WHAT RANK CALIBRATION ACTUALLY MEASURES: THE SPARSE, REPEATABLE, QUERY-SPECIFIC RESPONSE OF VIDEO LLMS TO KV-CACHE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank compression of a video language model's key cache is usually tuned by reading a rank budget off the cache's spectrum and reporting the change in average accuracy. We treat compression as a controlled intervention instead and ask *where* it acts. Holding the input and the query fixed and moving only the rank, the set D of inputs whose answer changes has three properties, measured on MVBench with Qwen2-VL-7B under rules frozen before the data. It is **sparse**: a task's accuracy-versus-rank knee is carried by 5–14% of its inputs, which is why a per-task allocator gains little over the best uniform rank ($+1.3$pp, CI $[-0.3, +3.0]$). Its allocation-relevant part is **unreadable** from the signals we test: whether a cheap rank breaks an input is predictable (AUC 0.755), but whether more rank fixes it is at chance (AUC 0.47, $[0.38, 0.57]$). And it **repeats**: across option order, frame sampling, a second rank pair, real video, a second compressor and a second architecture, an input that changed its answer once changes it again with probability 56% against a base rate of 12% (n=2,392, κ=0.50). That repeatability belongs to the video–question pair: profiling a video on some questions does not inform its others ($+0.19$pp against a video-level oracle of $+8.3$). Together these identify what a rank calibration has to measure—the pair, not the task, the spectrum or the video—and imply a measure-then-reuse pattern that labels only the inputs where two ranks disagree (25–49% of the pool, an exact bound) and returns $+2.4$pp (CI $[+1.25, +3.72]$) on held-out realizations at a matched key-cache budget, in workloads whose outputs must be recomputed rather than looked up.

1 INTRODUCTION

Compressing a video language model's key–value (KV) cache to low rank requires choosing how much rank to keep. The dominant answer reads the budget off the cache's own singular spectrum—per layer, per task, or per head (Saxena et al., 2024; Chen et al., 2026b; Bhatnagar et al., 2026)—and the adaptive-budget literature has moved that choice from prediction to calibration on data (Zhou et al., 2025; Yu & Chai, 2025; Tu et al., 2024). This paper asks a prior question: *what is the quantity being calibrated?* We argue, with evidence frozen before it was scored, that it is the set

$$D = \{x : \text{ compression changes the model's behavior on } x\},$$

operationally the set on which two candidate ranks disagree, and that D has three properties that together dictate what a rank calibration can and cannot do. It is **sparse**: a task's accuracy-vs-rank knee, the object per-task allocators target, is carried by a small minority of inputs. Its **allocation-relevant part is not recovered by the static signals we test**: whether a cheap rank breaks an input is partly predictable, but whether more rank fixes it is at chance from the spectrum of the cache and from every deployment-time signal we test. And it is **repeatable**: the inputs in D are substantially the same inputs across nuisance realizations, rank pairs, compressors and architectures. Sparse, and unreadable where it matters, means the budget must be *measured*; repeatable means the measurement can be *reused*.

The argument runs in six steps, each backed by a committed run file. (1) The representation is load-bearing: at a nominal $\sim 19\times$ key-cache ratio, mode-wise Tucker outperforms flat SVD by $+10.9$pp

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.

across six MVBench tasks (§3). (2) Task-level rank allocation has little leverage on top of it: a per-task allocator calibrated on accuracy ties the best cap-matched uniform rank, and in the tested matched-budget setting a hierarchical test finds the task axis adds $+0.00$pp once per-input response is known (§4). (3) The reason is sparsity: the knee that the task allocator measures is an aggregate of 5–14% of inputs (§4). (4) Static structure locates neither the task-level depth nor the inputs more rank will fix: nine pre-registered structural, predictive and allocation accounts fail at the task level, and the sign of an input's rank-benefit at the operating pair is at chance from leakage-free signals (§5). (5) So one must measure—and because two ranks that agree on an input contribute exactly zero to their accuracy difference, only the disagreement union needs labels (§6). (6) The measurement repeats and therefore amortizes (§7), which is what a deployment can use (§8).

