (b) The true token may be missing from a proposed set. (c) When it is present, different depths can help distinguish it from an incorrect candidate. The trajectories are schematic.

left-to-right verification uses the same conditioned states to choose among embedding neighbors and language-model proposals. Selection determines which constraints enter the inverse problem; conditioning makes those constraints comparable across fitting and verification.

The same trajectory shapes the fitted embeddings from which tokens are proposed and tests the resulting discrete candidates. For standardized cosine near a common exact fit, combining views intersects their unresolved local directions, even after positive weights are renormalized. A separate finite-candidate analysis explains why better verification need not improve exactness when the correct token is never proposed. These limitations motivate evaluating fitting and verification separately.

Five fixed-configuration initializations yield .9806 accuracy and .6067 exact match, with positive paired gains over conditioned boundary and dual-endpoint controls. An independent test after validation selection gives ARIAD .9869 accuracy, versus .9794 for PIA, .9762 for a conditioned boundary, and .9741 for dual endpoints. Boundary matching nevertheless recovers slightly more complete requests than ARIAD, distinguishing positional accuracy from exact recovery. Holding the fit fixed reveals a further role for trajectory verification: it improves partial recovery after a shared first error. Expanding the proposal budget increases candidate coverage. The experiments separate these effects from adaptive placement, compare trajectory recovery with repeated boundary search, and test permitted-target planning under model mismatch. Generated-field tests connect reconstruction to intact sensitive content.

Contributions. We introduce a compact trajectory attack that integrates request-dependent anchor selection, target-conditioned embedding fitting, and causal token verification in one comparison metric. We relate the actual cosine objective's local ambiguities to depth coverage and distinguish them from the candidate omissions that constrain decoding. Controlled experiments separate what conditioning, coverage, verification, and placement each contribute, alongside cross-domain recovery, fixed-placement model-mismatch controls, and explicit computation and sensitive-content measurements.

2 Related Work

Text reconstruction from model outputs. Embedding inversion recovers text from learned representations (Song & Raghunathan, 2020). GEIA trains a generative inverse, while Vec2Text iteratively corrects its reconstruction (Li et al., 2023; Morris et al., 2023). Other attacks recover prompts from next-token distributions or internal transformer states (Morris et al., 2024; Dong et al., 2025; Luo et al., 2025). These tasks differ from extracting memorized training examples (Carlini et al., 2021; Nasr et al., 2025): our target is the specific request associated with a recorded activation.

From one boundary to a selected trajectory. PIA provides the constrained fitting and proposal–calibration procedure on which ARIAD builds (Qu et al., 2025). The embedding/LM proposal union and causal prefix verification are inherited components. We change their comparison metric and fitted constraints by conditioning target features and retaining selected internal depths; the finite-candidate analysis formalizes a limitation of the inherited proposal mechanism. Representation inversion and log-determinant design provide established tools (Mahendran & Vedaldi, 2015; Dosovitskiy & Brox,

2016; Nemhauser et al., 1978; Krause et al., 2008); ARIAD combines them in a request-specific attack. Boundary (std.) and the dual-endpoint variant are controls introduced here to isolate these changes. Appendix A gives the extended comparison.

3 ARIAD: Adaptive Trajectory Reconstruction

3.1 Observation and Reconstruction Interface

The attack observes an activation generated by an unknown sequence $x = (x_1, \ldots, x_T)$ and returns a reconstruction $\widehat{x}$ of length T. Let E be the embedding map and $F_0, \ldots, F_{D-1}$ the causal transformer blocks (Vaswani et al., 2017). For a capture window a–b, the received state is $Y_0 = F_{a-1} \circ \cdots \circ F_0(E(x))$. The local trajectory is $Y_q = F_{a+q-1}(Y_{q-1})$ for $q = 1, \ldots, L$, with $L = b - a + 1$. A retained depth is an *anchor*. Write $\widehat{H}^{(q)}(Z)$ for the corresponding state of trial embeddings $Z \in \mathbb{R}^{T \times d}$. All target states remain fixed during an attack.

White box uses the victim model, while gray box uses its public base model against an unknown low-rank adaptation (LoRA) (Hu et al., 2022). The non-white-box extensions are *weight-hidden but activation-visible*: a worker or tracing component observes designated boundary, internal, and output states without access to the victim checkpoint for offline evaluation. Public prompts provide paired victim/surrogate states for calibration. The same observation interface provides the private request's target states, but not its text. Black box uses a 3B surrogate and a 7B victim with a shared tokenizer, 200 public calibration prompts, and 14 matched depths. Candidate forwards and gradients use the surrogate. Fixed controls use the mapped observations; archived adaptive non-white-box results use privileged warm-up targets and are diagnostic (Appendix C.2).

Figure 2 shows the three stages. Planning allocates a compact view budget from provisional tokens; fitting and verification share the selected targets, statistics, and weights. Fixed schedules provide controls within the same procedure.

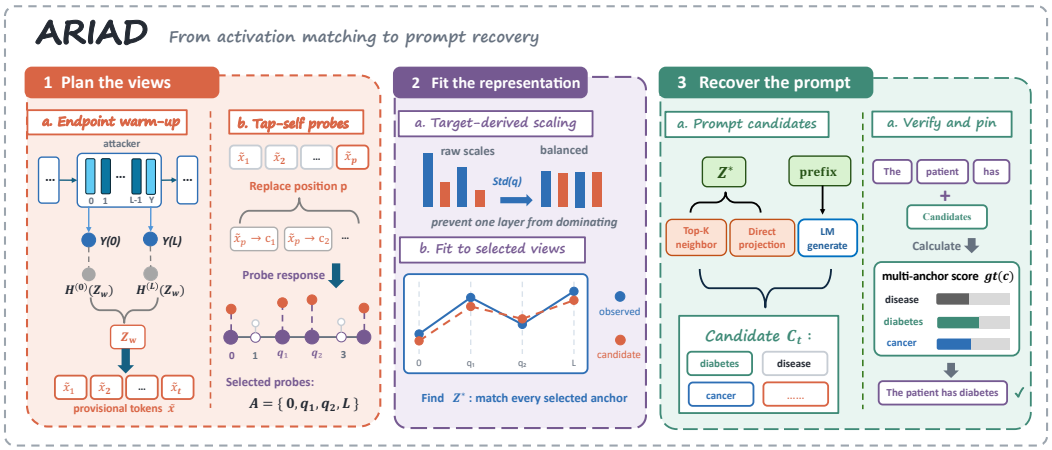


Figure 2: ARIAD plans, fits, and verifies one compact trajectory. Endpoint warm-up supplies the tokens used for substitution probes. The main fit restarts from the same initial tensor, now using the selected conditioned views. Its embedding candidates are combined with LM proposals and verified from left to right.

3.2 Plan the Trajectory with Self-Generated Probes

Endpoint warm-up. Draw each trainable embedding entry independently from $\mathcal{U}(-.1, .1)$. Fit the two raw endpoint states for 1,000 stochastic gradient descent (SGD) steps using the cosine and range terms below, without feature standardization or discrete verification. Project the resulting embeddings to provisional tokens $\widetilde{x}$. Their role is to guide the planner, not to initialize the main fit at the warm-up solution.

Internal anchor selection. Token-Activation Probing with Self-generated substitutions (TAP-Self) replaces four provisional positions with four plausible alternatives each. Replacing position p by c gives the response $\widehat{H}^{(q)}(E(\widetilde{x}_{p\to c}))_p - \widehat{H}^{(q)}(E(\widetilde{x}))_p$. Normalize these responses and stack them as columns of $S_q \in \mathbb{R}^{d\times B}$, where $B \le 16$ is the probe count. Starting from $\mathcal{A} = \{0, L\}$, greedily add two internal depths by the increase in $\log\det(\epsilon I_B + \sum_{q\in\mathcal{A}} S_q^\top S_q)$. Given a warm-up computed from permitted observations, the fixed-probe criterion selects complementary responses without requiring the private tokens. The tested untransformed white-box pipeline satisfies this condition; fixed placement bypasses planning.

3.3 Fit continuous embeddings in the selected metric

The selected anchors define the main inverse problem. For each target Y_q, compute its featurewise mean μ_q and standard deviation σ_q across positions. Set $P_q = \mathrm{diag}((\sigma_q \vee \epsilon_{\mathrm{std}})^{-1})$ and $\mathsf{Std}_q(H) = (H - \mathbf{1}\mu_q^\top)P_q$. Every candidate, including a shorter prefix, uses these same fixed statistics. Restart from the original initial tensor and minimize

$$\mathcal{L}(Z) = -\sum_{q\in\mathcal{A}} w_q \sum_{t=1}^{T} m_t \cos\Big(\mathsf{Std}_q(\widehat{H}^{(q)}(Z))_t, \mathsf{Std}_q(Y_q)_t\Big) + \lambda\mathcal{R}(Z). \tag{1}$$

Weights are normalized from 1 at each endpoint and .5 at each internal anchor, with uniform position weights $m_t = 1$. The range penalty is $\mathcal{R}(Z) = \sum_{t,j}[|Z_{tj}| - r_j]_+$, where r_j is the tenth-largest vocabulary-embedding value in coordinate j. Trainable entries are clipped to $[-.2, .2]$ before each loss evaluation. The main fit uses 2,000 SGD steps, $\lambda = .001$, and $\epsilon_{\mathrm{std}} = 10^{-6}$. Any fixed beginning-of-sequence (BOS) row is pinned. Cosine-nearest vocabulary rows to Z_t^* supply the projected token and its embedding-neighbor candidates.

3.4 Verify candidates along the same trajectory

At position t, Pinned Adaptive Discretization (PAD) combines ten embedding neighbors, ten language-model (LM) proposals conditioned on $\widehat{x}_{<t}$, and the projected token into $\mathcal{C}_t$. Duplicate candidates are removed without changing their order. For each candidate c, evaluate the discrete prefix $(\widehat{x}_{<t}, c)$ at the anchors. If its standardized cosine at view q is $s_{tq}(c)$, append the maximizer of $g_t(c) = \sum_{q\in\mathcal{A}} w_q s_{tq}(c)$, with a fixed tie rule.

Causality makes this decision independent of the unrecovered suffix. Earlier errors still change later prefixes, proposals, and scores. We therefore distinguish fitted views $\mathcal{A}_{\mathrm{fit}}$, which determine Z^*, from decoding views $\mathcal{A}_{\mathrm{dec}}$, which score a fixed fit. This distinction underlies the component controls in Section 5.2. Algorithm 1 summarizes the reconstruction stages.

4 Geometry and Decoding of Trajectory Reconstruction

Trajectory reconstruction faces two distinct limitations: ambiguous representation matches and missing token proposals. We analyze ARIAD's actual standardized cosine to relate conditioning and view coverage to the first, then characterize the finite-candidate constraint on decoding. Full proofs are in Appendix B.

An invariant coordinate comparison. For positive target variances and an inactive floor, applying the same positive diagonal scaling and translation to candidate and target leaves their standardized features unchanged (Proposition 1). The cosine test is therefore invariant to these coordinate choices. This motivates using fixed target statistics for the main fit and verification, rather than normalizing each candidate independently.

Complementary local constraints. A perfect normalized cosine identifies a positive ray, not a unique feature vector. To relate that ambiguity to the model, let $z_q(\theta)$ be a standardized candidate row, equal to nonzero target z_q^* at θ_*. Write $u_q = z_q^*/\|z_q^*\|$, $r_q = \|z_q^*\|$, and $J_q = Dz_q(\theta_*)$. With $G_q = r_q^{-2} J_q^\top (I - u_q u_q^\top) J_q$, Proposition 2 gives

$$1 - \cos(z_q(\theta_* + \eta), z_q^*) = \tfrac{1}{2}\eta^\top G_q \eta + o(\|\eta\|^2). \tag{2}$$

For positive anchor weights, $\ker(\sum_q w_q G_q) = \bigcap_q \ker G_q$, including after weight renormalization. A perturbation ignored by the boundary's angular test can therefore be constrained at a later depth. The trajectory improves local discrimination when its views resolve different directions, not simply because more loss terms are added. This expansion assumes a common exact fit; independently aligned black-box targets need not admit one, so their gains are established empirically.

From fitting to discrete verification. TAP-Self uses finite substitution responses to choose a compact set of candidate tests. Its log-determinant gain evaluates these fixed probes; the component experiments measure the resulting recovery. Once tokens are proposed, the trajectory score ranks them directly. Exact recovery also requires the true token to enter the proposed set. Theorem 1 and Corollary 2 formalize this requirement for greedy and beam decoding. This is why ARIAD combines a trajectory fit that can improve the proposals with a verifier that reuses the same conditioned geometry.

5 Experiments

We evaluate recovery, distinguish the trajectory's fitting and verification roles from adaptive placement, and examine computation and content leakage.

Table 1: Recovery across domains and attacker positions. Four adapted attacks and ARIAD; learned inverses use 2,000 public adaptation requests. Higher is better. Each setting has 150 prompts; CMS NERR is descriptive ($n = 5$). Bold marks the best point estimate.

Attacker position		2			3		
Dataset		Skytrax	CMS	ECHR	Skytrax	CMS	ECHR
Boundary layer		9–18	9–18	9–18	18–27	18–27	18–27
Acc.	$\mathcal{A}_1$	.878	.458	.616	.858	.422	.593
	GEIA	.011	.004	.003	.014	.003	.004
	Vec2Text	.000	.000	.001	.006	.003	.000
	PIA	.982	.979	**.973**	.937	.935	.929
	ARIAD	**.986**	**.992**	.969	**.975**	**.982**	**.973**
EM	$\mathcal{A}_1$	.000	.000	.000	.000	.000	.000
	GEIA	.000	.000	.000	.000	.000	.000
	Vec2Text	.000	.000	.000	.000	.000	.000
	PIA	.673	.800	.493	.200	.327	.280
	ARIAD	**.740**	**.880**	**.540**	**.540**	**.707**	**.413**
F1	$\mathcal{A}_1$	.863	.369	.501	.841	.323	.490
	GEIA	.167	.068	.087	.201	.066	.113
	Vec2Text	.033	.000	.086	.126	.052	.041
	PIA	.983	.979	**.974**	.951	.945	.937
	ARIAD	**.988**	**.993**	.969	**.974**	**.974**	**.968**
BLEU-4	$\mathcal{A}_1$	.727	.121	.194	.683	.103	.175
	GEIA	.026	.035	.029	.032	.031	.034
	Vec2Text	.000	.000	.002	.006	.021	.000
	PIA	.969	.965	**.950**	.916	.904	.881
	ARIAD	**.976**	**.986**	.941	**.950**	**.946**	**.937**
NERR	$\mathcal{A}_1$	.444	.500	.140	.390	.000	.091
	GEIA	.018	.000	.017	.032	.000	.010
	Vec2Text	.000	.000	.003	.005	.000	.001
	PIA	.973	**1.000**	**.992**	.830	**1.000**	.923
	ARIAD	**.982**	.700	.988	**.950**	.600	**.962**

5.1 Reconstruction performance and evaluation protocol

We evaluate 150 requests per standard setting from Skytrax reviews, Centers for Medicare & Medicaid Services (CMS) descriptions, and European Court of Human Rights (ECHR) case facts, with at most 64 tokens (Qu et al., 2025; Chalkidis et al., 2019). Extensions include 80 requests of length 256 and

150 generated identifier-bearing prompts. The model grids cover Qwen2.5 (3B–32B), Llama-3.1, Gemma-2, and Phi-3.5 (Yang et al., 2024; Grattafiori et al., 2024; Gemma Team, 2024; Abdin et al., 2024; Microsoft, 2024).

Published-method baselines and internal controls. We adapt $\mathcal{A}_1$, GEIA, Vec2Text, and PIA to the common observation interface (Luo et al., 2025; Li et al., 2023; Morris et al., 2023; Qu et al., 2025). The learned inverses use 2,000 public adaptation requests. In white box, our PIA implementation uses the shared reconstruction engine with a raw boundary target. Two controls constructed for this study, Boundary (std.) and the dual-endpoint variant, add target standardization and output-endpoint coverage respectively. They measure the benefit beyond a conditioned boundary. The 10,000-example adaptation extension (Table 37) improves several learned inverses, including $\mathcal{A}_1$ to .972 accuracy. The canonical runs use one NVIDIA H20 and fp16 with SDPA. Appendix C gives the parameters and training budgets.

We report positional accuracy (Acc.), exact match (EM), token F1, BLEU-4, and named-entity recovery ratio (NERR) (Papineni et al., 2002). NERR requires complete entity surfaces. Prompt intervals condition on initialization; repeated-initialization comparisons also account for the seed axis. Table 1 reports the fixed-initialization breadth benchmark. Repeated-initialization comparisons and the additional validation-selected test below use separately identified cohorts.

Table 2: Independent test after validation selection. Qwen2.5-7B, deep states; rates chosen on 20 disjoint validation requests and frozen before an additional 150-request test. One fixed initialization; EM is exact requests divided by 150. No test-time selection.

Method	Learning rate	Acc.	Exact / 150	EM
PIA	.5	.9794	84	.5600
Boundary (std.)	.3	.9762	104	.6933
Dual endpoint	.3	.9741	78	.5200
ARIAD	.3	.9869	101	.6733

Table 2 tests configurations frozen after learning-rate selection on 20 independent validation requests. On 150 additional requests, ARIAD improves accuracy over PIA, Boundary (std.), and dual endpoints by .0075, .0107, and .0128. Its EM is 101/150, compared with 104/150 for Boundary (std.); this single-initialization comparison supplies point estimates, not repeated-seed significance. Across five fixed-configuration initializations, ARIAD reaches .9806 accuracy and .6067 EM. Relative to dual endpoints, the paired gains are .0069 [.0014,.0124] accuracy and .0387 [.0071,.0699] EM. Repeated-initialization accuracy contrasts also favor ARIAD over the conditioned boundary on Qwen-7B, ECHR, and Qwen-3B; Phi favors the boundary (Appendix E).

Table 3: Participant-count and depth sweep. Qwen2.5-7B on Skytrax. Full five-method results are in Table 11.

Participant number		4			5			
Position		2	3	4	2	3	4	5
Boundary layer		7–14	14–20	20–27	5–11	11–16	16–22	22–27
Acc.	PIA	.982	.965	.931	.998	.974	.945	.929
	ARIAD	**.995**	**.981**	**.968**	**.999**	**.983**	**.980**	**.956**
EM	PIA	.807	.493	.147	.927	.640	.300	.040
	ARIAD	**.873**	**.680**	**.467**	**.947**	**.713**	**.580**	**.267**
F1	PIA	.983	.973	.947	.998	.980	.955	.938
	ARIAD	**.996**	**.983**	**.967**	**.999**	**.984**	**.979**	**.955**
BLEU-4	PIA	.975	.950	.907	.997	.964	.923	.891
	ARIAD	**.992**	**.967**	**.937**	**.998**	**.969**	**.958**	**.909**
NERR	PIA	.986	.936	.795	**1.000**	.959	.889	.820
	ARIAD	**.986**	**.956**	**.940**	**1.000**	**.963**	**.953**	**.926**

Table 3 varies depth within the same Qwen-7B model. At the last four-participant position, PIA/ARIAD EM is .147/.467 and NERR .795/.940. With five participants, the respective values are .040/.267 and .820/.926. The gain reaches intact entity surfaces as well as whole requests.

Appendix D reports every method at both positions for the size and family sweeps, including the stronger shallow PIA results on some families.

5.2 Conditioning, coverage, and verification

Target conditioning is a substantial component: its paired mean effect on EM is $+.2275$ [.1900,.2633]. Table 4 then isolates the additional contributions of internal coverage, adaptive placement, and verification. The comparisons average the prescribed matched arm pairs, keeping prompt identities paired.

Table 4: Paired fitting and verification contrasts. Fixed initialization and paired requests. The first three interventions compare fitted-view configurations; the last three hold the fit fixed and change only the verifier. The zero EM difference is reported without a separate interval.

Intervention	Metric	Δ	95% CI
Uniform-2 − endpoints	Acc.	+.0074	[.0016, .0132]
TAP − uniform	Acc.	+.0068	[.0011, .0125]
TAP − fixed random	Acc.	+.0079	[.0019, .0139]
All-view − boundary verifier	Acc.	+.0052	[.0034, .0070]
All-view − boundary verifier	EM	0	—
All-view − boundary verifier	NERR	+.0056	[.0009, .0103]

Trajectory fitting and trajectory verification are distinct interventions. The former changes the optimized embeddings; the latter reuses a fixed fit and candidate construction. The shared-fit comparison fixes the optimized embeddings and proposal construction, changing only the views used for causal scoring (Appendix F).

Additional verification improves partial text without creating another exact prompt. Across six shared-fit pairs, all-view scoring adds .0052 [.0034,.0070] accuracy and .0056 [.0009,.0103] NERR, with zero EM difference. The event summary contains 521 pair–prompt records sharing the first candidate omission and six positive suffix-accuracy intervals. Table 21 gives the event summary. These comparisons reuse the request cohort; the event count is not a count of independent requests. Appendix F connects the measurements to the finite-candidate condition in Theorem 1.