Each half of this story has been told before as a weakness: an allocator that barely beats a uniform rank looks like a method that does not work, and a per-input policy that cannot predict its own targets looks like one that cannot be deployed. Read together they are the same fact from two sides, and both say **aggregate accuracy hides the structure compression actually has**—by steps (2) and (3) the allocator is not weak, it is allocating at the wrong granularity, which needs a different measurement rather than a different allocator.

Our contributions are three. **An intervention framework.** We localize *where* compression changes behavior rather than summarizing it through average accuracy. The object is the behavioral support $\mathcal{S}(r_{\mathrm{lo}}, r_{\mathrm{hi}}) = \{i : \hat{y}_i(r_{\mathrm{lo}}) \neq \hat{y}_i(r_{\mathrm{hi}})\}$ of a controlled capacity intervention. **A stability protocol.** It tests whether such a support set survives a change of nuisance realization, rank pair, compressor, architecture or benchmark, and its rules were fixed before scoring. **The finding itself.** The response is sparse, unreadable where allocation needs it, and repeatable, which is what makes measure-then-amortize possible. It also bounds the labels: two ranks that agree on an input contribute exactly zero to their paired accuracy difference, McNemar's identity, so only the disagreement union needs labels, 25–49% of the pool here.

2 Related work

KV-cache compression. Eviction keeps a token subset by attention statistics (Zhang et al., 2023; Li et al., 2024c; Cai et al., 2024); quantization lowers precision (Liu et al., 2024; Hooper et al., 2024); low-rank methods keep a rank-r subspace of the projections or activations (Chang et al., 2024; Saxena et al., 2024; Xu et al., 2025; Lin et al., 2025). Our compression is a training-free per-sample HOSVD of the key tensor (Tucker, 1966; Kolda & Bader, 2009); the paper is about how much rank to keep, not about the factorization.

Adaptive budgets. DynamicKV (Zhou et al., 2025) adapts per-layer budgets to the task; EvolKV (Yu & Chai, 2025) searches layer budgets on downstream accuracy; STAR-KV (Bhatnagar et al., 2026) learns static rank thresholds; VL-Cache (Tu et al., 2024) tunes a sparsity-aware per-layer budget; GVote (Tang et al., 2025) sets a per-request eviction budget; DepthWeave-KV (Cordoba et al., 2026) routes rank per token. All determine the budget by measuring or searching on data rather than predicting it, and none amortizes a measured per-input response across repeated occurrences of the same input. The weight-space SVD literature has reached the same diagnosis from the other side: SVD-LLM whitens activations because singular values do not map to loss (Wang et al., 2025); IO-SVD, LACE-SVD and StiefAttention replace reconstruction error by loss-aligned objectives (Abbasi et al., 2026; Liu et al., 2026; Benfenati et al., 2026); Shin et al. (2024) find reconstruction-error minimization overfits calibration data. Each keeps a surrogate in place of the downstream metric; we calibrate on the downstream metric and show what that costs and what it buys. KV-CoRE (Chen et al., 2026b) correlates a spectral rank with perplexity; we show such diagnostics do not locate the task-accuracy response.

Heterogeneity of compression damage. That compression damage concentrates on a minority of inputs is an established finding outside this setting. Hooker et al. (2019) show that pruning a vision network leaves aggregate accuracy nearly unchanged while a small, *consistently identifiable* set of exemplars is disproportionately affected, and Dutta et al. (2024) show that compressed LLMs can match a baseline's accuracy while flipping many individual answers, and propose the flip rate as the metric aggregate accuracy hides. Our sparse, repeatable D is the same phenomenon measured for the rank of a video LLM's key cache; what we add is that the part of D an allocator needs—whether

Table 1: The substrate: mode-wise Tucker $(8, 16, 16)$ vs. flat SVD r=6 on the visual key cache of Qwen2-VL-7B, six MVBench tasks, N=200 per task (198 for object_existence). CR = per-sample key compression ratio logged by the compressor (mean over inputs); a row is budget-matched only if Tucker's CR is $\geq$ SVD's on every sample.

task	dense	Tucker	SVD	gap	exact p	CR T / S
moving_attribute	93.0	89.0	71.5	+17.5	$< 10^{-4}$	18.9 / 18.7
object_existence	92.9	82.3	66.7	+15.7	$< 10^{-4}$	18.9 / 18.7
counterfactual_inference	58.5	55.5	44.0	+11.5	0.002	18.9 / 18.7
moving_count	64.5	63.0	53.5	+9.5	0.004	18.9 / 18.7
moving_direction	50.5	43.5	34.5	+9.0	0.010	18.9 / 18.7
action_antonym (not budget-matched)	81.0	78.5	76.5	+2.0	0.39	11.8 / 16.6
mean, six tasks				+10.9pp		
mean, five budget-matched tasks				+12.6pp		

more rank *fixes* an input—is unreadable where membership is partly readable, that the set repeats across compressors and architectures rather than within one model, and that it does not transfer across questions on the same video. Two recent lines agree that the failure is not uniform over inputs: Chen et al. (2026a) find that under five KV-cache compression methods "certain instructions degrade much more rapidly with compression, effectively causing them to be completely ignored", and Haverbeck et al. (2026) characterize the minimax risk of compression in terms of the intrinsic compressibility of a cache rather than a shared budget. Both motivate treating the response as an input-level object; neither measures whether the responding inputs repeat, which is the question this paper answers. Our scope is the tensorized visual key cache of video LLMs; Salfati (2026) report that on text LLMs (124M–14B) quantization dominates rank reduction, and we make no claim against quantization or beyond the video setting we test.

3 Setup and the substrate

We compress the visual key tensor of each attention head, reshaped from $[T_{\text{vis}}, C]$ into $\mathcal{X} \in \mathbb{R}^{T \times S \times C}$ over frames, spatial patches and channels, with a Tucker decomposition of per-mode ranks (r_t, r_s, r_d); flat SVD treats the same activations as one $[T_{\text{vis}}, C]$ matrix of rank r. Values are left untouched, so every ratio below is a *key-cache* ratio and end-to-end KV reduction is bounded by $2\times$. Stored parameters are $r_t r_s r_d + T r_t + S r_s + C r_d$ (Tucker) and $r(T_{\text{vis}} + C)$ (SVD), chosen equal to within a few percent at the nominal grid. Because T and S vary with the clip, we define effective storage per sample as the retained count the compressor logs, and call a comparison budget-matched only where that count is equal or lower for Tucker on every sample.

We compare the two on Qwen2-VL-7B (Wang et al., 2024) at a nominal $\sim$19$\times$ ratio: Tucker $(8, 16, 16)$, effectively $(5, 16, 16)$ at five frames and 6,345 of 119,680 parameters per head, against SVD r=6 at 6,378. Tucker wins on all six MVBench (Li et al., 2024b) tasks, N=200 each (198 for object_existence): moving_attribute +17.5pp (exact McNemar $p < 10^{-4}$), object_existence +15.7 ($p < 10^{-4}$), counterfactual_inference +11.5 (p=0.002), moving_count +9.5 (p=0.004), moving_direction +9.0 (p=0.010), action_antonym +2.0 (p=0.39)—all six positive, mean +10.9pp. This substrate is five frames with the temporal mode uncompressed, while every later result is at 32 temporal positions and no flat-SVD arm was run there, so this result and the rest of the paper sit at different context lengths. Two checks and a per-sample budget rule make this a compression effect rather than an accounting one (Appendix A); on action_antonym the budgets are not matched, so its +2.0pp is descriptive only (Table 1). The substrate is thus verified; what the rest of the paper shows is that *how much* of it to keep is a property of inputs, not of tasks or spectra.