The pooled accuracy contrasts favor uniform internal coverage over endpoints by .0074 [.0016,.0132], and TAP over uniform and random placement by .0068 [.0011,.0125] and .0079 [.0019,.0139]. Additional placement/initialization comparisons show positive TAP contrasts on Phi and Qwen-3B (Appendix I). This within-planner improvement does not imply that the full pipeline wins in every setting: on Phi, ARIAD remains .0524 below Boundary (std.) in repeated-initialization accuracy.

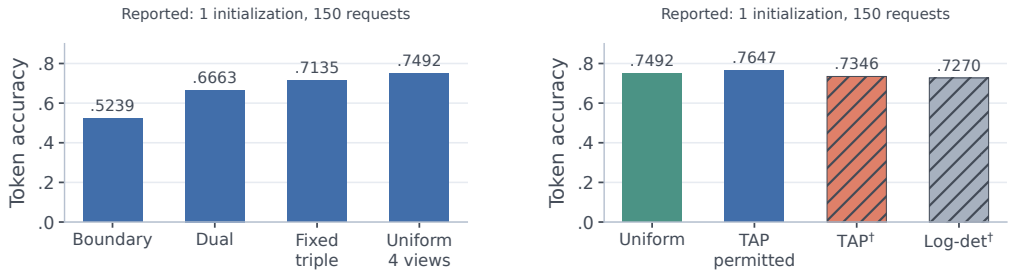
5.3 Reconstruction with calibrated victim states

Table 5: Black-box reconstruction with fixed and permitted-target placement. Procrustes calibration, 150 requests, one initialization. Fixed controls bypass planning. Dashes denote unavailable metrics; NERR uses 147 entity-bearing requests. Results are point estimates. The five-method comparison is in Table 24.

Configuration	Acc.	EM	F1	BLEU-4	NERR
Boundary (std.)	.5239	.0000	.550	.289	.251
Dual endpoints	.6663	.0000	.654	.400	.329
Fixed triple	.7135	.0000	.708	.479	.405
Uniform, 4 views	.7492	.0000	.735	.515	.3187
TAP, permitted targets	.7647	.0000	—	—	—

Table 5 compares reconstruction from calibrated victim states. Fixed boundary, dual-endpoint, triple, and uniform-four-view controls reach .5239, .6663, .7135, and .7492 accuracy. Uniform NERR is .3187, below the fixed triple's .4052 despite higher positional accuracy. TAP with permitted-target planning reaches .7647, a .0155 increase over uniform. All EM values are zero. These 150-request, single-initialization results are point estimates; no paired interval is supplied for the placement contrast.

Privileged-planner diagnostics use private-text reconstruction-model states during planning. Their .7346 TAP result is distinct from permitted-target planning and excluded from equal-access comparisons. Appendix G specifies the observation interfaces and separates these diagnostic extensions.



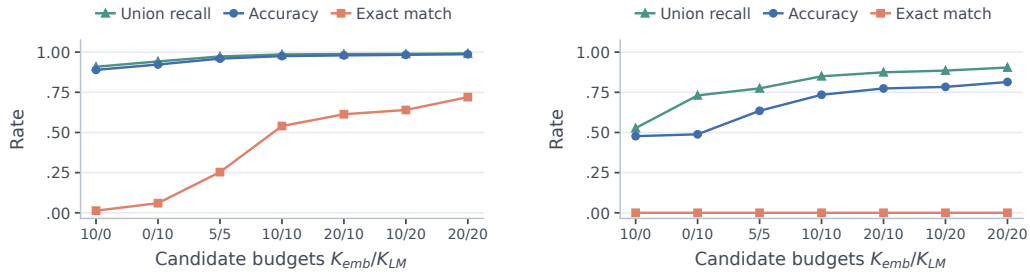
(a) Fixed-placement black-box controls.

(b) Planning access and placement.

Figure 3: Trajectory coverage and planning access. Left: controls share standardization and calibration and bypass planning. Right: permitted-target TAP is separate from the two privileged-planner diagnostics (†). Bars are point estimates.

Figure 3 supports richer observed trajectories under model mismatch, without isolating view count from placement or weights. A middle-only view reaches .675 accuracy. Appendix G retains the calibration sweeps as privileged-planner diagnostics.

5.4 Reconstruction under alternative search budgets



(a) White-box reconstruction.

(b) Privileged-planner black-box diagnostic.

Figure 4: Candidate-budget sweep without refitting. Budgets give embedding/LM top-K counts; the projected token is always retained. Union recall uses true-prefix diagnostics. The black-box panel conditions on the archived privileged fit.

ARIAD's candidate branches directly control proposal coverage without refitting. Expanding both budgets from 10 to 20 raises white-box union recall from .9858 to .9919 (Table 29). The retained white-box candidate sweep raises EM from .540 to .720, while a four-wide beam at the original candidate budget leaves EM at .540 (Figure 4). Larger sets can introduce both correct identities and stronger competitors. The black-box panel conditions on a privileged fit; Appendix G distinguishes its scope.

Repeated initialization instead gives the boundary inverse several attempts selected by boundary agreement. Two boundary starts reach .811 EM at 222.6 seconds; three starts reach .891 at 319.2 seconds. Appendix H compares these with ARIAD recovery–time profiles and reports cap-specific delivery counts separately. Mean operating points alone do not establish equivalence of latency distributions.

In the black-box resource comparison, fixed dual endpoints reach .6648 accuracy at 133.9 seconds, compared with .5973 at 321.3 seconds for three boundary-selected starts under the same 447.7-second allowance (Table 6). Both have zero EM and zero overruns over 450 evaluations. The reported paired accuracy difference is .0675 [.0581,.0769]. This comparison supports endpoint coverage; adaptive placement is evaluated separately.

Table 6: Endpoint coverage versus repeated boundary search. Three paired initializations on 150 requests under $H_2 = 447.7$ s. Dual minus three-start Boundary accuracy is +.0675, with 95% paired interval [.0581,.0769]. Both have zero EM and 0/450 overruns. Starts are selected by standardized-boundary agreement; neither method uses planning. Protocol: Appendix H.

Strategy	Acc.	EM	Seconds	Overruns
Boundary, 3 starts	.5973	.0000	321.3	0/450
Dual endpoints	.6648	.0000	133.9	0/450

The privileged-planner resource diagnostic is separate from the fixed-control comparison. Adaptive delivery under the permitted-target interface is not established by these fixed-control results (Appendix H).

We also apply the shared decoders to a different continuous inverse. On the same 138-request ER cohort, the 10,000-step fit improves from .161 accuracy with nearest projection to .471 with shared boundary PAD and .549 with shared four-view PAD (Dong et al., 2025). Using the same fitted outputs isolates the benefit of the discrete verifier. Appendix G explains the 12 incompatible ER outputs and keeps the other methods' 150-request cohorts separate.

5.5 Sensitive identifiers and longer requests

Generated identifier-bearing requests test fresh fields embedded in natural-language context. ARIAD exactly recovers 26/32 emails (.8125); field denominators are distinct from prompt EM (Table 55). PIA recovers 19/32 emails and 29/47 card suffixes, versus ARIAD's 26/32 and 38/47. PIA remains stronger on API keys (8/36 versus 1/36) and UUID fragments (17/28 versus 11/28). Turning off LM proposals lowers overall accuracy from .896 to .552 (Table 54), so removing that branch does not resolve the random-field limitation.

At 256 tokens, ARIAD retains .904 accuracy, against Boundary (std.)'s .892 and PIA's .874, while all EM values are zero. Boundary (std.) is slightly higher on F1 and BLEU-4. Table 57 retains these distinct outcomes and the complete entity results. Both studies motivate reporting intact sensitive content alongside full-prompt correctness.

6 Implications for Protection

Appendix J examines null-space-aligned masking (NSAM), including its local gain bound and privacy–utility diagnostics (Figure 6). The reference warm-up omits the defense transformation, and compliant historical execution remains unverified. These measurements therefore do not establish protection against a deployable attack. The intact identifiers recovered in the untransformed experiments motivate methods with formal privacy guarantees (Dwork et al., 2006).

7 Conclusion

ARIAD reconstructs private prompts using request-dependent anchors and a common target-conditioned metric for fitting and verification. Experiments separate conditioning, view coverage, candidate availability, and setting-dependent placement gains. Shared-fit controls reveal partial recovery even when exact match is unchanged. Privacy assessment must account for the intact identifiers exposed by internal computation.

Generative AI Usage Statement

Generative AI tools were used extensively in checking proofs, implementing and debugging code, polishing the manuscript. The submitting authors are responsible for the final text, mathematical claims, code, and interpretation.

Ethics Statement

Prompt reconstruction is dual-use: it can reveal sensitive information, but it also enables privacy testing of distributed inference. Our experiments use public research text and generated identifiers in controlled model executions, rather than traffic collected from deployed users. The attack assumes the stated access to intermediate activations; it does not provide a method for obtaining unauthorized access. Public text can still contain identifying details, so data sharing should respect the source licenses and avoid unnecessary disclosure. Generated identifiers are used to examine request-specific leakage without targeting real account credentials. The human assessment concerns reconstruction and response quality, not attributes of the annotators. The defense study reports measured privacy–utility trade-offs rather than a formal privacy guarantee. We intend the method for authorized evaluation and the design of safer inference systems.

Reproducibility Statement

Appendix C specifies observation interfaces, initialization, optimization parameters, hardware, recovery metrics, and uncertainty. Algorithms and proofs appear in Appendices C and B. Separate supplementary materials contain the reference implementation, a permitted-target warm-up patch and instrumented runner, fixed evaluation subsets, structured results, generators, and mathematical checks. Offline code checks are distinct from model evaluations; extension-specific execution traces are not included.

References

Marah Abdin et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv preprint arXiv:2404.14219, 2024. URL https://arxiv.org/abs/2404.14219.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *Proceedings of NAACL-HLT: Demonstrations*, pp. 54–59, 2019. doi: 10.18653/v1/N19-4010. URL https://aclanthology.org/N19-4010/.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. arXiv preprint arXiv:1607.06450, 2016. URL https://arxiv.org/abs/1607.06450.

Alexander Borzunov, Dmitry Baranchuk, Tim Dettmers, Maksim Riabinin, Younes Belkada, Artem Chumachenko, Pavel Samygin, and Colin Raffel. Petals: Collaborative Inference and Fine-tuning of Large Models. In *Proceedings of ACL: System Demonstrations*, pp. 558–568, 2023a. doi: 10.18653/v1/2023.acl-demo.54. URL https://aclanthology.org/2023.acl-demo.54/.

Alexander Borzunov, Max Ryabinin, Artem Chumachenko, Dmitry Baranchuk, Tim Dettmers, Younes Belkada, Pavel Samygin, and Colin Raffel. Distributed Inference and Fine-tuning of Large Language Models over the Internet. In *Advances in Neural Information Processing Systems*, 2023b. URL https://arxiv.org/abs/2312.08361.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium*, pp. 267–284, 2019. URL https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium*, pp. 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural Legal Judgment Prediction in English. In *Proceedings of ACL*, pp. 4317–4323, 2019. doi: 10.18653/v1/P19-1424. URL https://aclanthology.org/P19-1424/.

Tian Dong, Yan Meng, Shaofeng Li, Guoxing Chen, Zhen Liu, and Haojin Zhu. Depth Gives a False Sense of Privacy: LLM Internal States Inversion. In *34th USENIX Security Symposium*, pp. 1629–1648, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/dong-tian.

Alexey Dosovitskiy and Thomas Brox. Inverting Visual Representations with Convolutional Networks. In *Proceedings of CVPR*, 2016. URL https://arxiv.org/abs/1506.02753.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*, pp. 265–284, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Bradley Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal Aggregation Algorithms for Middleware. *Journal of Computer and System Sciences*, 66(4):614–656, 2003. doi: 10.1016/S0022-0000(03)00026-6. URL https://arxiv.org/abs/cs/0204046.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting Gradients: How Easy Is It to Break Privacy in Federated Learning? In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2003.14053.

Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. arXiv preprint arXiv:2408.00118, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori et al. The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783, 2024. URL https://arxiv.org/abs/2407.21783.

Kang Gu, Ehsanul Kabir, Neha Ramsurrun, Soroush Vosoughi, and Shagufta Mehnaz. Towards Sentence Level Inference Attack Against Pre-trained Language Models. *Proceedings on Privacy Enhancing Technologies*, 2023(3):62–78, 2023. doi: 10.56553/popets-2023-0070. URL https://petsymposium.org/popets/2023/popets-2023-0070.php.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering Private Text in Federated Learning of Language Models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.08514.

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

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1811.06965.

Youhe Jiang, Ran Yan, Xiaozhe Yao, Yang Zhou, Beidi Chen, and Binhang Yuan. HexGen: Generative Inference of Large Language Model over Heterogeneous Environment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21946–21961, 2024. URL https://proceedings.mlr.press/v235/jiang24f.html.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, 9:235–284, 2008. URL https://www.jmlr.org/papers/v9/krause08a.html.

Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence Embedding Leaks More Information than You Expect: Generative Embedding Inversion Attack to Recover the Whole Sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 14022–14040, 2023. doi: 10.18653/v1/2023.findings-acl.881. URL https://aclanthology.org/2023.findings-acl.881/.

Xinjian Luo, Ting Yu, and Xiaokui Xiao. Prompt Inference Attack on Distributed Large Language Model Inference Frameworks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2025. URL https://arxiv.org/abs/2503.09291.

Aravindh Mahendran and Andrea Vedaldi. Understanding Deep Image Representations by Inverting Them. In *Proceedings of CVPR*, pp. 5188–5196, 2015. URL https://arxiv.org/abs/1412.0035.

Microsoft. Phi-3.5-mini-instruct Model Card. Model documentation, 2024. URL https://huggingface.co/microsoft/Phi-3.5-mini-instruct.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text Embeddings Reveal (Almost) As Much As Text. In *Proceedings of EMNLP*, pp. 12448–12460, 2023. doi: 10.18653/v1/2023.emnlp-main.765. URL https://aclanthology.org/2023.emnlp-main.765/.

John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. Language Model Inversion. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/999fcab97007ebef0cda9949550b4a9e-Abstract-Conference.html.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable Extraction of Training Data from Aligned, Production Language Models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cce0e917b050208170151f77b497fc71-Abstract-Conference.html.

Murtaza Nazir, Matthew Finlayson, John X. Morris, Xiang Ren, and Swabha Swayamdipta. Better Language Model Inversion by Compactly Representing Next-Token Distributions. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/0ccd06ff26fd6a7829293ce90e0e7f7d-Abstract-Conference.html.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions—I. *Mathematical Programming*, 14:265–294, 1978. doi: 10.1007/BF01588971. URL https://doi.org/10.1007/BF01588971.

Art B. Owen. The Pigeonhole Bootstrap. arXiv preprint arXiv:0712.1111, 2007. URL https://arxiv.org/abs/0712.1111.

Ashwinee Panda, Xinyu Tang, Milad Nasr, Christopher A. Choquette-Choo, and Prateek Mittal. Privacy Auditing of Large Language Models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e76814a0012a46ade4f997591e86f972-Abstract-Conference.html.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, pp. 311–318, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Wenjie Qu, Yuguang Zhou, Yongji Wu, Tingsong Xiao, Binhang Yuan, Yiming Li, and Jiaheng Zhang. Prompt Inversion Attack against Collaborative Inference of Large Language Models. In *IEEE Symposium on Security and Privacy*, 2025. URL https://arxiv.org/abs/2503.09022.

Peter H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451.

Congzheng Song and Ananth Raghunathan. Information Leakage in Embedding Models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 377–390, 2020. doi: 10.1145/3372297.3417270. URL https://arxiv.org/abs/2004.00053.

Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida Correira, Antonia Adler, and Kristian Kersting. Plug & Play Attacks: Towards Robust and Flexible Model Inversion Attacks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20522–20545, 2022. URL https://proceedings.mlr.press/v162/struppek22a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 Technical Report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

An Yang et al. Qwen2.5 Technical Report. arXiv preprint arXiv:2412.15115, 2024. URL https://arxiv.org/abs/2412.15115.

Wentao Ye, Jiaqi Hu, Haobo Wang, Xinpeng Ti, Zhiqing Xiao, Hao Chen, Liyao Li, Lei Feng, Sai Wu, and Junbo Zhao. An Invariant Latent Space Perspective on Language Model Inversion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):27818–27826, 2026. doi: 10.1609/aaai.v40i33.40004. URL https://ojs.aaai.org/index.php/AAAI/article/view/40004.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1910.07467.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1906.08935.

APPENDIX CONTENTS

A Extended related work **16**
A.1 Observation interfaces 16
A.2 Fitting and activation calibration 16
A.3 Serving and model access 16
A.4 Adjacent privacy tasks 16
A.5 Geometry and score evaluation 17

B Geometry and decision guarantees **17**
B.1 Coordinate invariance 17
B.2 Angular ambiguity 17
B.3 First-failure decomposition 18
B.4 Correlated scores and view selection 20
B.5 Certified score evaluation 21

C Protocol and algorithms **21**
C.1 Capture windows and checkpoints 21
C.2 White-, gray-, and black-box access 21
C.3 Target statistics and configuration 23
C.4 Data and recovery metrics 25
C.5 Uncertainty and resource accounting 25
C.6 Reconstruction algorithms 26

D Complete benchmarks **27**
D.1 Participant count 27
D.2 Model size 27
D.3 Model family 28
D.4 Checkpoint-incompatible records 29

E Boundary controls and initialization **30**
E.1 Conditioned controls 30
E.2 Repeated comparisons 30
E.3 Validation and independent testing 31

F Attribution and first-error diagnostics **32**
F.1 Fitting and verification effects 32
F.2 Factorial estimands 32
F.3 First-error event counts 33
F.4 Recovery after the first failure 33

G Gray/black-box and decoder controls **35**
G.1 Five-method access comparison 35
G.2 Views and calibration data 36
G.3 Candidate-budget sweep 37

G.4 Candidate diagnostics and beam search . . . 38
G.5 Shared-decoder comparisons . . . 39
G.6 Public computation and training data . . . 40

H Search resources and delivery . . . 41
H.1 Budget protocol . . . 41
H.2 Canonical Qwen-7B . . . 42
H.3 ECHR . . . 42
H.4 Black-box resource results . . . 42
H.5 Elapsed time and timeouts . . . 42
H.6 Stage costs . . . 42
H.7 Exact-score evaluation . . . 43

I Anchor placement and count . . . 46
I.1 Adaptive and fixed placement . . . 46
I.2 Placement and initialization . . . 47
I.3 Internal-anchor count . . . 47
I.4 Probe geometry and recovery . . . 48

J Content, length, and protection . . . 48
J.1 Lexical recovery . . . 48
J.2 Generated identifiers . . . 49
J.3 Order and longer sequences . . . 50
J.4 All defense configurations . . . 50
J.5 Noise replication and semantic scores . . . 51

A Extended Related Work

A.1 Observations define the reconstruction problem

Embedding inversion reconstructs text from a pooled or contextual representation. Song & Raghunathan (2020) analyze embedding leakage, while GEIA, sentence-level inference, and Vec2Text learn generative inverses with different conditioning and correction procedures (Li et al., 2023; Gu et al., 2023; Morris et al., 2023). The experiments here adapt these learned inverses to tokenwise transformer states and vary their public adaptation budget. This produces a direct comparison between learning an inverse from public examples and fitting each observed request separately.

Language-model inversion reconstructs a prefix from its next-token distribution (Morris et al., 2024). Subsequent work changes the distribution representation or aligns latent spaces to improve inversion (Nazir et al., 2025; Ye et al., 2026). These results establish leakage through outputs, complementing the internal-state interface studied by Dong et al. (2025). Luo et al. (2025) study prompt inference in distributed serving, including the positionwise attack used here as $\mathcal{A}_1$. The source of the observation determines the available candidate tests and the public training data required to learn them.

A.2 Continuous fitting and causal activation calibration

PIA combines constrained continuous embeddings, embedding-neighbor and LM proposals, and causal activation calibration (Qu et al., 2025). Its sequential argument uses the dependence of each activation row on the current prefix. Its finite-candidate implementation also identifies the role of true-token inclusion. ARIAD builds on this interface through target-conditioned feature scaling and a retained trajectory shared by fitting and verification.

The experiments vary these operations independently. A fitted representation supplies candidate identities; the selected decoding views rank them. The first-failure decomposition formalizes both omission and misranking, and the paired traces locate which event occurs in the evaluated white-box runs. The shared-decoder study complements this analysis by applying nearest projection, boundary PAD, and four-view PAD to a cached continuous fit. It measures how a discrete procedure changes reconstruction with the optimization output fixed.

A.3 Distributed serving and model access

Pipeline parallelism partitions model computation (Huang et al., 2019). Petals extends inference and adaptation across independently operated machines, while heterogeneous serving introduces placement and scheduling constraints (Borzunov et al., 2023a;b; Jiang et al., 2024). ARIAD considers a participant that follows the protocol and records incoming states. Its resident blocks define a readily available set of target responses.