4 Task-level knees are aggregates of a sparse input-level response

A per-task allocator calibrated on accuracy ties the simplest baseline. Fix r_t, r_s and sweep the feature rank $r_d \in \{8, 11, 13, 15, 30\}$ at T=32. Each task's accuracy-versus-rank curve has its own

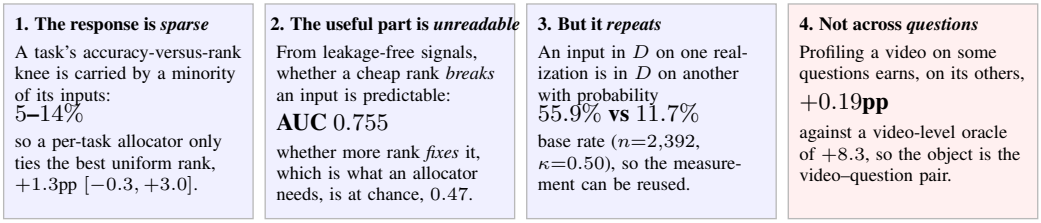

Figure 1: The argument in four measurements. **(1)** and **(2)** say the rank budget cannot be read off and has to be measured; **(3)** says the measurement can be reused, which is what makes measuring it worth the forwards; **(4)** says the unit measured must be the video–question pair, not the video. Sections 4, 5 and 7 give each number with its interval.

depth: object_existence is flat from 15 to 30 (75.3% at both, $b{=}c{=}4$ over 150 inputs), while moving_attribute gains +7.5pp over the same step ($p{=}7 \times 10^{-4}$) and counterfactual_inference +3.3pp, both still improving at 30. Give each task the depth this sweep measures, and fill the rest of the budget by core-maximization. That allocator beats the spectral heuristics the field uses: +4.5pp (CI $[+2.8, +6.2]$) over pure core-maximization and +6.0pp ($[+3.6, +8.4]$) over floor-matching, equal-task-weighted over three Qwen tasks at 22× and 35×. Against a *uniform* cube chosen on a held-out split under the same cap, however, it only ties: +1.3pp, CI $[-0.3, +3.0]$. The uniform cube is also cheaper, storing 7–16% fewer parameters in every cell. Once the representation is right, the task-level lever is small.

The task axis adds nothing once per-input response is known. A hierarchical test makes this exact: profiling and evaluating on different realizations (§7) at the same tasks and rank pair $\{15, 30\}$, held-out policy value is +2.68pp per-input against +0.42pp task-level and +0.25pp task-static, and combining the two adds nothing to per-input alone—+0.00pp, lower bound +0.00, on all three tasks. The zero is structural at this budget, not an accident of estimation. A task prior of weight λ can reorder inputs only if $\lambda/(1-\lambda)$ falls below the largest task-mean gap, 0.065; at the frozen $\lambda{=}0.357$ it is 0.555, so the joint policy selects exactly the inputs the per-input policy selects. A different budget or weight could give the task prior room; we claim only that at this operating point it adds nothing.

Why: the knee is carried by a minority. The reason is visible in the paired structure of the sweep (Figure 2). Each step of the ladder changes the prediction on 5–14% of a task's inputs, the steps that set the three tasks' depths included (Figure 2 gives the counts). (D below uses the same definition; correctness flips are a subset of prediction changes.) Over the whole ladder, the union of inputs on which any adjacent pair of ranks disagrees is 25%, 33% and 49% of the three pools. $|D|$ cannot fall below the net accuracy gap $|b - c|$ between the two ranks, and it does not sit there: over the twelve cells of §7 it is about three times that floor on average, though within a factor of 1.3 of it in two (Appendix B). A task's accuracy-vs-rank curve is therefore the aggregate of a response concentrated on a small subset—the set D—and a per-task rank is one number standing in for a per-input pattern.

We formalize the input-level view with the signed rank-benefit $\tau_{iv} = \mathbf{1}[\text{correct at } r_{\text{hi}}] - \mathbf{1}[\text{correct at } r_{\text{lo}}] \in \{-1, 0, +1\}$ of input i under realization v, decomposed as

$$\tau_{iv} = \mu_{t(i)} + u_i + \varepsilon_{iv},$$

a task mean (what a task-level policy uses), a repeatable per-input effect (what a profiling policy can capture) and a view residual (what no realization-blind policy reaches). A clairvoyant oracle that sets each input's rank from its observed correctness exploits u and ε together, and beats the best task-static policy by 13.4pp at matched budget on four tasks. Most of that is not there to be had. A random-intercept null with $u \equiv 0$, in which every rank-to-rank change is a fresh draw of ε, already reproduces 9.4pp of it (95% $[8.0, 10.8]$ over 400 simulated ladders). About 70% of the apparent headroom is therefore selection on noise, and the exploitable target is a u of about 4pp. Whether it can be found is the next question.