The three experimental access regimes correspond to a known victim, a known base with an unknown low-rank adaptation (Hu et al., 2022), and a surrogate aligned through public activation pairs. Their operational definitions are given in Section 3.1. Checkpoints from Qwen2.5, Llama, Gemma, and Phi supply model diversity (Yang et al., 2024; Grattafiori et al., 2024; Gemma Team, 2024; Abdin et al., 2024; Microsoft, 2024). Qwen3 provides an additional historical grid (Yang et al., 2025), whose replay identity is documented separately in Appendix C.

A.4 Related privacy tasks

Gradient inversion observes training updates rather than serving activations. Image reconstruction from gradients and text recovery in federated learning establish adjacent privacy risks (Zhu et al., 2019; Geiping et al., 2020; Gupta et al., 2022). Training-data extraction instead asks a model to emit memorized content (Carlini et al., 2021; Nasr et al., 2025). Canary-based memorization measurement and controlled privacy auditing motivate explicit evaluation targets and careful sample construction (Carlini et al., 2019; Panda et al., 2025).

The canary study here embeds generated fields in carrier text. It measures recovery of request-specific identifiers alongside public natural-language requests. Positional accuracy, lexical overlap, exact prompt recovery, and exact field recovery each measure a different disclosure event. The comparison

includes both structured fields and random hexadecimal strings, which behave differently under the same attack.

A.5 Geometry and exact score evaluation

Optimization-based and learned representation inversion precede the language setting (Mahendran & Vedaldi, 2015; Dosovitskiy & Brox, 2016; Struppek et al., 2022). Internal features define an inverse objective, and their numerical parameterization affects the optimization. Target-derived scaling differs from architectural layer normalization and root mean square normalization (Ba et al., 2016; Zhang & Sennrich, 2019): the statistics come from one observed target and are then frozen for every reconstruction candidate.

The planner's log-determinant criterion is related to submodular experimental design and information-based sensor placement (Nemhauser et al., 1978; Krause et al., 2008). Its fixed-probe objective describes response diversity; actual token recovery is measured through candidate decisions. Centered kernel alignment (CKA) offers a complementary representation-level summary (Kornblith et al., 2019). We distinguish these diagnostics from the empirical coverage and first-error events.

The optional partial-score evaluator specializes bounded aggregation (Fagin et al., 2003) to the existing weighted-cosine PAD score. Its correctness follows from intervals for uncomputed scores and causal prefix induction. Bootstrap and crossed-data resampling provide the statistical basis for separating prompt, initialization, and placement variability (Efron, 1979; Owen, 2007). Finally, the activation perturbations measure a privacy–utility trade-off; formal differential privacy supplies a separate guarantee-based protection framework (Dwork et al., 2006).

B Geometry, Candidate Decisions, and Proofs

This appendix separates three mathematical objects: the target-derived coordinate map, the planner's finite-difference probe matrices, and the verifier's actual candidate scores. The coordinate map acts on the fitting loss, the probe objective selects anchors, and the score defines the discrete decision. Algebraic claims use exact arithmetic unless stated otherwise.

B.1 Target-coordinate invariance

Let $Y \in \mathbb{R}^{T\times d}$ have column means μ and positive standard deviations σ. Fix the convention for the standard-deviation denominator. For an inactive floor, put $P = \operatorname{diag}(\sigma^{-1})$ and $\mathsf{Std}_Y(H) = (H - \mathbf{1}\mu^\top)P$.

Proposition 1 (Positive diagonal-affine invariance)**.** *For any positive diagonal matrix D and translation b, define $Y' = YD + \mathbf{1}b^\top$ and $H' = HD + \mathbf{1}b^\top$. Then $\mathsf{Std}_{Y'}(H') = \mathsf{Std}_Y(H)$. Among positive diagonal conditioners, $Q = c\operatorname{diag}(\sigma^{-1})$, $c > 0$, are exactly those for which $Q\operatorname{diag}(\sigma^2)Q$ has condition number one.*

Proof. The transformed mean is $D\mu + b$ and each transformed standard deviation is $D_{jj}\sigma_j$. Centering cancels b, so $(H' - \mathbf{1}\mu'^\top)\operatorname{diag}(\sigma'^{-1}) = (H - \mathbf{1}\mu^\top)DPD^{-1} = (H - \mathbf{1}\mu^\top)P$. For diagonal Q, the eigenvalues of the conditioned diagonal covariance are $Q_{jj}^2\sigma_j^2$. Their ratio is at least one, with equality exactly when all $Q_{jj}\sigma_j$ are the same positive constant. □

The optimized object is the diagonal target covariance. Full whitening would additionally mix features and may be singular when the number of positions is smaller than the feature dimension. The proposition does not optimize the nonconvex inversion Hessian. When a fixed numerical floor becomes active after rescaling, exact scale invariance can fail; the implementation's floor is therefore part of the configuration, not omitted from the method.

B.2 The angular ambiguity of the actual objective

For a single target row y, let $z_* = P(y - \mu)$ and suppose $r = \|z_*\|_2 > 0$. The standardized candidate is $z = P(h - \mu)$. Since cosine removes magnitude, equality to one has a precise geometric meaning.

Lemma 1 (A perfect cosine identifies a ray)**.** *If $z \neq 0$, then $\cos(z, z_*) = 1$ iff $h = \mu + \alpha(y - \mu)$ for some $\alpha > 0$.*

Proof. Equality in Cauchy–Schwarz with a positive inner product holds iff $z = \alpha z_*$ with $\alpha > 0$. Since P is invertible, multiplying by P^{-1} yields the stated affine ray. □

This ambiguity remains after exact coordinate balancing. It concerns the comparison objective, not necessarily a realizable perturbation through every preceding transformer block. The next result incorporates the candidate map.

Proposition 2 (Local angular pullback)**.** *Let $z_q(\theta)$ be the fixed-target standardized candidate row for view q, differentiable at θ_*, with $z_q(\theta_*) = z_q^* \neq 0$. Write $u_q = z_q^*/\|z_q^*\|$, $r_q = \|z_q^*\|$, and $J_q = Dz_q(\theta_*)$. For a perturbation η,*

$$1 - \cos(z_q(\theta_* + \eta), z_q^*) = \tfrac{1}{2}\eta^\top G_q \eta + o(\|\eta\|^2), \qquad G_q = r_q^{-2} J_q^\top (I - u_q u_q^\top) J_q. \tag{3}$$

For positive weights, $G_A = \sum_{q \in A} w_q G_q$ satisfies $\ker G_A = \bigcap_{q \in A} \ker G_q$. This kernel identity remains true when the positive weights are renormalized.

Proof. The derivative of normalization $n(z) = z/\|z\|$ at z_q^* is $r_q^{-1}(I - u_q u_q^\top)$. Because $1 - n(z)^\top u_q = \frac{1}{2}\|n(z) - u_q\|^2$, differentiability and substitution give Equation (3). Each G_q is positive semidefinite. Thus $\eta^\top G_A \eta = 0$ iff every positively weighted squared norm $\|(I - u_q u_q^\top) J_q \eta\|^2$ is zero, establishing the intersection. Changing positive magnitudes cannot alter that zero set. □

For the full sequence loss, sum these matrices over the positive position weights. A direction that is radial for one normalized target can be transverse to another. Extra views then remove some locally unresolved directions. Conversely, if all views have the same null directions, no such removal occurs. The proposition does not claim eigenvalue monotonicity under weight renormalization or convergence from a remote random initialization.

Table 7: Local angular expansion on 80 fixed perturbation directions. Target statistics remain fixed. Floor-activation counts are zero. Relative residuals increase at the smallest perturbations; these data do not establish asymptotic convergence.

Views	δ	Measured loss	Quadratic term	Mean $\lvert R\rvert/\delta^2$
boundary	0.0001	1.422e-09	3.312e-10	1.091e-01
boundary	0.0003	4.576e-09	2.981e-09	1.780e-02
boundary	0.001	3.461e-08	3.312e-08	2.445e-03
boundary	0.003	3.016e-07	2.981e-07	1.082e-03
endpoints	0.0001	2.444e-09	5.243e-10	1.920e-01
endpoints	0.0003	7.684e-09	4.719e-09	3.316e-02
endpoints	0.001	5.522e-08	5.243e-08	4.459e-03
endpoints	0.003	4.802e-07	4.719e-07	2.109e-03
4views	0.0001	3.402e-09	7.276e-10	2.674e-01
4views	0.0003	1.072e-08	6.549e-09	4.670e-02
4views	0.001	7.666e-08	7.276e-08	6.299e-03
4views	0.003	6.669e-07	6.549e-07	3.016e-03

Table 7 evaluates the expansion along 80 supplied perturbation directions on the actual model. At $\delta = .003$, measured and quadratic mean losses are close. At 10^{-4}, the normalized residual is much larger. This pattern is compatible with numerical resolution limiting the smallest perturbations; it does not establish an asymptotic empirical rate. No floor activations are recorded in this diagnostic.

B.3 The first failure of finite-candidate greedy reconstruction

Theorem 1 makes no perfect-cosine or true-token margin assumption. Let a deterministic proposal operator $C_t(p)$ use prefix p, fixed optimized embeddings, and any fixed external model. Let verifier v return a candidate $v_t(p, C_t(p))$ according to its specified score and tie rule. A fixed initial special token may be removed from the variable-position indexing.

On the true prefix define $M_t = [x_t \notin C_t(x_{<t})]$ and $R_t^v = [x_t \in C_t(x_{<t}) \wedge v_t(x_{<t}, C_t(x_{<t})) \neq x_t]$. The free-running decoder substitutes its own prefix at every step.

Theorem 1 (First-failure decomposition)**.** *Fix the continuous fit, candidate rule, verifier v, and tie order. For nonempty candidate sets and causal deterministic proposal and verification rules, free-running greedy decoding first fails at*

$$\tau_v = \min\{t : M_t \text{ or } R_t^v\}, \tag{4}$$

with an empty minimum equal to $T+1$. Verifiers sharing proposals have the same first error whenever neither misranks a token before the first omission.

Proof of Theorem 1. Let $j = \min\{t : M_t \vee R_t^v\}$, with $j = T+1$ if there is no such position. We prove inductively that the free-running prefix is $x_{<t}$ for every $t \leq j$. At the first variable position both executions start with the same fixed prefix. If the prefixes agree at $t < j$, then M_t and R_t^v are both false, so the true token is proposed and selected. The prefixes agree at $t+1$. At $j \leq T$, either the true token is unavailable or it is available and another token is selected. Both cases force the first wrong token exactly there. If $j = T+1$, all positions are correct. For two verifiers with no pre-omission ranking failure, both minima equal the first M_t. □

Corollary 1 (The proposal-limited regime)**.** *Suppose the true token is selected whenever it is available along every true prefix, for each of two verifiers sharing the proposals. Their first error equals $\tau_C = \min\{t : M_t\}$ and their exact-match indicators are identical. Their suffix accuracies after τ_C can differ arbitrarily subject to the available candidates.*

Proof. Every R_t^v is false, so Theorem 1 gives the common first omission. After the first wrong token, the theorem imposes no condition on the resulting incorrect-prefix proposal sets or scores. The pinned error itself remains and precludes exact match. □

The shared-fit experiment evaluates the events in this corollary rather than inferring them from equal EM. Its membership and selection counters report no correct-prefix ranking errors and identify the first omission as the first free-running error. These reported event counts identify the proposal-limited regime in the evaluated white-box comparisons. In calibrated transfer, conditional verification is below one, so the stronger sufficient premise no longer holds throughout the trace.

Corollary 2 (Candidate coverage bounds append-only search)**.** *Fix the fitted representation and a prefix-conditioned candidate operator $C_t(p)$. Let $\mathcal{P}_T$ be the complete paths reachable by repeatedly appending a token from $C_t(p)$. Then $x \in \mathcal{P}_T$ iff $x_t \in C_t(x_{<t})$ at every position. Consequently, any greedy, beam, or exhaustive prefix search restricted to this operator has*

$$\Pr(\widehat{x} = x) \leq \Pr\left[\bigcap_{t=1}^{T} \{x_t \in C_t(x_{<t})\}\right].$$

The bound holds for arbitrary path scores and beam widths, conditional on any fixed algorithmic randomness.

Proof. The path to x passes through $x_{<t}$ and appends x_t at each step. Every such edge exists precisely when its token is in $C_t(x_{<t})$. Induction gives the equivalence. A decoder restricted to this tree cannot output an unreachable path, and taking expectations gives the probability bound. □

The coverage event is necessary for any such prefix search, but a reachable path can still be discarded by pruning or lose under the final score. In the proposal-limited regime of Corollary 1, greedy verification attains this coverage ceiling on each request. Increasing beam width then cannot improve exact match without changing the expansion rule, although partial reconstructions can change. This explains what the candidate-budget and beam controls compare: new candidate identities versus additional paths through the existing candidate tree. The result is a consequence of constrained prefix search, not a guarantee that the measured decoder always attains its ceiling.

An event factorization, not independent tokens. Let E_t be the event that the first t variable tokens are correct, C_t true-token inclusion under the current prefix, and V_t successful verification. For any stochastic distribution over requests and fixed algorithmic randomness with positive conditioning probabilities,

$$\Pr(E_T) = \prod_{t=1}^{T} \Pr(C_t \mid E_{t-1}) \Pr(V_t \mid E_{t-1}, C_t). \tag{5}$$

This follows from $E_t = E_{t-1} \cap C_t \cap V_t$ and the chain rule. A factor is zero when the preceding event has positive probability but success is impossible; a zero-probability conditioning event can instead be handled by truncating at the first zero factor. Pooled token recall is not a substitute for these prefix-conditioned quantities. Raising a mean recall to the sequence length would introduce an independence assumption absent here.

B.4 Correlated score gaps and view selection

Let Z_A and Z_q be false-minus-true distance-score gaps for an existing verifier and a proposed view, conditional on a fixed candidate set and correct prefix. Write their means m_A, m_q. Assume their centered joint moment generating function obeys

$$\mathbb{E} \exp\left(t_A(Z_A - m_A) + t_q(Z_q - m_q)\right) \le \exp\left(\tfrac{1}{2}[t_A, t_q] \begin{bmatrix} v_A & c \\ c & v_q \end{bmatrix} [t_A, t_q]^\top\right) \tag{6}$$

for all real t_A, t_q, with a positive-semidefinite proxy matrix. This matrix is an assumption about candidate score errors, not the covariance of scalar residual magnitudes.

Proposition 3 (Normalized correlated-view criterion)**.** *For a normalized mixture $(1-\alpha)Z_A + \alpha Z_q$, let $m_\alpha = (1-\alpha)m_A + \alpha m_q$ and $v_\alpha = (1-\alpha)^2 v_A + 2\alpha(1-\alpha)c + \alpha^2 v_q$. If both are positive, its misranking probability is at most $\exp[-m_\alpha^2/(2v_\alpha)]$. For $m_A, v_A > 0$, the derivative of the exponent's positive magnitude at zero is strictly positive iff $m_q/m_A > c/v_A$.*

Proof. Apply Equation (6) in direction $[1-\alpha, \alpha]$. Chernoff's inequality gives $\exp(-tm_\alpha + t^2 v_\alpha/2)$ for $t \ge 0$, minimized at $t = m_\alpha/v_\alpha$. Differentiating the logarithm of m_α^2/v_α at zero gives $2(m_q/m_A - c/v_A)$. □

The criterion permits correlated errors and the implementation's renormalized view weights. It improves a stated upper bound, not necessarily the empirical error. A positive-mean gap with zero proxy variance has zero error in this ideal model; a nonpositive mean receives no useful one-sided guarantee. A finite candidate set can be handled by a union bound over its competitors, always conditional on that set.

The stacked-response convention. Each $S_q \in \mathbb{R}^{d \times B}$ contains the same ordered substitution probes as columns. The implementation forms $F_A = [S_q^\top]_{q \in A} \in \mathbb{R}^{B \times d|A|}$ by horizontal concatenation. Its probe Gram matrix is

$$F_A F_A^\top = \sum_{q \in A} S_q^\top S_q \in \mathbb{R}^{B \times B}.$$

Thus the planner compares probe responses jointly across the selected depths in a fixed B-dimensional space. Both endpoints are included before the greedy internal-depth selection. This convention differs from summing feature-space Gram matrices $S_q S_q^\top$; the latter is not the implemented criterion.

Probe information is a different objective. For fixed response matrices and $K_A = \epsilon I_B + \sum_{q \in A} S_q^\top S_q$, the determinant lemma yields

$$f(A \cup \{q\}) - f(A) = \log\det(I_d + S_q K_A^{-1} S_q^\top). \tag{7}$$

In an isotropic linear Gaussian probe channel, half this expression is conditional information about the latent probe coefficient. Conditioning on the fixed endpoints, the normalized internal-depth set function remains monotone and submodular, so its fixed-cardinality greedy solution has the standard $1 - 1/e$ bound (Nemhauser et al., 1978; Krause et al., 2008). This is a guarantee for the fixed probe matrices, not for the recovered private sequence.

To see the distinction, let a new response Gram matrix be a positive multiple of an existing nonzero one. The ridge log-det marginal is positive, even though its span need not introduce a new direction. If a new candidate score is itself a positive affine transform of the existing score, every candidate ranking is unchanged under a positive normalized mixture. A positive proxy gain therefore cannot replace the empirical fixed-placement comparison.

B.5 Certified evaluation of a fixed PAD score

Proposition 4 (Partial-score certificate)**.** *Fix candidates, target statistics, and nonnegative weights. Suppose every per-view score lies in* $[-1, 1]$*. For candidate c, let* p_c *be the weighted sum of evaluated terms and* r_c *the total unevaluated weight. The full score is in* $[L_c, U_c] = [p_c - r_c, p_c + r_c]$*. If* $L_{c^*} > \max_{c \neq c^*} U_c$*, then* c^* *is the unique full-score maximizer. Discarding candidates with* $U_c < \max_d L_d$*, and otherwise completing evaluation with the original tie rule, returns the same token as full evaluation.*

Proof. Every missing weighted term is bounded by its nonnegative weight in absolute value, giving the interval. A strict separation implies $g(c^*) > g(c)$ for every competitor. An eliminated candidate cannot tie or exceed the lower bound already attained by a survivor, so it cannot become a full-score maximizer. If ambiguity remains, completing the surviving scores and applying the original tie order resolves it identically. □

The rule is the bounded-aggregation principle (Fagin et al., 2003) applied to PAD. Induction on the *computed* prefix proves sequence equivalence: if two decoders have generated the same prefix, they propose the same candidates, and the certified evaluator returns the same next token. The prefix need not be correct. This guarantee concerns agreement with the original decoder, not the truth of the request.

With a common evaluated view set, a partial lead greater than $2r$ suffices. The threshold is sharp given only independent range bounds: assign every unevaluated leader score -1 and every competitor score $+1$. If the evaluated weight is no greater than the remaining weight, its maximal possible partial gap cannot strictly exceed $2r$. Thus certification may only become possible late in a depth-ordered pass. A smaller query count also need not mean fewer transformer operations when one forward supplies several scores. Model-level timing and a production-compatible numerical enclosure remain separate from this exact-arithmetic result.

C Protocol, Algorithms, and Reproducibility

The protocol below distinguishes the permitted observation interface from archived extension runs. The author's corrected black-box replay confirms extra planner access in historical TAP-Self; those results are retained as diagnostics, not as compliant executions of that interface.

C.1 Observation and model identity

A run's boundary label a–b identifies the input hook of block a and the output hook of block b. The resident trajectory has $L = b - a + 1$ executed blocks. Block indices are zero-based; a block's input precedes its computation and its output follows it. Candidate activations are generated by running the known prefix before this window and the selected local blocks.

Table 8 fixes the depth notation. Checkpoint replay passes for Qwen2.5-7B/3B/32B, Llama-3.1, Gemma-2, and Phi-3.5. Qwen3 fails two of three replay checks and is therefore excluded from current selector and cross-model inference. Its original numerical grid remains separately available. The original 14B benchmark is included; the later selector replication does not include 14B.

C.2 White-, gray-, and black-box access

White-box reconstruction evaluates candidates with the known victim checkpoint. The retained local window specifies which downstream features enter the objective. The all-downstream experiment in Table 36 evaluates a larger computational scope using the same received message.

Table 8: Breadth-benchmark capture windows. Each position is an independent attack run. Inclusive windows are not a claim that one deployed partition assigns the same block to two participants.

Model	Blocks	Position 2	Position 3
Qwen2.5-3B	36	12–23	23–35
Qwen2.5-7B	28	9–18	18–27
Qwen2.5-14B	48	16–31	31–47
Qwen2.5-32B	64	21–42	42–63
Llama-3.1-8B	32	10–21	21–31
Gemma-2-9B	42	14–27	27–41
Phi-3.5-mini	32	10–21	21–31
Qwen3-8B (separate record)	36	12–23	23–35

In gray box, the victim contains an unknown low-rank adapter and the inverse uses the public base checkpoint. As in the calibrated black-box extension, the observer supplies the designated victim target states, including internal and output states used by the trajectory attack. These states cannot in general be recovered by replaying the base model from one victim boundary.

The recorded victim adaptation uses Qwen2.5-7B-Instruct with rank-16 LoRA on the query and value projections, scaling parameter 32, and zero adapter dropout. Training uses AdamW with learning rate 10^{-4}, 120 steps, batch size 4, maximum length 128, and fp16. The training pool contains 4,000 Skytrax reviews with at least 80 characters, shuffled with seed 123. Reviews matching the first 60 characters of a test or validation request are excluded. This is the recorded overlap check, rather than a semantic deduplication guarantee.

The low-rank residual calibration (LRC) variant estimates rank-16 drift bases from 200 public prompts of at most 64 tokens. The recorded basis keys are the input of block 18 and outputs of blocks 21, 24, and 27. At selected depths with a basis, the supplied implementation adds a position-shared correction $\mathbf{1}(U_q c_q)^\top$ to the candidate state before standardization. Each c_q starts at zero and shares SGD, learning rate, and step count with the continuous embeddings. PAD subsequently scores uncorrected base-model candidate states with the fixed target statistics; the fitted residual is not passed to PAD. Tables 24 and 57 distinguish base-only and LRC results.

The black-box experiment uses Qwen2.5-7B as victim and Qwen2.5-3B as the inverse model, with a shared tokenizer. The victim weights are hidden from the reconstruction process, but an execution observer can retain the designated input and output states. Public calibration prompts are known to the attacker, so their victim states can be paired with surrogate states. For the private target request, the same observation interface provides states but not text. This permits surrogate-only optimization without giving the attacker the victim checkpoint.

At each calibrated depth, collect centered victim and surrogate rows in $X_q \in \mathbb{R}^{n_q \times d_v}$ and $Y_q \in \mathbb{R}^{n_q \times d_s}$, where $d_v \geq d_s$. If $X_q^\top Y_q = U_q \Sigma_q V_q^\top$ is a thin singular value decomposition, then

$$R_q = U_q V_q^\top \quad \text{maximizes} \quad \operatorname{tr}(R^\top X_q^\top Y_q) \quad \text{subject to} \quad R^\top R = I. \tag{8}$$

The affine map restores the surrogate mean after transforming the centered victim state. Candidate computation and gradients use the surrogate. This is rectangular trace-maximizing Procrustes alignment (Schönemann, 1966); the least-squares variant instead minimizes the unweighted residual over an unconstrained map. Appendix G retains both recorded alignment variants.

For main fitting and PAD on a private request, each target is a recorded victim state mapped at its corresponding depth. It is not generated by passing the projected input boundary through the surrogate. In white box the receiver can replay the local trajectory from its boundary; in calibrated black box the matched target states are part of the specified activation interface.

Table 9 gives the exact correspondence. The 36-block surrogate has more output depths in its window than the 28-block victim, so three victim depths map to two surrogate depths. Each map is fitted and stored in fp32 from width 3,584 to width 2,048. At application, the reference code first casts its coefficients to the state dtype, performs the matrix product in fp32, and casts the result back. The same mapped victim boundary is used for fitting and restart selection.

The default projection calibration set contains 200 Skytrax reviews with at least 60 characters, excluding first-60-character matches to the 150 test and 20 validation requests. Paired activations

Table 9: Role-matched depths for the calibrated black-box experiment. Block indices are zero-based. Affine maps are fitted and stored in fp32, from victim width 3,584 to surrogate width 2,048. Repeated victim depths map to distinct surrogate depths.

State type	Victim: Qwen2.5-7B	Surrogate: Qwen2.5-3B
Block input	18	23
Block output	18	23
Block output	19	24
Block output	20	25
Block output	20	26
Block output	21	27
Block output	22	28
Block output	22	29
Block output	23	30
Block output	24	31
Block output	25	32
Block output	26	33
Block output	26	34
Block output	27	35

from all positions, up to 64 tokens per prompt, are pooled at each matched depth. The calibration-size experiment uses nested subsets of 25, 50, 100, and 200 with the same matched-depth convention; its 200-prompt condition reuses the archived map. The victim window is 18–27, while the 3B surrogate window is 23–35. Black-box PIA in the five-method table is the feature-standardized single-boundary implementation, matching the first row of the controlled view ladder. The raw/std factorial is a separate comparison.

Planning access. Permitted-target TAP uses the same mapped victim endpoint observations for its warm-up as for reconstruction. Its probes depend only on provisional tokens and reconstruction-model computations. A privileged-planner diagnostic instead constructs warm-up targets from private-text reconstruction-model states; plain log-det selects directly from such states. These diagnostics have additional information and are excluded from equal-access attack claims. Fixed boundary, dual, triple, and uniform controls bypass planning. In ordinary white box, the victim and reconstruction model coincide.

Gray LRC, defense and shuffled-order diagnostic rows are not evidence for a fully access-compliant attack under their transformed interfaces: that interpretation requires the declared transformations throughout both planning and reconstruction. They are identified separately from the permitted-target black-box result.

C.3 Target statistics and optimization configuration

Table 10 specifies the canonical reconstruction. Target means and standard deviations are fixed during the main fit and verification. Standard deviations use one degree-of-freedom correction. Warm-up uses the two raw endpoints and projects its output to tokens for planning.

Initialization. Every trainable element is drawn independently from a fixed symmetric uniform distribution, not from a coordinate-dependent range. The draw is made in fp32 using PCG64 and then cast to the model precision. Warm-up and main optimization begin from the identical tensor. Only the projected warm-up tokens carry over to anchor selection; the converged warm-up embedding does not carry over to the main fit. The main-table generator seed is $s + i$ for run seed s and zero-based request index i. The resource experiments use $10000 + 100s + j$ for outer seed s and restart index j. Canonical runs use $s = 0$, the five-initialization study uses $s = 0, \ldots, 4$, and the resource study uses $s = 0, 1, 2$. Greedy token selection introduces no sampling seed.

Probe and candidate construction. The planner fixes the input hook of block a and the output hook of block b. Eligible internal hooks are block outputs $a+1, \ldots, b-1$; the output of block a is not an additional candidate. Thus selection measures complementarity relative to the same endpoints used by the fit. For $T \geq 2$, the reference implementation takes $P = \min(4, T)$ evenly spaced zero-based

Table 10: Canonical reconstruction configuration. The warm-up selects the trajectory; the main fit restarts from the same initial tensor.

Quantity	Setting
Main steps / endpoint warm-up	2,000 / 1,000
Initialization per trainable entry	$\mathcal{U}(-.1, .1)$, fp32 draw then cast
Default learning rate	.1
Internal anchors	2, in addition to both endpoints
Endpoint / internal weight	1 / .5 before normalization
Range-penalty coefficient	.001
Standard-deviation floor / cosine safeguard	10^{-6} / 10^{-8}
Probe ridge / response-norm safeguard	.001 / 10^{-8}
Probe allocation	4 positions, 4 substitutions per position
Embedding / LM candidates	10 / 10, plus the projected token
Vocabulary coordinate range	Tenth-largest signed value per coordinate
Embedding clipping	$[-.2, .2]$ before each loss evaluation
SGD momentum / weight decay	0 / 0
Gradient clipping	Disabled
Position mask	$m_t = 1$
Main initialization after warm-up	Original tensor restored; optimized warm-up not reused
Warm-up standardization / PAD	Off / off
Default numerical type / attention	fp16 / scaled dot-product attention
Candidate batch / verification passes	64 / 1
Canonical device	One NVIDIA H20

positions from 1 to $T - 1$, rounds to the nearest integer, and removes duplicates. At each position it first collects cosine-nearest alternatives to the provisional token, then fills from next-token proposals of the reconstruction model. The original token, special tokens, invalid vocabulary rows, and duplicate candidates are excluded. If fewer than four alternatives remain, a fixed PCG64 seed-zero vocabulary sampler fills the set, with at most 50 draws. Only the first four alternatives are retained, so four valid embedding neighbors already fill the probe budget. The same reconstruction model supplies PAD's LM proposals; no LM proposals are added at position zero. PAD enumerates embedding neighbors, LM proposals, and the projected token in that order, removes duplicates, and retains the first maximum under an exact score tie. TAP scans depths in increasing order and likewise retains the first maximum. These tie rules apply to the supplied candidate order; equal-valued top-K entries retain the underlying library's ordering.

For batched PAD evaluation, the reference implementation substitutes each candidate into the full current sequence but scores only its current-position state. Causal attention makes that state equal to the corresponding prefix-only computation in Algorithm 2. This equivalence concerns the score, not the number of token positions computed by the implementation.

Optimization details. Warm-up assigns weight one to each raw endpoint without normalizing their sum. Main fitting normalizes the selected anchor weights. The range threshold r_j is computed from signed embedding values before applying the absolute-value penalty to the trainable embeddings. Each iteration clips those trainable entries to $[-.2, .2]$, evaluates the loss, and takes an SGD step; a fixed BOS embedding is excluded from the trainable tensor. These rules match the supplied white-box reference implementation.

Hardware and numerical execution. Experiments run on NVIDIA H20 accelerators; the host has eight devices, and the canonical runs use one. The environment is Python 3.12.11, NumPy 1.26.0, PyTorch 2.7.1 with CUDA 12.8 and cuDNN 9.7.0, and Transformers 4.54.0.dev0. The attack uses fp16 and scaled dot-product attention (SDPA). Geometric Jacobian–vector-product checks instead use fp32 with eager attention. The separately retained Qwen3 records use bf16. These precision and attention choices are specified independently of library defaults.

Our PIA baseline is an adaptation of the two-stage method to the shared reconstruction engine. It uses the same tokenization, range penalty, cosine loss, and candidate procedure as the conditioned controls, with a raw received-boundary target. Thus these comparisons isolate modifications within a common implementation while retaining the two-stage structure of the original PIA method.

Tuning and run identity. The main canonical learning rate is .1. The ECHR comparison uses an independent ECHR validation set, choosing .5 for Boundary (std.) and .1 for ARIAD. CMS retains a transferred airline-selected rate because its original corpus was unavailable. The newer 20-prompt validation grids are separately reported in Table 19; the selected Skytrax configurations are instead evaluated on an additional disjoint 150-request test (Table 2). They are not substituted retroactively into prior test rows. Similarly, the measured $k = 2$ default is not presented as a validation-established optimum.

C.4 Data and metric identity

The standard datasets contain 150 requests each, with up to 64 tokens. ECHR uses its English fact text in paragraph order. CMS uses the archived processed descriptions rather than a newly collected CMS corpus. The long-context study uses 80 requests at 256 tokens. Generated canaries comprise 150 requests with annotated sensitive fields. Public calibration and adaptation data are separate from the target requests according to the evaluation protocol.

The latest breadth re-export uses token-ID exactness: after excluding a genuinely fixed tokenizer-inserted prefix token, the remaining predicted sequence must have the same length and every ID must match. Positional accuracy uses the number of target positions as denominator, with missing positions incorrect. Token F1 measures lexical overlap; BLEU-4 measures four-gram agreement (Papineni et al., 2002). NERR counts complete recovered ground-truth entity surfaces, using the archived Flair evaluator (Akbik et al., 2019). It is not the classifier's token accuracy or the canary's field-level exactness.

The original breadth and matched-control tables use token-ID exactness. On canonical PIA this gives .200 rather than the earlier evaluator's .220; Boundary (std.) and ARIAD remain at .540. Historical shuffling, canary, and single-realization defense tables retain their explicitly labeled evaluation convention. Comparisons use results from the same convention.

NERR denominators differ by the archived tokenized requests and entity outputs: 147 entity-bearing Skytrax requests in the canonical study, 148 in ECHR, and only five in CMS. Gemma and Phi breadth records report 146 and 145, respectively. These denominators are carried with the rows, not normalized to 150. For token-shuffled requests, intact-span NERR is not applicable under the fixed ground-truth entity definition. Only 203 of the 474 natural entity spans survive verbatim after shuffling, so that score would confound destruction of the reference spans with reconstruction. The four shuffled NERR entries are therefore omitted rather than set to zero.

C.5 Uncertainty and resource accounting

Fixed-initialization intervals use 10,000 paired prompt resamples, seed zero. In the factorial design, one resample carries all arms for the same prompt, rather than treating repeated arm–prompt contributions as independent observations. Repeated-initialization summaries average seed means equally. The supplied crossed intervals resample seed IDs and prompt IDs as two shared axes, keeping methods paired; seed-only intervals use paired seed differences. This follows the distinction between crossed variation and independent observations (Efron, 1979; Owen, 2007). With three seeds, neither a large collection of tokens nor a narrow conditional prompt interval provides many independent initialization replications.

Fixed-random anchors are drawn once for a run and reused for all its prompts. The six-placement study varies the draw while retaining initialization. The three-initialization selector study keeps the random placement fixed. Repeating a placement by chance is a valid draw outcome, but not a new anchor configuration. Defense noise seeds are a further, distinct source of randomness.

Resource comparisons include target preparation, optimization, candidate generation, verification, and restart selection in the stated full-request timing. Mean-time operating points and cap-specific overrun/timeout counts are reported separately; neither is inferred from the other. The separate shared-decoder times cover discretization of a cached fit and are not added to unrelated historical timing batches.

C.6 ALGORITHMS

Algorithm 1 specifies complete reconstruction using permitted target views; the historical extension departures are identified above. Algorithm 2 exposes the distinction between fitting and verification. Algorithm 3 is an optional, correctness-preserving evaluator for the same fixed scores. Appendix H.7 compares its recorded output and execution cost with full scoring.

Algorithm 1 Conditioned trajectory reconstruction

Require: Permitted target activations, victim model or calibrated surrogate, capture window, configuration

Acquire target views at the permitted hooks; apply per-depth maps in black box
Draw Z_0 with independent entries from $\mathcal{U}(-.1, .1)$
Fit a copy of Z_0 to the two raw endpoints for 1,000 steps, without standardization or PAD
Project to provisional tokens $\widetilde{x}$
Form normalized finite-difference substitution responses S_q; set $\mathcal{A} \leftarrow \{0, L\}$
Greedily add k internal views to $\mathcal{A}$ by the probe-Gram log-det gain
Compute and freeze target statistics for the selected views
Normalize the anchor weights; restore $Z \leftarrow Z_0$ for the main fit
for each main optimization step **do**
    Clip trainable embedding entries to $[-.2, .2]$
    Run the candidate prefix and selected trajectory; evaluate Equation (1)
    Update the trainable embeddings with SGD
end for
Cache raw projection and embedding-neighbor candidates
Decode with Algorithm 2, using $\mathcal{A}_{\mathrm{dec}} = \mathcal{A}$

Algorithm 2 Causal verification with a fixed fitted representation

Require: Fixed Z, projected token and embedding neighbors, target views, weights, LM, tie order

Initialize the recovered prefix with any fixed tokenizer-inserted token
for each variable position t **do**
    Query LM proposals from the recovered prefix
    $\mathcal{C}_t \leftarrow \mathcal{C}_t^{\mathrm{emb}} \cup \mathcal{C}_t^{\mathrm{LM}} \cup \{\widehat{x}_t^{\mathrm{raw}}\}$
    for candidate $c \in \mathcal{C}_t$ **do**
        Run its causal prefix; score $g_t(c) = \sum_{q \in \mathcal{A}_{\mathrm{dec}}} w_q s_{tq}(c)$
    end for
    Append the maximizer with the fixed tie rule
end for
return the recovered token sequence

Algorithm 3 Exact bounded-score evaluation of one PAD decision

Require: Fixed ordered candidates, nonnegative normalized weights, deterministic scores in $[-1, 1]$

Initialize partial sums $p_c = 0$, remaining weights $r_c = 1$, and all candidates active
for each positive-weight view in the chosen order **do**
    Query scores for active candidates; update p_c and r_c
    Set $L_c = p_c - r_c$, $U_c = p_c + r_c$
    Remove c only if $U_c < \max_d L_d$
    if one candidate remains or some $L_{c^*} > \max_{c \neq c^*} U_c$ **then**
        return the certified winner
    end if
end for
return the full-score maximizer among survivors with the original tie order

The mathematical checks cover finite score tables, local expansions, and first-failure identities. The score-trace comparisons and runtime measurements in Appendix H.7 evaluate the corresponding model implementation. Main recovery results use full verification.

D Complete Breadth Benchmarks

The following grids retain every method and all five recovery metrics from the latest breadth re-export. They expand the latest token-ID evaluation in Table 1. There are 29 recorded setting blocks with five methods each, including the separately retained Qwen3 rows. Each valid comparison uses the same observed boundary and target requests within its setting.

D.1 Participant count

Table 11 expands the main participant-count table to all learned and optimization-based attacks. Three-participant results already appear in Table 1. The four- and five-participant groups are arranged horizontally, so later positions can be compared without repeating their group label in every row.

Table 11: Complete participant sweep, five methods and five metrics. Labels show participant count/position and inclusive capture window.

Metric	Method	4/P2 7–14	4/P3 14–20	4/P4 20–27	5/P2 5–11	5/P3 11–16	5/P4 16–22	5/P5 22–27
Acc.	$\mathcal{A}_1$	.886	.869	.854	.891	.876	.864	.856
	GEIA	.014	.014	.009	.012	.009	.013	.013
	Vec2Text	.002	.003	.002	.006	.002	.003	.000
	PIA	.982	.965	.931	.998	.974	.945	.929
	ARIAD	**.995**	**.981**	**.968**	**.999**	**.983**	**.980**	**.956**
EM	$\mathcal{A}_1$	.000	.000	.000	.000	.000	.000	.000
	GEIA	.000	.000	.000	.000	.000	.000	.000
	Vec2Text	.000	.000	.000	.000	.000	.000	.000
	PIA	.807	.493	.147	.927	.640	.300	.040
	ARIAD	**.873**	**.680**	**.467**	**.947**	**.713**	**.580**	**.267**
F1	$\mathcal{A}_1$	.872	.854	.839	.877	.861	.849	.837
	GEIA	.183	.198	.171	.194	.177	.197	.196
	Vec2Text	.088	.100	.094	.125	.122	.016	.036
	PIA	.983	.973	.947	.998	.980	.955	.938
	ARIAD	**.996**	**.983**	**.967**	**.999**	**.984**	**.979**	**.955**
BLEU-4	$\mathcal{A}_1$	.746	.709	.678	.756	.723	.701	.675
	GEIA	.030	.032	.029	.031	.029	.031	.032
	Vec2Text	.001	.001	.001	.013	.003	.001	.000
	PIA	.975	.950	.907	.997	.964	.923	.891
	ARIAD	**.992**	**.967**	**.937**	**.998**	**.969**	**.958**	**.909**
NERR	$\mathcal{A}_1$	.469	.428	.383	.460	.449	.398	.388
	GEIA	.029	.021	.019	.034	.029	.016	.022
	Vec2Text	.005	.003	.003	.003	.003	.005	.002
	PIA	.986	.936	.795	**1.000**	.959	.889	.820
	ARIAD	**.986**	**.956**	**.940**	**1.000**	**.963**	**.953**	**.926**

ARIAD's deepest-window improvement reaches intact entities as well as correct positions. Nonetheless, the positionwise classifier can recover many local identities while reconstructing no full prompt. The learned decoders have low positional accuracy in this 2,000-example adaptation setting, even when their lexical-overlap metrics remain nonzero. The stronger learning-data study in Appendix G is important for interpreting these rows: limited adaptation is a resource choice, not a general inability of generative inversion to use activations.

D.2 Model size

Table 12 preserves both depths for every Qwen2.5 size. The 14B rows use the original benchmark outputs. The later selector study covers the checkpoints available for that separate replication.

Size is not a monotone predictor of recovery across all settings. At the shallow boundary, PIA has higher positional accuracy than ARIAD on 3B, 14B, and 32B, whereas ARIAD has higher exact match. The differences locate errors differently across sequences rather than producing a single

Table 12: Model-size benchmark with paired columns for the two attacker positions.