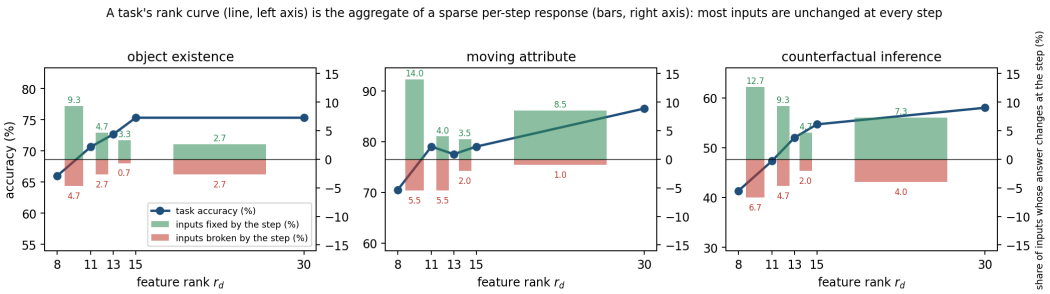

Figure 2: A task's accuracy-vs-rank curve (line) is the aggregate of a sparse per-step response (bars: share of inputs whose answer is fixed or broken by each rank step). At every step most inputs are unchanged. The steps that set the three tasks' depths move 19 of 200 inputs (moving_attribute; 17 rescued, 2 regressed), 8 of 150 (object_existence, b=c=4) and 21 of 150 (counterfactual_inference). Data: the rank ladders behind §4, N=150/200/150.

Table 2: Nine structural, predictive and allocation accounts, each frozen before it was tested, each tested as a candidate mechanism and none supported: FALS = falsified by a counterexample, ART = a measurement artifact, N.E. = not established. Rows 1–6 concern the Tucker-vs-SVD win; rows 7–9 concern unsupervised allocators.

hypothesis	decisive control	verdict
temporal mode is most compressible (rank ≈1)	per-mode SVD, keys-only, per layer	ART: layer-0 value spliced with K+V pooling
temporal is *least* compressible (inversion)	fractional rank at T=5	ART: T=5 small-dimension
temporal saturates ⇒ savings ∝ length	diversity ladder (scenes, fixed frames)	FALS in the tested ladder
higher temporal rank aids accuracy	core-matched pair $(4, 28, 3)$ vs $(1, 21, 16)$	N.E.: one pair, confounded (feature mode starved); p=0.029 *against*
core volume orders accuracy universally	T=32 dose-response	FALS at T=32
floor normalization adds predictive value	paired bootstrap vs raw dimension	N.E.: p=0.12, variant-dependent; ρ=0.78 (floor law) vs 0.61–0.67 (raw dimension)
floor coverage translates into an allocator	matched-CR vs core-maximization	FALS: too conservative
unsupervised-floor allocator benefits across tasks	eight cells; the frozen pooled test invalid once cells were added, post-hoc re-pool p=0.089	N.E.: 2 of 4 added cells favor baseline
violation depth orders the scope	Spearman(violation, effect)	N.E.: ρ=0.06, wide CI

5 THE TESTED STATIC SIGNALS DO NOT LOCATE THE ALLOCATION-RELEVANT PART OF D

Structural accounts of the substrate win. Before each of nine explanatory experiments we froze the arms, the parameter budgets and a decision rule in version control, then looked at numbers. None of the nine is supported (Table 2).