Model		Qwen-3B		Qwen-7B		Qwen-14B		Qwen-32B	
Attacker position		2	3	2	3	2	3	2	3
Boundary layer		12–23	23–35	9–18	18–27	16–31	31–47	21–42	42–63
Acc.	$\mathcal{A}_1$	.887	.880	.878	.858	.876	.857	.873	.851
	GEIA	.007	.017	.011	.014	.016	.015	.018	.014
	Vec2Text	.030	.004	.000	.006	.001	.001	.003	.001
	PIA	**.977**	.914	.982	.937	**.979**	.846	**.972**	.859
	ARIAD	.967	**.940**	**.986**	**.975**	.977	**.911**	.964	**.895**
EM	$\mathcal{A}_1$	.007	.000	.000	.000	.000	.000	.007	.000
	GEIA	.000	.000	.000	.000	.000	.000	.000	.000
	Vec2Text	.000	.000	.000	.000	.000	.000	.000	.000
	PIA	.327	.027	.673	.200	.360	.013	.273	.027
	ARIAD	**.427**	**.107**	**.740**	**.540**	**.473**	**.073**	**.320**	**.047**
F1	$\mathcal{A}_1$	.874	.865	.863	.841	.862	.841	.858	.834
	GEIA	.189	.194	.167	.201	.200	.210	.196	.199
	Vec2Text	.255	.089	.033	.126	.078	.093	.077	.086
	PIA	**.979**	.916	.983	.951	**.980**	.863	**.969**	.884
	ARIAD	.969	**.935**	**.988**	**.974**	.978	**.909**	.965	**.894**
BLEU-4	$\mathcal{A}_1$	.751	.729	.727	.683	.727	.681	.717	.671
	GEIA	.032	.032	.026	.032	.032	.034	.032	.032
	Vec2Text	.031	.003	.000	.006	.001	.001	.001	.001
	PIA	**.961**	.838	.969	.916	**.956**	.755	**.933**	.780
	ARIAD	.940	**.871**	**.976**	**.950**	.955	**.824**	.930	**.799**
NERR	$\mathcal{A}_1$	.467	.434	.444	.390	.445	.382	.427	.377
	GEIA	.017	.022	.018	.032	.024	.015	.015	.018
	Vec2Text	.078	.003	.000	.005	.000	.000	.003	.002
	PIA	**.948**	.822	.973	.830	**.995**	.753	**.992**	.763
	ARIAD	.931	**.878**	**.982**	**.950**	.980	**.829**	.964	**.834**

total ordering. At the deeper boundary, the complete ARIAD pipeline improves accuracy and entity recovery over raw PIA in the listed size settings. Repeated comparisons against Boundary (std.) remain the more relevant attribution test for the added trajectory.

D.3 Model family

Table 13 makes the architecture dependence visible. Llama's generative baseline has substantially higher lexical overlap than its positional accuracy, illustrating that reconstructing many words is not equivalent to putting them back in their original token positions. On deep Llama, Vec2Text exceeds ARIAD on F1, BLEU-4, and NERR while having lower positional accuracy. This separates lexical-content reconstruction from positional identity.

Phi's large difference between raw PIA and ARIAD should be read together with its repeated Boundary (std.) control. Once conditioning is supplied, the simple boundary objective is stronger than the full trajectory in the new Phi replication. The conditioned-boundary control isolates the added trajectory from the initial scale correction.

Table 13: Model-family benchmark with paired columns for the two attacker positions.

Model		Qwen2.5		Llama-3.1		Gemma-2		Phi-3.5	
Attacker position		2	3	2	3	2	3	2	3
Boundary layer		9–18	18–27	10–21	21–31	14–27	27–41	10–21	21–31
Acc.	$\mathcal{A}_1$	.878	.858	.830	**.857**	.877	.854	.878	.866
	GEIA	.011	.014	.036	.050	.030	.036	.019	.015
	Vec2Text	.000	.006	.504	.400	.019	.026	.030	.002
	PIA	.982	.937	**.998**	.674	**.982**	.806	.761	.228
	ARIAD	**.986**	**.975**	.998	.774	.976	**.880**	**.995**	**.893**
EM	$\mathcal{A}_1$	.000	.000	.000	.000	.000	.000	.000	.007
	GEIA	.000	.000	.000	.000	.000	.000	.000	.000
	Vec2Text	.000	.000	.000	**.007**	.000	.000	.000	.000
	PIA	.673	.200	**.947**	.000	**.553**	.007	.000	.000
	ARIAD	**.740**	**.540**	.900	**.007**	.427	**.013**	**.860**	**.047**
F1	$\mathcal{A}_1$	.863	.841	.815	.840	.853	.829	.854	.840
	GEIA	.167	.201	.185	.222	.193	.191	.190	.173
	Vec2Text	.033	.126	.894	**.869**	.086	.129	.215	.071
	PIA	.983	.951	**.998**	.692	**.983**	.820	.760	.249
	ARIAD	**.988**	**.974**	.998	.783	.977	**.878**	**.994**	**.885**
BLEU-4	$\mathcal{A}_1$	.727	.683	.628	.677	.697	.644	.712	.679
	GEIA	.026	.032	.030	.037	.032	.030	.036	.035
	Vec2Text	.000	.006	.786	**.727**	.004	.009	.030	.000
	PIA	.969	.916	**.997**	.451	**.965**	.653	.573	.064
	ARIAD	**.976**	**.950**	.994	.598	.951	**.761**	**.989**	**.783**
NERR	$\mathcal{A}_1$	.444	.390	.365	.453	.446	.397	.440	.445
	GEIA	.018	.032	.028	.098	.029	.034	.019	.035
	Vec2Text	.000	.005	.669	**.642**	.001	.006	.096	.000
	PIA	.973	.830	**.990**	.428	**.984**	.744	.519	.112
	ARIAD	**.982**	**.950**	.988	.551	.978	**.835**	**.989**	**.719**

D.4 CHECKPOINT-INCOMPATIBLE ROWS

The older Qwen3 values are reproduced in Table 14 for continuity. They are not pooled with verified checkpoint results and do not count as current model-wide wins. Matching architecture dimensions is insufficient when a replay reproduces different token outputs under otherwise identical settings.

Table 14: Qwen3 measurements with unsuccessful checkpoint replay, excluded from validated cross-model and selector comparisons.

Metric	Method	P2: 12–23	P3: 23–35
Acc.	$\mathcal{A}_1$	.876	.865
	GEIA	.012	.011
	Vec2Text	.005	.009
	PIA	.793	.513
	ARIAD	.837	.699
EM	$\mathcal{A}_1$	.000	.000
	GEIA	.000	.000
	Vec2Text	.000	.000
	PIA	.000	.000
	ARIAD	.007	.000
F1	$\mathcal{A}_1$	.861	.848
	GEIA	.175	.193
	Vec2Text	.117	.146
	PIA	.795	.532
	ARIAD	.840	.696
BLEU-4	$\mathcal{A}_1$	.721	.697
	GEIA	.029	.031
	Vec2Text	.005	.010
	PIA	.619	.249
	ARIAD	.695	.468
NERR	$\mathcal{A}_1$	.435	.400
	GEIA	.032	.025
	Vec2Text	.007	.008
	PIA	.602	.299
	ARIAD	.703	.551

E Strong Boundary Controls and Initialization

E.1 Conditioned controls and distinct evaluation cohorts

The original fixed-configuration domain comparison in Table 15 reports all five metrics. ECHR's .0332 accuracy advantage for ARIAD coexists with lower EM (.4133 versus .4533), and CMS favors Boundary (std.) on four metrics. These fixed-initialization comparisons are distinct from both repeated-initialization corrections and the additional validation-selected test.

Table 15: Standardized-boundary controls. All use 2,000 main steps; ARIAD also performs its planning warm-up. ECHR learning rates: .5 for Boundary (std.) and .1 for ARIAD; CMS uses the Skytrax-selected rate .1. Independent validation: Table 19.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
Q7D	PIA	.937	.200	.951	.916	.830
	Boundary (std.)	.961	.540	.966	.938	.922
	ARIAD	.975	.540	.974	.950	.950
ECHR	Boundary (std.)	.940	.453	.941	.884	.951
	ARIAD	.973	.413	.968	.937	.962
CMS	Boundary (std.)	.984	.767	.983	.966	.600
	ARIAD	.982	.707	.974	.946	.600

E.2 Repeated-initialization comparisons

Table 16 reports paired accuracy contrasts. Qwen-7B uses five paired initializations and the extensions use three, each on the same 150 requests per setting. A seed changes the optimizer initialization, not the request sample. Positive intervals on Qwen-7B, ECHR, and Qwen-3B coexist with a negative Phi contrast; a planner-level gain does not establish full-pipeline dominance.

Table 16: Paired accuracy contrasts over the prescribed initialization sets, with 95% paired seed–prompt bootstrap intervals (Appendix C).

Setting	ARIAD minus	Acc. difference	95% interval
Qwen-7B	PIA	+.0368	[+.0251,+.0486]
Qwen-7B	Boundary (std.)	+.0076	[+.0021,+.0131]
Qwen-7B	Dual endpoint	+.0069	[+.0014,+.0124]
ECHR	Boundary (std.)	+.0364	[+.0249,+.0482]
ECHR	Dual endpoint	+.0072	[+.0011,+.0134]
Qwen-3B	Boundary (std.)	+.0387	[+.0250,+.0521]
Qwen-3B	Dual endpoint	+.0066	[+.0008,+.0129]
Phi	Boundary (std.)	-.0524	[-.0630,-.0417]

Table 17 reports absolute outcomes and EM counts. ARIAD recovers 455 of 750 evaluations, versus 424 for Boundary (std.) and 426 for dual endpoints. Table 18 gives their paired differences alongside the retained negative Phi and CMS results. These pooled counts do not imply that the 750 evaluations are independent requests.

Table 17: Core white-box outcomes. EM pools five initializations on the same 150 requests. Accuracy is reported separately; a dash denotes an unavailable absolute value. No SD is inferred from pooled exact counts.

Method	Acc.	Exact / 750	EM
Boundary (std.)	—	424	.5653
Dual endpoint	.9737	426	.5680
ARIAD	.9806	455	.6067

Table 18: Paired exact-match contrasts with 95% intervals. The core uses five initializations; Phi and CMS use three. Point differences agree with the core pooled exact counts, while intervals use paired outcomes.

Setting	ARIAD minus	ΔEM	95% interval
Qwen-7B	Boundary (std.)	+.0413	[+.0108,+.0716]
Qwen-7B	Dual endpoint	+.0387	[+.0071,+.0699]
Phi	Boundary (std.)	-.0822	[-.1356,-.0333]
CMS	Boundary (std.)	-.0867	[-.1556,-.0178]

Intervals follow the paired seed–prompt resampling protocol in Appendix C. Pooled exact counts determine the displayed rates, while uncertainty depends on paired outcomes and variation across initializations. Neither token counts nor repeated evaluations of the same requests provide additional independent request samples.

E.3 Validation selection and independent testing

Table 19 preserves the independent 20-request learning-rate grids. On Skytrax, validation accuracy selects .5 for PIA and .3 for Boundary (std.), dual endpoints and ARIAD. Those configurations were frozen before an additional disjoint 150-request test, reported in Table 2. All methods use one fixed initialization. The respective test accuracies are .9794, .9762, .9741 and .9869; exact request counts are 84, 104, 78 and 101. Thus ARIAD has the highest positional accuracy, while Boundary (std.) recovers three more whole requests. No paired interval or repeated-initialization estimate is inferred for this additional test.

Table 19: Independent learning-rate validation: all four methods and four rates on 20 prompts per domain. These are validation outcomes, not matched test reruns.

Dataset	Method	Rate	Acc.	EM	F1	BLEU-4	NERR
Skytrax	PIA	0.03	.887	.050	.905	.827	.704
	PIA	0.1	.924	.200	.942	.902	.907
	PIA	0.3	.958	.550	.970	.948	.889
	PIA	0.5	.980	.550	.981	.969	1.000
	Boundary (std.)	0.03	.911	.250	.926	.871	.643
	Boundary (std.)	0.1	.962	.500	.965	.928	.949
	Boundary (std.)	0.3	.976	.700	.982	.960	.982
	Boundary (std.)	0.5	.902	.650	.920	.864	.907
	Dual endpoint	0.03	.915	.100	.924	.868	.782
	Dual endpoint	0.1	.967	.450	.965	.937	.961
	Dual endpoint	0.3	.973	.500	.970	.943	.968
	Dual endpoint	0.5	.967	.650	.965	.934	.958
	ARIAD	0.03	.913	.100	.923	.857	.764
	ARIAD	0.1	.930	.250	.929	.867	.949
	ARIAD	0.3	.983	.650	.985	.974	.982
	ARIAD	0.5	.974	.550	.974	.948	1.000
ECHR	PIA	0.03	.885	.000	.882	.790	.789
	PIA	0.1	.934	.350	.937	.882	.948
	PIA	0.3	.959	.550	.960	.922	.948
	PIA	0.5	.983	.650	.978	.959	.979
	Boundary (std.)	0.03	.930	.150	.931	.867	.935
	Boundary (std.)	0.1	.958	.500	.953	.903	.928
	Boundary (std.)	0.3	.961	.550	.966	.934	.991
	Boundary (std.)	0.5	.975	.700	.973	.948	.977
	Dual endpoint	0.03	.968	.250	.961	.916	.878
	Dual endpoint	0.1	.981	.550	.979	.956	.949
	Dual endpoint	0.3	.981	.550	.974	.950	.988
	Dual endpoint	0.5	.958	.550	.958	.928	.921
	ARIAD	0.03	.965	.150	.958	.913	.845
	ARIAD	0.1	.974	.500	.977	.957	.973
	ARIAD	0.3	.981	.650	.978	.960	.990
	ARIAD	0.5	.970	.600	.961	.930	.882

F Controlled Attribution and First-Error Diagnostics

F.1 Fitting, placement, and verification

The contrasts in Table 4 separate the contributions of internal coverage, adaptive placement, and verification. Uniform internal views improve accuracy over endpoints by .0074 [.0016,.0132]. TAP improves over uniform and fixed-random placement by .0068 [.0011,.0125] and .0079 [.0019,.0139], respectively. These positive intervals establish gains for the fixed-initialization comparisons. Holding the fit fixed, all-view verification improves accuracy by .0052 [.0034,.0070] and NERR by .0056 [.0009,.0103], with zero EM change.

F.2 Factorial estimands and paired comparisons

The design crosses raw or target-standardized features with boundary, endpoint, fixed-random, uniform-internal, or TAP-selected fitted views. Internal-view fits are each decoded using the boundary alone or all fitted views. The conditioning effect averages eight standardized-minus-raw pairs; coverage averages two uniform-minus-endpoint pairs; TAP–uniform and TAP–random each average four matched pairs; verification averages six shared-fit pairs. Each request contributes its paired arm outcomes before averaging across requests. Thus 1,200, 300, 600, and 900 arm–prompt contributions do not represent those numbers of independent requests.

Conditioning retains the measured EM effect .2275 [.1900,.2633]. Table 20 reports independently defined entity contrasts: the boundary-only NERR contrast (.0925) and pooled standardization effect (.0660) average different comparisons. The 147 entity-bearing requests are kept paired.

Table 20: Boundary-specific entity contrasts, evaluated separately from pooled factorial NERR effects. The contrast direction follows its method labels.

Contrast	ΔNERR	95% CI
Boundary standardization	+.0925	[+.0493, +.1382]
Endpoints - boundary (std)	+.0390	[+.0036, +.0753]
TAP - uniform (std, all)	-.0129	[-.0444, +.0165]

F.3 Which event creates the first error?

The event summary records 521 jointly failed pair–prompt outcomes sharing the first candidate omission. All six paired suffix-accuracy intervals are positive (Table 21). Together with the zero EM difference in Table 4, these results distinguish recovery after an error from complete-prompt exactness.

Table 21: Shared-fit first-error and suffix summaries. A pair–prompt record compares the two verifiers on one request with a shared fit. Suffix intervals concern all-view minus boundary-only accuracy after the shared first error.

Diagnostic	Confirmed result
Joint failures sharing the first candidate omission	521 records
Positive suffix-accuracy intervals	6 of 6 pairs

Teacher forcing is used only to diagnose the recorded reconstruction. Replaying the true prefix identifies whether the true token was proposed and selected; the free-running record identifies the actual first error. Comparing these events tests the proposal-limited regime directly. Each request–pair uses one fitted continuous tensor, shared raw projection, embedding neighbors, and target statistics. Changing the verifier leaves that fit unchanged. Theorem 1 relates the events without requiring the true-token score to equal one.

Table 22: Candidate coverage on the true prefix. Each pair evaluates 9,418 token positions; both verifiers share its candidate set. Both verifiers select every available true token; coverage is measured by candidate membership.

Pair	Emb. hits	LM hits	Raw hits	Union hits	Correct given union	Ranking failures
P01	8247	6326	6450	9187	9187	0
P02	8329	6326	6587	9200	9200	0
P03	8307	6326	6582	9171	9171	0
P04	8560	6326	7095	9280	9280	0
P05	8541	6326	7042	9269	9269	0
P06	8556	6326	7098	9288	9288	0

Table 22 retains the original membership and selection counters. The LM branch contributes 6,326 hits in each fixed-prefix pair, while embedding-neighbor and raw-projection hits vary with the fit. Their union covers 9,171–9,288 of 9,418 positions. Every available true token is selected in these counters. The two decoders can therefore agree on the true prefix while reacting differently after it becomes wrong. Table 23 retains the positional decomposition and denominators of that trace export.

F.4 What changes after the shared failure?

The six-of-six suffix result identifies a benefit that EM cannot express: an early error prevents complete exactness, but later private content can still be recovered more accurately. The six-of-six summary reports the direction of the paired intervals; it does not quantify the distribution of suffix effect sizes.

The suffix comparison conditions on jointly failed prompts with the same first error. Changing the fit or proposal budget can move that first error and change the cohort. The candidate and restart

Table 23: True-prefix coverage and free-running first-error counts by position block. Verification conditions on candidate-union membership; unequal block counts reflect request lengths.

Pair	Positions	Tokens	Union recall	Boundary V	All-view V	First errors (b/a)
P01	1-16	2400	.9267	1.0000	1.0000	94/94
P01	17-32	2400	.9908	1.0000	1.0000	5/5
P01	33-48	2376	.9920	1.0000	1.0000	5/5
P01	49-64	2242	.9938	1.0000	1.0000	3/3
P02	1-16	2400	.9329	1.0000	1.0000	86/86
P02	17-32	2400	.9900	1.0000	1.0000	6/6
P02	33-48	2376	.9933	1.0000	1.0000	3/3
P02	49-64	2242	.9924	1.0000	1.0000	6/6
P03	1-16	2400	.9267	1.0000	1.0000	86/86
P03	17-32	2400	.9862	1.0000	1.0000	7/7
P03	33-48	2376	.9916	1.0000	1.0000	4/4
P03	49-64	2242	.9920	1.0000	1.0000	5/5
P04	1-16	2400	.9717	1.0000	1.0000	39/39
P04	17-32	2400	.9862	1.0000	1.0000	19/19
P04	33-48	2376	.9912	1.0000	1.0000	6/6
P04	49-64	2242	.9929	1.0000	1.0000	5/5
P05	1-16	2400	.9675	1.0000	1.0000	48/48
P05	17-32	2400	.9862	1.0000	1.0000	17/17
P05	33-48	2376	.9899	1.0000	1.0000	6/6
P05	49-64	2242	.9938	1.0000	1.0000	6/6
P06	1-16	2400	.9725	1.0000	1.0000	41/41
P06	17-32	2400	.9900	1.0000	1.0000	11/11
P06	33-48	2376	.9895	1.0000	1.0000	15/15
P06	49-64	2242	.9933	1.0000	1.0000	5/5

studies therefore report full-request outcomes rather than treating their suffix averages as the same comparison.

G GRAY- AND BLACK-BOX RECONSTRUCTION AND DECODER CONTROLS

Permitted-target black-box TAP plans from calibrated victim observations and reaches .7647 accuracy with zero complete-prompt EM. Privileged-planner TAP instead uses private-text reconstruction-model states during warm-up; plain log-det reads such states directly. Those variants, gray LRC, and their decoding follow-ups are diagnostic comparisons, excluded from equal-access conclusions. Fixed boundary, dual, triple, and uniform controls bypass planning. Appendix C.2 defines the interfaces.

G.1 WHITE-, GRAY-, AND BLACK-BOX COMPARISON

Table 24 compares five methods across the access regimes in Section 3.1. The white-box checkpoint, gray-box base model, and black-box surrogate define each inverse. In the black-box comparison, the classifier leads on accuracy while the privileged-planner ARIAD diagnostic recovers more complete entities.

Table 24: Complete white-, gray-, and black-box comparison. The black-box interface uses public paired activations for calibration, as defined in Section 3.1. Gray/black ARIAD entries use privileged planning and are diagnostic. Boundary denotes PIA in white and gray box, and its standardized implementation in black box. The victim is Qwen2.5-7B, the surrogate is Qwen2.5-3B, and calibration uses 200 public prompts at 14 matched depths.

Metric	Method	White-box	Gray-box	Black-box
Acc.	$\mathcal{A}_1$	.858	.874	**.862**
	GEIA	.014	.012	.007
	Vec2Text	.006	.007	.004
	Boundary	.937	.888	.524
	ARIAD	**.975**	**.906**	.735
EM	$\mathcal{A}_1$	.000	.000	.000
	GEIA	.000	.000	.000
	Vec2Text	.000	.000	.000
	Boundary	.200	.000	.000
	ARIAD	**.540**	**.040**	.000
F1	$\mathcal{A}_1$	.841	.858	**.848**
	GEIA	.201	.176	.128
	Vec2Text	.126	.098	.072
	Boundary	.951	.903	.550
	ARIAD	**.974**	**.904**	.728
BLEU-4	$\mathcal{A}_1$	.683	.714	**.697**
	GEIA	.032	.030	.026
	Vec2Text	.006	.008	.000
	Boundary	.916	**.830**	.289
	ARIAD	**.950**	.820	.501
NERR	$\mathcal{A}_1$	.390	.414	.395
	GEIA	.032	.018	.013
	Vec2Text	.005	.003	.003
	Boundary	.830	.773	.251
	ARIAD	**.950**	**.800**	**.488**

The historical least-squares and Procrustes comparisons in Table 25 use the same calibrated victim-target interface, with extra planner access in the TAP pipeline. Procrustes produces higher positional and entity recovery for both recorded optimization pipelines. This motivates separating alignment fit from the angular candidate comparisons used for reconstruction, without establishing compliant TAP transfer.

Table 26 compares permitted-target TAP with the archived placement diagnostics. Permitted-target TAP reaches .7647 accuracy, versus .7492 for uniform placement, a point difference of .0155; both have zero complete-prompt EM. Uniform NERR is .3187. Permitted-target TAP's F1, BLEU-4, NERR, variability, and timing were not supplied, so the record does not support a matched-timing or statistical-significance claim. Privileged TAP-Self (.7346) and plain log-det (.7270) retain their

Table 25: Black-box alignment variants. Single-initialization comparison of unconstrained least squares and Procrustes alignment, with 150 requests. Both boundary baselines use feature standardization. Five-method Procrustes results: Table 24. ARIAD uses privileged planning; its entries are diagnostic.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
Least squares	Boundary (std.)	.392	.000	.442	.169	.078
	ARIAD	.642	.000	.637	.370	.269
Procrustes	Boundary (std.)	.524	.000	.550	.289	.251
	ARIAD	.735	.000	.728	.501	.488

privileged-planner diagnostic status and are excluded from the permitted-target comparison. Gray-box uniform, TAP-Self, and plain log-det attain .9058, .9059, and .9013 accuracy, respectively, with different planner access; their ranking alone does not isolate the causal value of placement.

Table 26: Black-box anchor placement. Procrustes calibration, 150 requests, one initialization. † denotes private-text planning diagnostics, excluded from equal-access comparisons. Dashes denote unavailable metrics. No significance or matched-latency claim is made.

Placement	Acc.	EM	F1	BLEU-4
Uniform	.7492	.0000	.7352	.5148
TAP, permitted targets	.7647	.0000	—	—
Privileged TAP-Self†	.7346	.0000	.7279	.5014
Plain log-det†	.7270	.0000	.7144	.4859

Within the gray privileged-planner diagnostic, LRC modestly increases exact recovery for ARIAD (.040 to .053), while reducing accuracy (.906 to .903) and F1 (.904 to .900; Table 57). It does not close the gap to the white-box result or validate the warm-up's access.

G.2 View coverage and public calibration data

Table 27 compares the archived black-box view-coverage configurations for the same requests. The three-view variant uses both endpoints and a fixed middle view; the archived four-view ARIAD variant additionally uses privileged TAP planning. Its selected pair need not contain that fixed middle depth, so this is neither a strictly nested view-count intervention nor a matched-access test of the last increment. The fixed controls and permitted-target TAP result appear in Table 5.

Table 27: View-coverage configurations under the activation-calibrated black-box interface. Scaling and calibration are fixed; the four-view variant also changes placement through TAP planning. The four-view ARIAD row uses privileged planning (†); it is diagnostic. Values are point estimates; the archived adaptive row changes both placement and planning access.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
1 view	Boundary	.524	.000	.550	.289	.251
2 views	Endpoints	.666	.000	.654	.400	.329
3 views	Fixed triple	.714	.000	.708	.479	.405
4 views	ARIAD†	.735	.000	.728	.501	.488

The boundary-to-dual accuracy difference is .1424 (.5239 to .6663); the fixed triple reaches .7135 and uniform four-view placement reaches .7492. Their successive accuracy differences are .0472 and .0357. The permitted-target TAP point estimate is .0155 above uniform, without a reported paired interval or timing comparison. This separates fixed view coverage from the old privileged four-view row in Table 27; that archived row and its entity metrics do not describe permitted-target TAP. A separately recorded middle-only variant uses one internal target and reaches .6754 accuracy with zero EM, compared with .6663 for dual endpoints.

Table 28 varies the number of public paired prompts while holding the test set fixed. The subsets are nested and the same depth map is used throughout. ARIAD's privileged-planner mean accuracy with

25 calibration prompts exceeds the standardized boundary's with 200, but its entity recovery at 25 remains lower. These recorded calibration contrasts do not establish the public-data requirement of a compliant TAP attack.

Table 28: Calibration-size sweep under the fixed same-tokenizer 7B-to-3B interface. Public calibration subsets are nested; test requests are unchanged. ARIAD entries use privileged planning; they do not establish deployable black-box calibration efficiency. Entries are per-initialization scores; calibration-sample uncertainty is not estimated.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
PROC	Boundary (std.)	25 / s0	.258	.000	.298	.088	.055
	ARIAD	25 / s0	.537	.000	.540	.255	.151
	Boundary (std.)	25 / s1	.255	.000	.292	.087	.059
	ARIAD	25 / s1	.527	.000	.536	.250	.133
	Boundary (std.)	25 / s2	.263	.000	.314	.097	.053
	ARIAD	25 / s2	.535	.000	.543	.255	.165
	Boundary (std.)	50 / s0	.356	.000	.403	.153	.111
	ARIAD	50 / s0	.628	.000	.626	.353	.264
	Boundary (std.)	50 / s1	.353	.000	.396	.149	.096
	ARIAD	50 / s1	.625	.000	.629	.361	.201
	Boundary (std.)	50 / s2	.371	.000	.409	.156	.132
	ARIAD	50 / s2	.618	.000	.612	.343	.247
	Boundary (std.)	100 / s0	.444	.000	.488	.220	.179
	ARIAD	100 / s0	.698	.000	.690	.452	.383
	Boundary (std.)	100 / s1	.458	.000	.487	.221	.203
	ARIAD	100 / s1	.682	.000	.676	.428	.366
	Boundary (std.)	100 / s2	.444	.000	.482	.214	.145
	ARIAD	100 / s2	.686	.000	.682	.434	.346
	Boundary (std.)	200 / s0	.524	.000	.550	.289	.251
	ARIAD	200 / s0	.735	.000	.728	.501	.488
	Boundary (std.)	200 / s1	.510	.000	.545	.280	.255
	ARIAD	200 / s1	.714	.000	.707	.479	.422
	Boundary (std.)	200 / s2	.511	.000	.535	.269	.264
	ARIAD	200 / s2	.727	.000	.712	.484	.440

G.3 Candidate coverage is a measurable resource

Table 29 reports white-box union recall. EM is .540/.720 for the 10/10 and 20/20 budgets.

Table 29: Post-correction candidate-union coverage on a fixed white-box fit. Both proposal branches expand together. True-prefix union recall is a token-level diagnostic, distinct from full-request exact match.

K_{emb}/K_{LM}	Union recall
10/10	.9858
20/20	.9919

Table 30 varies the two proposal budgets with the continuous fit and anchors fixed. The raw projected token remains in every union, including when one top-K branch is disabled. This isolates changes in candidate construction from changes in optimization.

On white-box Qwen-7B, increasing the union from 10/10 to 20/20 raises exact recovery by .180. The true-prefix union recall changes from .9858 to .9919. The true-prefix diagnostics give conditional verification accuracy one for both settings. These pooled recalls do not imply independent token errors; the sequence-level effect depends on where the remaining omissions occur. Under calibrated transfer, conditional verification is below one and declines slightly as more alternatives are considered. A larger set can both add the correct identity and create more difficult competition for it.

Table 30: Candidate budgets with the fitted embeddings and anchors fixed. The condition is $K_{\mathrm{emb}}/K_{\mathrm{LM}}$; the raw candidate is always retained. PROC entries condition on a privileged-planner fit and are decoder diagnostics.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
Q7D	ARIAD	10/0	.889	.013	.878	.758	.847
	ARIAD	0/10	.923	.060	.922	.846	.887
	ARIAD	5/5	.960	.253	.959	.918	.928
	ARIAD	10/10	.975	.540	.974	.950	.950
	ARIAD	20/10	.980	.613	.979	.958	.951
	ARIAD	10/20	.983	.640	.983	.967	.954
	ARIAD	20/20	.987	.720	.987	.975	.951
PROC	ARIAD	10/0	.477	.000	.436	.149	.315
	ARIAD	0/10	.489	.000	.489	.205	.255
	ARIAD	5/5	.635	.000	.627	.355	.398
	ARIAD	10/10	.735	.000	.728	.501	.488
	ARIAD	20/10	.774	.000	.768	.564	.531
	ARIAD	10/20	.784	.000	.778	.587	.545
	ARIAD	20/20	.814	.000	.808	.635	.579

G.4 Searching prefixes versus proposing identities

Table 31 reports true-prefix union recall and verification conditional on coverage for each candidate budget. Coverage is the fraction of correct-prefix positions whose true token enters the union. Conditional verification measures correct selection within that covered set. These quantities distinguish candidate availability from the ability to rank it.

Table 31: Candidate-budget diagnostics on true prefixes. Coverage and conditional verification have different denominators; all rates are reported directly. PROC entries condition on a privileged-planner fit. Q7D uses 9,418 token positions; PROC uses the same count. One PROC first-omission mean has an unresolved index convention and is excluded from first-error conclusions.

Setting	K_{emb}/K_{LM}	Union recall	Verify | covered	Mean first omission	PAD (s)
Q7D	10/0	.9089	1.0000	15.967	5.09
Q7D	0/10	.9422	1.0000	15.233	6.49
Q7D	5/5	.9730	1.0000	29.207	6.05
Q7D	10/10	.9858	1.0000	41.820	10.25
Q7D	20/10	.9880	1.0000	45.660	14.67
Q7D	10/20	.9891	1.0000	46.287	14.58
Q7D	20/20	.9919	1.0000	50.040	19.08
PROC	10/0	.5270	.9692	1.433	2.58
PROC	0/10	.7305	.9769	0.780	4.39
PROC	5/5	.7744	.9763	1.980	4.00
PROC	10/10	.8495	.9679	2.667	6.06
PROC	20/10	.8744	.9654	3.880	8.16
PROC	10/20	.8851	.9653	3.207	8.12
PROC	20/20	.9040	.9636	4.693	10.14

Table 32 holds the proposal branches and fitted representation fixed while changing beam width. On Qwen-7B, a four-wide beam raises positional accuracy from .9753 to .9785 but leaves EM at .5400. It does not recover the .180 EM gain produced by enlarging the candidate union. In transfer, beam four raises accuracy from .7346 to .7464, while NERR falls from .4884 to .4657. More paths are not a uniformly better privacy reconstruction under every metric.

The beam is scored by accumulated PAD agreement, not by a ground-truth oracle. This score choice does not remove the transfer fit's earlier privileged planning. A path with a high summed activation score can differ from the one recovering the most entities, especially under an imperfect calibration map. The path score and the content-recovery metric measure different properties of the sequence.

Table 32: Beam width with fixed continuous reconstruction. Accumulated PAD scores select prefixes; beam one is greedy. PROC entries condition on a privileged-planner fit and are decoder diagnostics.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
Q7D	ARIAD	beam1	.975	.540	.974	.950	.950
	ARIAD	beam2	.976	.540	.975	.952	.952
	ARIAD	beam4	.979	.540	.976	.952	.953
PROC	ARIAD	beam1	.735	.000	.728	.501	.488
	ARIAD	beam2	.742	.000	.735	.514	.468
	ARIAD	beam4	.746	.000	.739	.515	.466

G.5 A SHARED DISCRETE INTERFACE FOR FIXED CONTINUOUS FITS

Tables 33, 34, and 35 use nearest projection, shared standardized-boundary PAD, and shared fixed-four-view PAD on each cached fit. The latter uses uniform internal hooks, so it is not a relabeling of the original TAP-selected pipeline. Candidate-generation budgets and rules are shared, but embedding neighbors still depend on the fit being decoded.

Table 33: Shared-decoder control, initialization 0. Each fitted embedding is evaluated by nearest projection and the same two PAD rules. ER uses 138 prompts versus 150 for other fits, precluding paired cross-cohort comparisons. Repeated ER vectors do not provide independent initialization evidence.

Continuous fit	Decoder	Acc.	EM	F1	BLEU-4	NERR	n
PIA	NN	.741	.000	.736	.522	.675	150
PIA	Boundary PAD	.917	.200	.934	.883	.823	150
PIA	Shared four-view PAD	.935	.200	.944	.907	.856	150
Boundary (std.)	NN	.798	.000	.790	.604	.779	150
Boundary (std.)	Boundary PAD	.961	.540	.966	.938	.922	150
Boundary (std.)	Shared four-view PAD	.971	.540	.972	.949	.926	150
Dual	NN	.758	.000	.741	.510	.763	150
Dual	Boundary PAD	.968	.540	.971	.944	.947	150
Dual	Shared four-view PAD	.975	.540	.973	.947	.961	150
ARIAD fit	NN	.753	.000	.737	.508	.762	150
ARIAD fit	Boundary PAD	.970	.540	.971	.944	.950	150
ARIAD fit	Shared four-view PAD	.976	.540	.975	.951	.950	150
ER (2k)	NN	.027	.000	.033	.001	.005	138
ER (2k)	Boundary PAD	.066	.000	.099	.014	.019	138
ER (2k)	Shared four-view PAD	.112	.000	.130	.022	.019	138
ER (10k)	NN	.161	.000	.124	.021	.068	138
ER (10k)	Boundary PAD	.470	.000	.521	.266	.245	138
ER (10k)	Shared four-view PAD	.548	.000	.569	.313	.275	138

The released ER fit benefits from either PAD variant. At 10,000 optimization steps, initialization zero gives accuracy .1614 under nearest projection, .4705 under the boundary verifier, and .5485 under the shared four-view verifier. This within-fit comparison isolates the importance of the discrete procedure. The shared decoder changes recovery substantially, while the remaining fit-specific differences motivate keeping continuous optimization and discretization as separate axes.

The two ER budgets use the same 138 requests for all three decoder comparisons. Twelve saved continuous outputs have a shape incompatible with their matched prompt and are excluded rather than truncated: request indices 23, 29, 35, 36, 47, 48, 64, 66, 94, 113, 123 and 124. The other methods cover all 150 requests, so a paired cross-method comparison would require restricting them to the shared 138. The reported within-fit decoder comparisons already use a common cohort. Repeated identical ER result vectors are not treated as independent initialization evidence. Decoder times exclude fitting; full-pipeline costs are in Appendix H.

Table 34: Shared-decoder control, initialization 1. Each fitted embedding is evaluated by nearest projection and the same two PAD rules. ER uses 138 prompts versus 150 for other fits, precluding paired cross-cohort comparisons. Repeated ER vectors do not provide independent initialization evidence.

Continuous fit	Decoder	Acc.	EM	F1	BLEU-4	NERR	n
PIA	NN	.725	.000	.721	.506	.612	150
PIA	Boundary PAD	.909	.153	.922	.864	.826	150
PIA	Shared four-view PAD	.929	.153	.932	.886	.851	150
Boundary (std.)	NN	.810	.000	.809	.626	.768	150
Boundary (std.)	Boundary PAD	.968	.547	.970	.943	.934	150
Boundary (std.)	Shared four-view PAD	.978	.547	.977	.955	.937	150
Dual	NN	.758	.000	.748	.531	.710	150
Dual	Boundary PAD	.964	.487	.967	.936	.909	150
Dual	Shared four-view PAD	.974	.487	.973	.947	.920	150
ARIAD fit	NN	.757	.000	.754	.531	.748	150
ARIAD fit	Boundary PAD	.980	.573	.983	.966	.936	150
ARIAD fit	Shared four-view PAD	.984	.573	.985	.970	.954	150
ER (2k)	NN	.027	.000	.033	.001	.005	138
ER (2k)	Boundary PAD	.066	.000	.099	.014	.019	138
ER (2k)	Shared four-view PAD	.112	.000	.130	.022	.019	138
ER (10k)	NN	.161	.000	.124	.021	.068	138
ER (10k)	Boundary PAD	.470	.000	.521	.266	.245	138
ER (10k)	Shared four-view PAD	.548	.000	.569	.313	.275	138

Table 35: Shared-decoder control, initialization 2. Each fitted embedding is evaluated by nearest projection and the same two PAD rules. ER uses 138 prompts versus 150 for other fits, precluding paired cross-cohort comparisons. Repeated ER vectors do not provide independent initialization evidence.

Continuous fit	Decoder	Acc.	EM	F1	BLEU-4	NERR	n
PIA	NN	.744	.000	.735	.527	.697	150
PIA	Boundary PAD	.916	.220	.936	.889	.826	150
PIA	Shared four-view PAD	.941	.220	.948	.909	.847	150
Boundary (std.)	NN	.807	.000	.803	.625	.773	150
Boundary (std.)	Boundary PAD	.968	.593	.971	.947	.946	150
Boundary (std.)	Shared four-view PAD	.975	.593	.975	.951	.945	150
Dual	NN	.762	.000	.751	.534	.760	150
Dual	Boundary PAD	.972	.567	.973	.951	.945	150
Dual	Shared four-view PAD	.977	.567	.978	.956	.951	150
ARIAD fit	NN	.747	.000	.729	.506	.742	150
ARIAD fit	Boundary PAD	.967	.513	.970	.942	.922	150
ARIAD fit	Shared four-view PAD	.972	.513	.971	.946	.931	150
ER (2k)	NN	.027	.000	.033	.001	.005	138
ER (2k)	Boundary PAD	.066	.000	.099	.014	.019	138
ER (2k)	Shared four-view PAD	.112	.000	.130	.022	.019	138
ER (10k)	NN	.161	.000	.124	.021	.068	138
ER (10k)	Boundary PAD	.470	.000	.521	.266	.245	138
ER (10k)	Shared four-view PAD	.548	.000	.569	.313	.275	138

G.6 Public computation and adaptation data

Table 36 tests the resource convention behind the local window. Computing all public downstream targets does not improve over the compact resident trajectory in this run: accuracy falls from .9786 to .9729 and EM from .5667 to .5000. The compact resident trajectory is therefore a useful computational choice in this run, even when later public blocks are available.

Table 37 gives the separately trained public-data controls. The positionwise attack rises from .9065 to .9723 accuracy when training examples increase from 2,000 to 10,000, and acquires .2667 EM. GEIA also improves substantially on accuracy and lexical overlap, while this Vec2Text adaptation does not. The learned inverse benefits from more public examples, whereas per-request optimization spends its resources on the observed target.

Table 36: Public downstream computation beyond the resident window. The observed packet is unchanged; computation and target views differ.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
Q7D	Boundary	seed0	.966	.587	.970	.945	.925
	Resident + two	seed0	.979	.567	.977	.955	.953
	All downstream	seed0	.973	.500	.973	.945	.928

Table 37: Public adaptation-data sweep using decoder checkpoints trained separately from the breadth benchmark. Training configurations differ by method. Inference times are unavailable.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
Q7D	$\mathcal{A}_1$	2000	.906	.000	.894	.792	.570
	GEIA	2000	.014	.000	.217	.033	.022
	Vec2Text	2000	.018	.000	.203	.021	.019
	$\mathcal{A}_1$	10000	.972	.267	.968	.932	.860
	GEIA	10000	.553	.000	.755	.535	.751
	Vec2Text	10000	.005	.000	.095	.004	.004

Table 38 reports the accompanying adaptation resources. The positionwise and generative adaptations have different validation-set sizes; the reported training/test overlap is zero for every row. These comparisons evaluate recovery at the stated adaptation budgets, without an inference-latency ranking.

Table 38: Adaptation resources for Table 37. Epoch denotes the selected checkpoint; validation-set sizes differ by method. All six configurations report zero training/test overlap.

Method	Training requests	Epoch	Validation requests	GPU hours
$\mathcal{A}_1$	2,000	4	12,226	.0438
GEIA	2,000	2	200	.0777
Vec2Text	2,000	23	200	.1565
$\mathcal{A}_1$	10,000	5	12,226	.2096
GEIA	10,000	9	200	.3763
Vec2Text	10,000	39	200	.7527

H Search Resources and Delivery Constraints

Tables 39 and 40 compare recovery–time operating points and cap-specific delivery. Each aggregate covers three initializations on the same 150 requests. The fixed-placement black-box comparison appears in Table 6.

H.1 Budgets and completed reconstructions

The recorded budget protocol uses two allowances, H_1 and $H_2 = 2H_1$, calibrated on 20 requests disjoint from testing. The smaller allowance is 1.05 times the largest complete ARIAD request time in that calibration set. For calibrated black box, the recorded limits are $H_1 = 223.8438$ seconds and $H_2 = 447.6875$ seconds. Restart selection uses the mean standardized-boundary cosine score of completed candidates, without a recovery metric, and is included in full-request timing.

Under the restart policy, each fit runs its fixed main-step count. A deadline is checked before and after a restart. A late restart is discarded although its elapsed time is counted, and fallbacks remain in the recovery denominator. Configurations use 2,000 main steps per restart; ARIAD uses a 1,000-step warm-up and 2,000-step main fit. The Q7D ARIAD profiles share the nominal settings: 1,000 warm-up steps, 2,000 main steps, 10/10 candidate budgets, and one ARIAD start. The two recovery–time profiles are descriptive operating points; the nominal settings alone do not define a switch between their implementations. Their cap-specific latency distributions are not inferred from mean times.

Table 39: Recovery and computation. Three initializations on the same 150 requests. Boundary is standardized; counts denote total starts. ARIAD profiles share 1,000 warm-up steps, 2,000 main steps, 10/10 proposals and one start. They are descriptive implementation profiles, without a specified switch between them. Exact counts are 360/450 for lower-cost Q7D and 196/450 and 202/450 for ECHR Dual and ARIAD. Other EM entries are rounded rates; — denotes an unavailable metric.

Setting	Strategy	Acc.	EM	Seconds
Q7D	Boundary, 1 start	—	.571	112.2
Q7D	Boundary, 2 starts	—	.811	222.6
Q7D	Boundary, 3 starts	—	.891	319.2
Q7D	**ARIAD**, recovery-gain setting	—	.851	220.4
Q7D	**ARIAD**, lower-cost setting	—	.8000	176.8
ECHR/H_2	Dual endpoint	.97463	.43556	115.9
ECHR/H_2	**ARIAD**	.98210	.44889	168.4

H.2 Canonical Qwen-7B

At the measured operating points, standardized Boundary reaches .571, .811, and .891 EM with one, two, and three starts, taking 112.2, 222.6, and 319.2 seconds per request. The ARIAD recovery-gain setting reaches .851 EM in 220.4 seconds: 4.0 percentage points above the two-start control with 2.2 seconds less mean time. These EM entries retain the reported rounded rates. The lower-cost ARIAD setting has a final exact count of 360/450, or .8000 EM, about 1.1 percentage points below the two-start control. Its mean time is 176.8 seconds, a reduction of 20.6%. These point estimates describe the observed recovery–time trade-off; no noninferiority interval is supplied.

H.3 ECHR

At H_2, ARIAD accuracy is .98210 versus .97463 for Dual endpoint, a difference of +.00747. The exact counts are 202/450 for ARIAD and 196/450 for Dual, giving EM .44889 and .43556 and a difference of 6/450, or +.01333. Their mean request times are 168.4 and 115.9 seconds, respectively, so ARIAD takes 45.3% longer.

H.4 Calibrated black-box reconstruction

The fixed-control comparison in Table 6 gives .6648 accuracy for Dual endpoint and .5973 for three-start Boundary, with paired difference +.0675 and reported 95% interval [.0581,.0769]. Mean times are 133.9 and 321.3 seconds. Both methods have zero prompt EM and zero overruns at the 447.7-second nominal cap. This comparison concerns endpoint coverage at the measured resource point; it does not evaluate adaptive internal placement.

The privileged-planner black-box ARIAD resource rows use privileged TAP planning and remain diagnostic. The fixed-control aggregates do not establish an adaptive result under permitted-target planning. Earlier accuracy and resource campaigns also used different checkpoint roots with the archived Procrustes map; the one-request canaries do not establish campaign-wide equivalence (Appendix C.2).

H.5 Elapsed time and timeout accounting

Table 40 reports zero timeouts and overruns for Q7D ARIAD at $H_1 = 185.31$ seconds, ECHR ARIAD at $H_1 = 182.56$ seconds, and three-start Q7D Boundary at $H_2 = 370.61$ seconds. These are reported event counts, not conclusions drawn from mean latency. Cap-specific mean and p95 times are not available for these conditions; they are not inferred from other operating points.

H.6 Separating total time from stage profiles

The total-time summaries pool three initializations. Stage profiles were collected in a separate instrumented run with seed zero. Tables 41, 42, and 43 report those profiles without combining them with totals from another run.

Table 40: Delivery under nominal allowances. Each condition comprises three initializations on the same 150 requests, or 450 seed–request evaluations. Counts concern actual timeout and elapsed-time overrun events; cap-specific mean and p95 latencies are unavailable.

Setting	Strategy	Cap	Limit (s)	Overrun	Timeout
Q7D	ARIAD	H_1	185.31	0/450	0/450
ECHR	ARIAD	H_1	182.56	0/450	0/450
Q7D	Boundary, 3 starts	H_2	370.61	0/450	0/450

Table 41: Instrumented seed-zero stage times and peak memory: Q7D. End-to-end resource times use separate runs. Preparation includes target construction, warm-up and probes; decoding includes proposals, PAD and start selection.

Cap	Strategy	Main (s)	Prep. (s)	Decode (s)	Peak GiB
H1	PIA	81.6	0.0	10.5	16.44
H1	Boundary, 1 start	93.2	0.0	13.3	16.45
H1	Boundary, 2 starts	202.3	0.1	30.8	16.45
H1	Boundary, 3 starts	186.2	0.0	27.6	16.45
H1	Dual endpoint	102.7	0.0	10.2	16.45
H1	ARIAD	107.1	52.5	10.4	16.45
H2	PIA	79.6	0.0	10.2	16.44
H2	Boundary, 1 start	81.1	0.0	10.2	16.45
H2	Boundary, 2 starts	160.3	0.0	20.4	16.45
H2	Boundary, 3 starts	245.4	0.1	30.6	16.45
H2	Dual endpoint	105.5	0.0	10.3	16.45
H2	ARIAD	104.1	51.1	10.3	16.45

Table 42: Instrumented seed-zero stage times and peak memory: ECH. End-to-end resource times use separate runs. Preparation includes target construction, warm-up and probes; decoding includes proposals, PAD and start selection.

Cap	Strategy	Main (s)	Prep. (s)	Decode (s)	Peak GiB
H1	PIA	79.2	0.0	10.3	16.44
H1	Boundary, 1 start	78.4	0.0	10.3	16.45
H1	Boundary, 2 starts	161.4	0.0	20.6	16.45
H1	Boundary, 3 starts	200.4	0.1	25.8	16.45
H1	Dual endpoint	104.6	0.0	10.4	16.45
H1	ARIAD	105.4	51.8	10.4	16.45
H2	PIA	80.1	0.0	10.3	16.44
H2	Boundary, 1 start	81.6	0.0	10.3	16.45
H2	Boundary, 2 starts	163.4	0.0	20.6	16.45
H2	Boundary, 3 starts	250.8	0.1	31.0	16.45
H2	Dual endpoint	108.2	0.0	10.4	16.45
H2	ARIAD	108.7	53.2	10.5	16.45

Table 44 supplies the complete additive profile for calibrated black box under H_2. Main fitting, preparation, decoding and residual bookkeeping add to the measured elapsed time of the same requests. For ARIAD, this total is 212.2489 seconds. The archived seed-zero total is 204.8985 seconds and its three-seed average is 202.5189 seconds. The same-seed difference is 7.3504 seconds, or 3.59%. Stage-level synchronization was disabled, so that difference is not identified as synchronization overhead. The profile describes where time was spent in its own run, not a decomposition of the archived mean.

H.7 Exact-score evaluation in practice

The bounded-score rule in Proposition 4 leaves the full objective and candidate set unchanged. Table 45 compares production full scoring, exact rational full scoring, exact certified scoring, and production lazy scoring on the recorded trajectories. There are no winner or sequence disagreements across 33,556 decisions. Truncating and resuming a candidate forward also preserves the recorded

Table 43: Instrumented seed-zero stage times and peak memory: PROC. End-to-end resource times use separate runs. ARIAD uses privileged planning; these are implementation profiles. Preparation includes target construction, warm-up and probes; decoding includes proposals, PAD and start selection.

Cap	Strategy	Main (s)	Prep. (s)	Decode (s)	Peak GiB
H1	PIA	99.7	0.0	6.1	21.39
H1	Boundary, 1 start	101.4	0.0	6.2	21.40
H1	Boundary, 2 starts	205.3	0.1	12.3	21.40
H1	Boundary, 3 starts	268.4	0.1	15.9	21.40
H1	Dual endpoint	134.4	0.0	6.2	21.40
H1	ARIAD	137.4	67.8	6.2	21.40
H2	PIA	104.8	0.0	6.2	21.39
H2	Boundary, 1 start	103.8	0.0	6.2	21.40
H2	Boundary, 2 starts	209.6	0.1	12.4	21.40
H2	Boundary, 3 starts	316.5	0.1	18.6	21.40
H2	Dual endpoint	136.4	0.0	6.2	21.40
H2	ARIAD	138.0	68.0	6.2	21.40

Table 44: An additive timing decomposition from one instrumented black-box run at the larger cap. All means use the same 150 requests and seed zero. ARIAD uses privileged planning; this table audits timing arithmetic. The ARIAD total (212.2489 s) is from a separate run from the resource study (seed zero: 204.8985 s; three-seed mean: 202.5189 s).

Strategy	Main (s)	Prep. (s)	Decode (s)	Other (s)	Sum (s)	Elapsed (s)
PIA	104.7666	0.0312	6.2249	0.0304	111.0531	111.0531
Boundary, 1 start	103.7637	0.0327	6.1979	0.0384	110.0327	110.0326
Dual endpoint	136.4402	0.0336	6.2035	0.0378	142.7151	142.7150
ARIAD	138.0179	67.9835	6.2095	0.0380	212.2489	212.2489
Boundary, 2 starts	209.6132	0.0531	12.3942	0.0751	222.1356	222.1355
Boundary, 3 starts	316.4662	0.0760	18.5848	0.1113	335.2383	335.2383

anchor values. These checks test the declared score contracts; agreement on the black-box traces does not certify the earlier planner's access.

Table 45: Full and bounded-score verification on shared trajectories. All comparisons have zero winner or sequence disagreements, score-range violations, ties, and fallbacks. PROC uses an archived privileged-planner trajectory; these are scoring-execution checks. Long-prefix uses the first 64 tokens of 80 requests; the 256-token recovery cohort is excluded.

Cell	Comparison	Requests	Decisions	Winner diff.	Sequence diff.
Qwen-7B	Production / exact full	150	9418	0	0
Qwen-7B	Exact full / certified	150	9418	0	0
Qwen-7B	Production / lazy	150	9418	0	0
ECHR	Production / exact full	150	9600	0	0
ECHR	Exact full / certified	150	9600	0	0
ECHR	Production / lazy	150	9600	0	0
Black box	Production / exact full	150	9418	0	0
Black box	Exact full / certified	150	9418	0	0
Black box	Production / lazy	150	9418	0	0
Long-prefix (64)	Production / exact full	80	5120	0	0
Long-prefix (64)	Exact full / certified	80	5120	0	0
Long-prefix (64)	Production / lazy	80	5120	0	0

The certification workload is capped at 64 tokens. The long-prefix cohort contains the first 64 tokens of each of 80 requests drawn from the long-prompt sample, for 5,120 decisions. It is separate from the full 256-token recovery experiment in Table 57. Qwen-7B and black-box requests range from 40 to 64 tokens; ECHR requests all have 64 tokens.

Table 46 reports the exact query denominators. The certified evaluator still uses .9954–1.0000 of all candidate–view scores and terminates early at only three positions. A third of the total weight lies on the deepest endpoint, leaving a wide possible score interval until that endpoint is evaluated. For a shared queried-view subset, the largest possible partial-score gap is twice its queried weight. It cannot exceed twice the remaining weight while the queried weight is at most one half.

Table 46: Exact score-query accounting for bounded-score evaluation. Counts cover candidate–view pairs, not transformer forward calls. PROC uses an archived privileged-planner trajectory; these are scoring-execution checks. Only 3 of 33,556 decisions terminate early; survivor counts do not measure avoided deep computation.

Cell	Decisions	Candidates	Full pairs	Queried	Queried/full	Early exits
Qwen-7B	9,418	177,490	709,960	707,672	0.9968	0
ECHR	9,600	177,380	709,520	706,232	0.9954	3
Black box	9,418	180,218	720,872	720,871	1.0000	0
Long-prefix (64)	5,120	96,620	386,480	385,316	0.9970	0

The recorded final-survivor count is one per decision. Dividing it by the candidate count gives approximately .053, but this counts winners, not skipped deep forwards. The near-complete query count is the relevant measure of score work. A narrower empirical score band would permit more exits, but its observed violations prevent it from serving as a valid certificate.

Tables 47 and 48 compare the production evaluator, a shared-cache full evaluator, and its certified counterpart at candidate batches one and eight. The shared-cache control separates certification from caching. The cached-full/certified PAD ratios in Table 49 range from .9794 to .9994, and no interval establishes acceleration. The exact rule preserves decisions, but the loose score bounds do not reduce execution cost on these workloads.

Table 47: Execution of full and certified scoring with matched candidate batches. Times are per request; intervals concern PAD, while total time includes the unchanged reconstruction fit. PROC uses an archived privileged-planner trajectory; these are scoring-execution checks. All rows have zero sequence disagreements. The shared-cache full evaluator controls for caching; the certified variant differs only in its evaluation schedule. Long-prefix (64) is the truncated certification cohort in Table 45.

Cell	Batch	Evaluator	PAD (s)	95% interval	Total (s)	10^6 block-tokens	Peak GiB
Qwen-7B	1	Production full	21.008	[20.410, 21.509]	177.464	2.056	14.310
Qwen-7B	1	Cached full	21.651	[21.037, 22.164]	178.107	2.056	14.310
Qwen-7B	1	Cached certified	21.682	[21.064, 22.200]	178.137	2.056	14.310
Qwen-7B	8	Production full	10.989	[10.561, 11.345]	167.444	2.056	14.418
Qwen-7B	8	Cached full	10.505	[10.101, 10.840]	166.960	2.056	14.418
Qwen-7B	8	Cached certified	10.628	[10.213, 10.976]	167.083	2.081	14.418
ECHR	1	Production full	21.289	[21.139, 21.447]	176.405	2.122	14.310
ECHR	1	Cached full	21.890	[21.740, 22.051]	177.006	2.122	14.310
ECHR	1	Cached certified	21.904	[21.755, 22.062]	177.019	2.122	14.310
ECHR	8	Production full	11.283	[11.242, 11.325]	166.399	2.122	14.452
ECHR	8	Cached full	10.780	[10.742, 10.820]	165.896	2.122	14.452
ECHR	8	Cached certified	11.007	[10.932, 11.085]	166.123	2.169	14.452

Table 48: Execution of full and certified scoring with matched candidate batches. Times are per request; intervals concern PAD, while total time includes the unchanged reconstruction fit. PROC uses an archived privileged-planner trajectory; these are scoring-execution checks. All rows have zero sequence disagreements. The shared-cache full evaluator controls for caching; the certified variant differs only in its evaluation schedule. Long-prefix (64) is the truncated certification cohort in Table 45.

Cell	Batch	Evaluator	PAD (s)	95% interval	Total (s)	10^6 block-tokens	Peak GiB
Black box	1	Production full	25.927	[25.296, 26.471]	220.250	2.706	20.180
Black box	1	Cached full	26.183	[25.553, 26.725]	220.507	2.706	20.180
Black box	1	Cached certified	26.218	[25.590, 26.760]	220.541	2.706	20.180
Black box	8	Production full	6.981	[6.775, 7.151]	201.305	2.706	20.207
Black box	8	Cached full	6.818	[6.621, 6.981]	201.141	2.706	20.207
Black box	8	Cached certified	6.824	[6.628, 6.985]	201.148	2.707	20.207
Long-prefix (64)	1	Production full	21.602	[21.472, 21.726]	176.655	2.166	14.310
Long-prefix (64)	1	Cached full	22.225	[22.091, 22.351]	177.277	2.166	14.310
Long-prefix (64)	1	Cached certified	22.250	[22.112, 22.379]	177.302	2.166	14.310
Long-prefix (64)	8	Production full	11.483	[11.448, 11.516]	166.536	2.166	14.423
Long-prefix (64)	8	Cached full	10.966	[10.933, 10.998]	166.019	2.166	14.423
Long-prefix (64)	8	Cached certified	11.072	[11.021, 11.124]	166.125	2.187	14.423

Table 49: Matched-cache timing ratios: cached-full time divided by certified time. A ratio above one indicates acceleration. PROC uses an archived privileged-planner trajectory; these are scoring-execution checks. The valid bounds preserve decisions but do not yield an end-to-end speedup on these workloads.

Cell	Batch	PAD ratio	95% interval	Total ratio	95% interval
Qwen-7B	1	.9986	[.9976, .9995]	.9998	[.9997, .9999]
Qwen-7B	8	.9884	[.9838, .9924]	.9993	[.9990, .9995]
ECHR	1	.9994	[.9986, 1.0002]	.9999	[.9998, 1.0000]
ECHR	8	.9794	[.9740, .9846]	.9986	[.9983, .9990]
Black box	1	.9987	[.9980, .9994]	.9998	[.9998, .9999]
Black box	8	.9991	[.9980, 1.0002]	1.0000	[.9999, 1.0000]
Long-prefix (64)	1	.9989	[.9980, .9997]	.9999	[.9997, 1.0000]
Long-prefix (64)	8	.9904	[.9873, .9934]	.9994	[.9992, .9996]

I Internal Placement, Anchor Count, and Calibration Scope

The primary selector comparisons concern untransformed white-box cells, where the victim and reconstruction model coincide. Two archived gray-box rows remain separate diagnostics. Their historical planning access is described in Appendix C.2; those rows are not combined with the white-box comparisons below.

I.1 Adaptive selection versus fixed schedules

Table 50 reports comparisons over random positions and initializations. On Phi, TAP improves accuracy over uniform placement by .0195 [.0064,.0326] and over fixed random placement by .0162 [.0084,.0240]. On Qwen-3B, the corresponding gains are .0073 [.0012,.0134] and .0068 [.0009,.0127]. All four intervals exclude zero. The extension supports the selected schedules over the stated fixed alternatives in these two model settings; it does not establish a gain at every placement or initialization.

The documented controls use six random placements with initialization fixed and three optimization initializations with the random placement fixed. They are distinct designs, not an eighteen-cell crossed experiment. The broader fixed-schedule grid provides context for the aggregate extension.

Table 51 gives the original fixed-placement comparison in 17 white-box cells and marks two gray-box diagnostics. Each fixed-random reference is sampled once per run, rather than anew for each request. Its prompt interval conditions on that placement. Phi deep has a positive TAP–uniform interval, while 32B deep has a negative TAP–random interval; most other intervals include zero. This broader grid provides context for the two-model extension without turning it into a universal placement claim. The mixed 19-cell aggregate is not used for a white-box conclusion.

Table 50: Selector comparisons with additional random placements and initializations. Entries are mean TAP minus comparator accuracy and their reported 95% intervals. These aggregate comparisons extend the earlier fixed-placement records.

Model	Comparator	Mean ΔAcc.	95% CI
Phi	Uniform	+.0195	[.0064, .0326]
Phi	Fixed random	+.0162	[.0084, .0240]
Qwen-3B	Uniform	+.0073	[.0012, .0134]
Qwen-3B	Fixed random	+.0068	[.0009, .0127]

Table 51: Selection at a fixed view count: 17 white-box cells and two gray-box diagnostics (†). Intervals condition on the recorded initialization and fixed-random placement. Gray-box rows are excluded from white-box conclusions because their reference warm-up omits the victim context. No pooled 19-cell claim is made.

Setting	TAP−uniform	95% CI	TAP−random	95% CI
N5 / P2	+.0004	[-.0008, +.0017]	+.0019	[+.0000, +.0043]
N4 / P2	+.0006	[-.0035, +.0047]	+.0003	[-.0038, +.0041]
N5 / P3	+.0038	[-.0049, +.0128]	-.0012	[-.0060, +.0038]
N4 / P3	-.0025	[-.0109, +.0058]	+.0029	[-.0050, +.0113]
N5 / P4	+.0029	[-.0048, +.0114]	+.0031	[-.0053, +.0121]
N4 / P4	+.0031	[-.0080, +.0146]	+.0031	[-.0080, +.0146]
N5 / P5	+.0053	[-.0075, +.0192]	-.0051	[-.0164, +.0060]
size3B / P2	-.0046	[-.0148, +.0054]	-.0031	[-.0130, +.0067]
size3B / P3	+.0104	[-.0009, +.0223]	+.0015	[-.0100, +.0129]
phi / P2	-.0004	[-.0033, +.0025]	+.0010	[-.0013, +.0034]
phi / P3	+.0306	[+.0130, +.0484]	+.0135	[-.0002, +.0273]
Gemma-2 / P2	-.0014	[-.0038, +.0008]	-.0007	[-.0036, +.0020]
Gemma-2 / P3	-.0018	[-.0121, +.0082]	+.0023	[-.0075, +.0118]
Llama-3.1 / P2	+.0009	[-.0001, +.0022]	+.0005	[-.0003, +.0014]
Llama-3.1 / P3	-.0065	[-.0197, +.0063]	+.0054	[-.0072, +.0179]
size32B / P2	+.0016	[-.0039, +.0071]	+.0073	[+.0000, +.0156]
size32B / P3	+.0019	[-.0097, +.0131]	-.0105	[-.0206, -.0007]
gray†	+.0001	[-.0109, +.0107]	-.0025	[-.0142, +.0088]
Gray LRC†	-.0024	[-.0141, +.0095]	+.0045	[-.0076, +.0170]

The separately exported difficult-subset aggregate is inconsistent with its three named cell means and remains outside these pooled comparisons, with its original values retained in the source record.

I.2 Placement and initialization variation

Fixed-random anchors are sampled once per run and shared across its prompts. The placement study uses six draws with initialization fixed; the initialization study uses three optimization seeds with a fixed placement. Repeated anchor sets are valid random draws but not distinct configurations. These designs distinguish placement variation from optimizer variation; they are not an eighteen-cell crossed study.

TAP adapts views to provisional tokens. Its advantage over fixed schedules does not isolate request-specific adaptation from finding generally useful depths. A transferred or globally frozen selected schedule would be needed for that attribution. The present claim concerns the selected trajectory's measured reconstruction performance.

I.3 Internal-anchor count

Table 52 gives the complete corrected $k = 1, 2, 4, 8$ results. The four accuracy values for TAP span .9718–.9764. Exact match varies across this narrow accuracy range, with $k = 1$ slightly above the default $k = 2$. The default is a compact evaluated choice rather than a uniquely optimal count.

Table 52: Internal-anchor count with a common budget of 2,000 main steps. Two internal anchors is the default; the sweep does not establish a validated optimum.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
$k = 1$	**TAP-Self**	.976	.567	.976	.954	.953
	Uniform	.972	.540	.972	.945	.962
$k = 2$	**TAP-Self**	.975	.540	.974	.950	.950
	Uniform	.975	.487	.976	.950	.963
$k = 4$	**TAP-Self**	.974	.567	.973	.949	.949
	Uniform	.972	.500	.971	.944	.932
$k = 8$	**TAP-Self**	.972	.500	.973	.948	.936
	Uniform	.971	.427	.969	.939	.931

Table 53: Independent anchor-count validation on 20 Skytrax prompts.

Setting	Method	Condition	Acc.	EM	F1	BLEU-4	NERR
Skytrax val.	**TAP-Self**	k=1	.965	.400	.969	.938	.893
	TAP-Self	k=2	.930	.250	.929	.867	.949
	TAP-Self	k=4	.977	.450	.972	.946	.893
	TAP-Self	k=8	.959	.450	.954	.917	.940
	Uniform-M	k=1	.967	.400	.966	.940	.912
	Uniform-M	k=2	.969	.300	.969	.939	.957
	Uniform-M	k=4	.968	.250	.967	.934	.857
	Uniform-M	k=8	.950	.250	.946	.904	.809

The independent anchor-count validation in Table 53 is a separate configuration-selection exercise. A preferred anchor count would need its own frozen test comparison before replacing a reported k-sweep row. The measured default remains two internal anchors throughout the main pipeline.

I.4 Relating probe geometry to recovery

The archived restricted-Jacobian measurements report higher overlap between internal responses and the endpoint span than between the second endpoint and the first. They also report internal ridge gains around 1.5–1.6, rather than nearly zero. Equation (7) explains why shared directions can still produce positive regularized gains. That evidence supports a redundancy ordering, not a numerical zero-gain approximation.

The proposed cross-setting explanation that greater layer-gain variability predicts a larger selector benefit was not established by the supplied geometry diagnostics. Similarly, residual-error headroom within a small same-model window sweep did not establish a placement law. The coverage ladder and the fixed-placement comparisons address two distinct questions: whether additional views help, and whether an input-dependent planner chooses better ones.

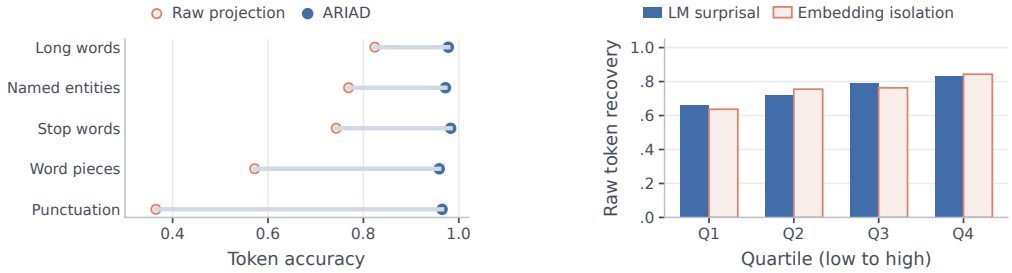
J Content, Sequence Length, and Protection

The ordinary content and length analyses are distinct from the transformed-observation and gray LRC diagnostics retained below. The reference warm-up omits victim, defense, and shuffling arguments; the author has confirmed the historical black-box defect, not the source of every transformed campaign. Without a verified compliant warm-up source, the defense and shuffled-order attack results do not establish deployable recovery under their stated transformed interfaces (Appendix C.2).

J.1 Lexical recovery and candidate identity

The original content diagnostic compares nearest projection after continuous fitting with the final candidate-verified output. Figure 5 preserves its token categories and independent lexical partitions. Long words and entities have higher raw recovery than punctuation and word pieces. The final

pipeline closes much of the gap. The comparison concerns recovery by the fitted inverse: a distinctive token can be identifiable from its activation even when it is difficult to predict from context.



(a) Projection and complete reconstruction.

(b) Independent lexical quartiles.

Figure 5: Predictability and activation identifiability differ. Token categories overlap. Isolation bins reverse the source's crowding order so that the displayed axis consistently runs from low to high isolation. This diagnostic uses the original content-analysis export.

Raw recovery increases from .662 to .830 across LM-surprisal quartiles. This is consistent with a token being hard to guess from a generic prefix but identifiable from its request-specific activation. It is not a causal intervention on surprisal. The reported joint frequency of "ARIAD correct and LM-only wrong" is .627, and the final attack recovers .960 of the 934 entity tokens missed by the LM-only predictor. Neither quantity is the recall of the full proposal union; the new candidate traces measure that separately.

J.2 Fresh identifiers and their limits

Tables 54 and 55 report overall prompt metrics and exact annotated fields. The corrected ARIAD Email count is 26/32, versus 19/32 for PIA. ARIAD also recovers more card suffixes (38/47 versus 29/47), while PIA recovers more API keys (8/36 versus 1/36) and UUID fragments (17/28 versus 11/28). The field denominators vary and may count multiple fields in a prompt, so they are not interchangeable with prompt EM or Flair NERR.

Table 54: Recovery from 150 generated canary requests.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
CAN	PIA	.865	.033	.704	.511	.722
	ARIAD	.896	.187	.821	.672	.772
	ARIAD w/o LM	.552	.000	.465	.193	.515

Table 55: Exact sensitive-field recovery. Header counts are fields. ARIAD Email is 26/32 (.8125), displayed as .813 at three decimals. Fields in the same request share its reconstruction; field counts are distinct from prompt EM.

Method	Phone $n = 32$	Email $n = 32$	Card $n = 47$	MRN $n = 35$	Amount $n = 28$	API $n = 36$	UUID $n = 28$
PIA	.906	.594	.617	.829	.679	.222	.607
ARIAD	.938	.813	.809	.829	.643	.028	.393
ARIAD w/o LM	.031	.031	.106	.000	.071	.000	.036

Removing the LM proposal branch worsens random-field recovery instead of reversing the gap. The reversal cannot therefore be repaired by simply removing LM proposals. Conditioning, fitting and the remaining candidate coverage jointly determine which random fields can be recovered. The field table consequently distinguishes demonstrated structured-identifier recovery from the weaker random-string results.

Generating new fields reduces dependence on memorizing an entire public evaluation request. The carrier text and field formats still supply useful priors, so this tests request-specific recovery rather than an absence of all prior knowledge.

J.3 Order and longer sequences

Table 56 crosses natural/shuffled order with the LM branch. Shuffling changes the contextual representation and the proposal distribution together. The archived diagnostic shows a ranking reversal, but the unverified warm-up source prevents attributing it to a compliant attack on arbitrary token permutations. It is also not a clean intervention on only one prior.

Table 56: Token-order and LM-proposal controls under the historical EM convention. TAP-Self results under shuffling require validation of the warm-up target and remain diagnostic. Shuffled-text entity recovery is unavailable because its annotation alignment is unspecified.

Setting	Method	Acc.	EM	F1	BLEU-4	NERR
Natural/LM=1	PIA	.937	.220	.951	.916	.830
	ARIAD	.975	.540	.974	.950	.950
Natural/LM=0	PIA	.857	.007	.863	.747	.741
	ARIAD	.889	.013	.878	.758	.847
Shuffled/LM=1	PIA	.944	.140	.935	.865	—
	ARIAD	.936	.113	.927	.842	—
Shuffled/LM=0	PIA	.917	.047	.907	.809	—
	ARIAD	.905	.013	.888	.768	—

Table 57 includes the 256-token benchmark. ARIAD has higher positional accuracy than Boundary (std.), whereas Boundary (std.) has slightly higher F1 and BLEU-4. All methods have zero EM, yet their entity recovery remains substantial. Longer sequences create more opportunities for an omission or ranking failure; that observation does not require independent token errors or identify prefix propagation as the only cause.

Table 57: Residual-calibrated gray-box and 256-token extensions. Adaptive gray-box entries are diagnostic pending permitted-target warm-up validation.

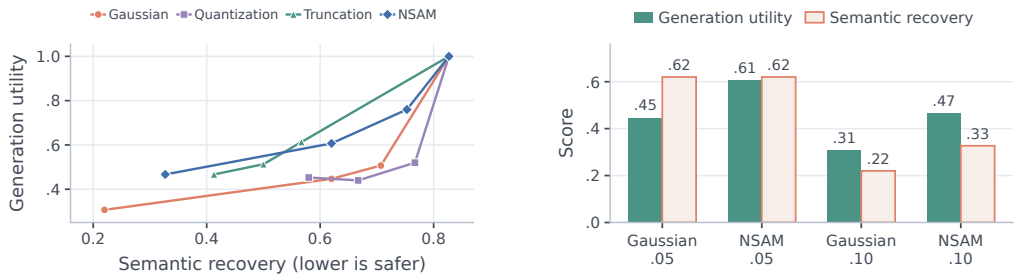
Setting	Method	Acc.	EM	F1	BLEU-4	NERR
gray-LRC	PIA	.882	.020	.897	.816	.743
	ARIAD	.903	.053	.900	.815	.802
long256, n=80	PIA	.874	.000	.887	.792	.796
	Boundary (std.)	.892	.000	.905	.819	.847
	ARIAD	.904	.000	.904	.815	.854

The gray residual-calibrated rows in the same table are privileged-planner diagnostics, separate from the base-only gray summaries. A position-shared low-rank correction is not equivalent to recovering every unknown adapter parameter and does not repair the reference warm-up's omitted victim.

J.4 All defense configurations

Figure 6 evaluates null-space-aligned masking (NSAM) alongside Gaussian noise, quantization, and variance truncation. The method uses the RMSNorm gain g (Zhang & Sennrich, 2019) when constructing its local projection proxy.

In the reference implementation, main fitting and PAD use the defended target trajectory, while the recursive warm-up does not forward the defense configuration. The corrected black-box report does not verify which source generated every historical defense row. Table 58 therefore preserves the complete five-metric Gaussian, quantization, truncation, and NSAM results as historical single-realization diagnostics, rather than validating a fully defended attack interface.



(a) Measured privacy–utility points.

(b) Utility and leakage separately.

Figure 6: Archived defense operating points. Lines connect measured configurations. Attack recovery remains diagnostic because compliant warm-up provenance is unverified. Semantic outcomes use the fixed local judge; the independent noise-seed study measures token, prompt, and entity recovery.

Table 58: Activation-defense results under the historical EM convention, using one noise realization. Attack rows remain diagnostic pending validation of the defended warm-up target.

Defense	Strength	Acc.	EM	F1	BLEU-4	NERR
None	–	.975	.540	.974	.950	.950
Gaussian	0.03	.966	.367	.963	.925	.912
Gaussian	0.05	.943	.127	.942	.883	.881
Gaussian	0.10	.844	.013	.846	.704	.706
Quantization	64	.971	.527	.972	.944	.959
Quantization	16	.956	.207	.957	.915	.881
Quantization	8	.932	.100	.927	.855	.849
Truncation	0.90	.913	.093	.911	.828	.858
Truncation	0.70	.895	.033	.883	.772	.857
Truncation	0.50	.854	.000	.840	.696	.778
NSAM	0.02	.972	.500	.971	.943	.936
NSAM	0.05	.954	.280	.952	.907	.899
NSAM	0.10	.877	.033	.871	.745	.744

For NSAM, let V_{tail} contain orthonormal right-singular vectors of $M = W_{qkv}\,\text{diag}(g)$ with singular values at most τ. Every sampled perturbation δ in their span satisfies $\|M\delta\| \le \tau\|\delta\|$. Sampling $\delta = \alpha\,\text{rms}(H)V_{\text{tail}}z$ with standard Gaussian z therefore limits one local projection of the perturbation. The bound does not include the input-dependent normalization denominator or subsequent nonlinear blocks. It is a design criterion, not a differential-privacy or inversion-resistance guarantee.

J.5 Noise replication and semantic judging

Tables 59 and 60 add five-noise-seed recovery replication with initialization fixed. Gaussian .10 drives EM near zero, but mean NERR remains .701; NSAM .10 gives .031 EM and .727 NERR. A near-zero exact-prompt count therefore does not imply that the sensitive entity is protected.

Table 59: Five-noise-realization defense replication. Entries are means ± sample SD over noise-seed means with attack initialization fixed. Attack rows remain diagnostic pending validation of the defended warm-up target. These replications measure attack recovery; semantic and utility judgments are reported separately.

Defense / scale	Acc.	EM	F1	BLEU-4	NERR
Gaussian:0.03	.968±.002	.397±.047	.966±.003	.935±.007	.917±.008
Gaussian:0.05	.946±.006	.179±.045	.946±.006	.893±.013	.871±.013
Gaussian:0.1	.845±.004	.019±.009	.846±.003	.706±.007	.701±.013
NSAM:0.02	.970±.004	.480±.024	.969±.004	.939±.008	.932±.013
NSAM:0.05	.962±.006	.324±.044	.960±.006	.921±.011	.893±.011
NSAM:0.1	.878±.003	.031±.010	.875±.003	.754±.006	.727±.016

Table 60: Repeated noise realizations, with attack initialization fixed. Each row contains 150 target requests. The five metrics retain their original meaning. Attack rows remain diagnostic pending validation of the defended warm-up target.

Defense / scale	Noise seed	Acc.	EM	F1	BLEU-4	NERR
Gaussian:0.03	0	.966	.367	.963	.925	.912
	1	.966	.393	.964	.931	.906
	2	.967	.340	.967	.935	.925
	3	.971	.453	.970	.944	.918
	4	.968	.433	.969	.938	.925
Gaussian:0.05	0	.943	.127	.942	.883	.881
	1	.949	.207	.947	.894	.866
	2	.940	.167	.940	.880	.858
	3	.954	.240	.956	.913	.889
	4	.942	.153	.945	.892	.862
Gaussian:0.1	0	.844	.013	.846	.704	.706
	1	.843	.013	.845	.709	.705
	2	.850	.020	.851	.714	.692
	3	.847	.013	.843	.695	.684
	4	.840	.033	.846	.705	.717
NSAM:0.02	0	.972	.500	.971	.943	.936
	1	.967	.480	.966	.935	.926
	2	.967	.460	.965	.930	.951
	3	.976	.507	.975	.952	.917
	4	.969	.453	.967	.934	.930
NSAM:0.05	0	.954	.280	.952	.907	.899
	1	.965	.380	.963	.927	.897
	2	.969	.360	.968	.935	.903
	3	.959	.287	.957	.914	.890
	4	.960	.313	.958	.920	.876
NSAM:0.1	0	.877	.033	.871	.745	.744
	1	.876	.013	.873	.752	.710
	2	.875	.033	.874	.756	.723
	3	.879	.040	.877	.759	.715
	4	.882	.033	.878	.758	.745

Table 61 retains every historical utility operating point including all configurations shown in Figure 6. The semantic-recovery and generation-utility assessments use a fixed local `gpt-oss-120b-yd`

Table 61: Semantic-recovery and generation-utility point estimates from a fixed local judge. Attack recovery remains diagnostic pending validation of the defended warm-up target. One noise realization per setting, without repeated-noise or human-adjudicated intervals.

Defense	Setting	Semantic recovery ↓	Generation utility ↑
None	–	.827	1.000
Gaussian	.03	.707	.507
Gaussian	.05	.620	.447
Gaussian	.10	.220	.307
Quantization	64	.767	.520
Quantization	16	.667	.440
Quantization	8	.580	.453
Truncation	.90	.567	.613
Truncation	.70	.500	.513
Truncation	.50	.413	.467
NSAM	.02	.753	.760
NSAM	.05	.620	.607
NSAM	.10	.327	.467

judge. The original operating points and noise-seed-zero results use 150 requests; the additional noise realizations use 20 requests each. Tables 62 and 63 report every cohort separately. The smaller

cohorts estimate repeated-noise behavior but are not pooled with the 150-request measurements as five equal-sized repetitions.

Table 62: Semantic recovery, generation utility, and perturbation cost: Gaussian. Noise seed 0 uses 150 requests; seeds 1–4 use 20 each. Attack recovery remains diagnostic pending validation of the defended warm-up target. Lower semantic recovery and higher generation utility are better. Unequal cohorts are reported separately.

Scale	Noise seed	Judged n	Semantic recovery	Generation utility	Mask (ms)
0.03	0	150	.7067	.5067	4.9433
0.03	1	20	.7000	.5000	4.6774
0.03	2	20	.8000	.7000	4.6669
0.03	3	20	.8000	.5500	6.0817
0.03	4	20	.6000	.7500	6.6066
0.05	0	150	.6200	.4467	6.1823
0.05	1	20	.7500	.5000	5.8870
0.05	2	20	.6500	.6000	5.8777
0.05	3	20	.8500	.6000	5.8837
0.05	4	20	.7000	.6000	5.8574
0.1	0	150	.2200	.3067	5.9693
0.1	1	20	.1000	.3500	5.8486
0.1	2	20	.3000	.2500	5.8862
0.1	3	20	.1000	.6000	5.8635
0.1	4	20	.3000	.3000	5.9660

Table 63: Semantic recovery, generation utility, and perturbation cost: NSAM. Noise seed 0 uses 150 requests; seeds 1–4 use 20 each. Attack recovery remains diagnostic pending validation of the defended warm-up target. Lower semantic recovery and higher generation utility are better. Unequal cohorts are reported separately.

Scale	Noise seed	Judged n	Semantic recovery	Generation utility	Mask (ms)
0.02	0	150	.7533	.7600	2.0941
0.02	1	20	.8500	1.0000	2.1483
0.02	2	20	.8000	.9000	2.1616
0.02	3	20	.8500	.9500	2.1173
0.02	4	20	.8500	1.0000	2.1302
0.05	0	150	.6200	.6067	1.9349
0.05	1	20	.7500	.9000	2.0018
0.05	2	20	.8500	.9000	2.0086
0.05	3	20	.8000	.9500	2.0139
0.05	4	20	.7000	1.0000	2.0084
0.1	0	150	.3267	.4667	2.0003
0.1	1	20	.4500	.6500	2.0185
0.1	2	20	.3500	.7500	2.0000
0.1	3	20	.4000	.9500	2.0037
0.1	4	20	.5000	.9500	2.0236

Request-time perturbation costs average 5.40–5.94 ms for Gaussian noise and 1.99–2.13 ms for NSAM across the three reported scales. These times cover noise construction and application, not the offline singular-subspace calculation or full generation. They measure a local cost of the defense rather than an end-to-end serving speed advantage.

Table 64 reports two-annotator assessments of 40 items per task. The annotators agree on 38 semantic-recovery items and 35 utility items, with $\kappa = .8958$ and $.7143$, respectively. The semantic judge agrees with the reported consensus on 26 items; the corresponding utility count is unavailable. Human agreement and agreement with the automatic judge are therefore kept separate. These small assessments provide a check on label consistency, while the five-noise-seed token, sequence and entity measurements assess recovery on the full request cohort.

The archived diagnostics illustrate why exact match and entity recovery must be reported separately. NSAM targets low-gain directions of the next projection; its selected operating points do not establish

Table 64: Human agreement on binary semantic-recovery and utility judgments. Each task has 40 labeled items and two annotators. Annotators agree on 38 semantic and 35 utility items. Utility judge agreement is unavailable.

Task	Labeled n	Both yes	Both no	A yes/B no	A no/B yes	Judge agrees	κ
semantic	40	23	15	1	1	26	.8958
utility	40	25	10	0	5	—	.7143

protection against a compliant end-to-end attack. Protection of the full request requires evaluating both the representation and its execution interface against the attacks they permit.