The nine share a shape: each offers a structural quantity—a mode's rank, a core volume, a spectral floor—as a stand-in for behavioral depth, and each is either refuted by a control that moves the structure while holding the behavior, or turns out to be an artifact of how the spectrum was pooled or of a small dimension. The temporal-redundancy family fails on its own measurement three times over: pooling keys with values across layers puts the temporal rank near 4 where per-layer keys-only gives 1–4 and the per-dimension ordering inverts; a scene-diversity ladder shows temporal rank

tracking scene count, not frame count; and a core-matched pair with *higher* temporal rank scores 54.0% against 60.6%, p=0.029 in the opposite direction. Core volume breaks at T=32, where a $3.8\times$ range of it scores identically once one mode is starved. Four further rulers on the key stream—attention- and gradient-weighted floors, a causal block knockout, an activation-patching frontier—return values differing by at most one unit on the two tasks whose depths differ, and two joint-subspace tests point the wrong way (supplement). A construction explains why no spectral criterion can succeed in general: a task whose answer is read from a single coordinate k has behavioral depth k whatever the spectrum is, so depth is not a function of the spectrum absent an assumption about readout alignment.

Per-input signals predict fragility, not recoverability. From one dense forward, using only signals computable without the answer (top-2 option-logit margin and key-spectrum tail statistics), a logistic probe with video-grouped cross-validation predicts *fragility*—correct at full rank, wrong at r_{lo}—at held-out AUC 0.755 ($[0.682, 0.825]$), and transfers to real video (STAR) at 0.722 ($[0.635, 0.807]$). The allocation-relevant quantity is the other one, whether more rank *fixes* an input, and at the operating pair $\{15, 30\}$ it is at chance from the same features: AUC 0.47 ($[0.38, 0.57]$, permutation p=0.68), on 37 positives in 650 inputs. That design could not have separated a true AUC below $\approx$0.60 from chance, so we claim only the operative negative: a per-input policy built from these signals gains $+0.2$pp over the best task-static policy at the same mean rank, 1% of the oracle headroom, and that from an in-sample fit. The negative is also pair-specific—a wider label that mixes in fragility reaches 0.61–0.65—while a deployment-legal finite-difference feature over ranks already paid for is at chance too (AUC 0.50, n=110; supplement). Every predictive number here is leakage-free. The conclusion is the methodological pivot of the paper: the sign of the response is an intervention property, found by intervening. All three facts replicate on Qwen2.5-VL-7B under rules frozen in advance. The $15\to30$ step is carried by 6.5–10.7% of the inputs on all four tasks, and the union any adjacent step changes is 32–54%. Fragility is again predictable (AUC 0.714, $[0.625, 0.800]$) and recoverability again is not (0.470, $[0.382, 0.558]$, p=0.68). Knees land at r_d=30 on two tasks and 15 on two.

6 Measure: the labels a rank calibration actually needs

Calibration reads behavior instead of structure, which is what makes it work here, but it spends labels, and the number it needs is set by the gap it must resolve: halving the accuracy difference between two candidate ranks requires on the order of four times as many labeled examples. At MVBench's $\approx$200 examples per task this resolves the $+7.5$pp step that places moving_attribute's depth above 15 and does not separate neighbouring ranks differing by a point or two. A cheaper consequence follows from the same paired structure. Each step of a sweep is a paired comparison, so an example on which two ranks predict identically contributes exactly zero to their accuracy difference whatever its label—the discordant-pair structure of McNemar's test (McNemar, 1947), restated for rank selection. Ground-truth labels are therefore needed only on the union of examples where some ladder step changes the prediction, a set identifiable from unlabeled inference. On our three Qwen tasks that union is 25%, 33% and 49% of the pool (2.0–$4.0\times$ fewer labels). It is an upper bound on what needs labelling—two ranks can disagree while both are wrong (0–33% of the discordant examples here)—so the saving is a lower bound. This is an exact identity, not a learned selector or a heuristic: the set that must be labelled is D itself, computed by running the candidate ranks, which is the measurement the policy performs anyway.

7 The measurement repeats, and therefore amortizes

D is repeatable across realizations, and the repeatability recurs across rank pairs, compressors and architectures. If the response were realization noise, measuring it once would be worthless. We froze the definition before testing it: $D_v(x) = \mathbf{1}[\hat{y}^v_{r_{\text{lo}}}(x) \neq \hat{y}^v_{r_{\text{hi}}}(x)]$, from predicted letters only. The test runs on cached views spanning two nuisance axes (option order re-shuffled, frames re-sampled), a second rank pair, STAR real video, a fixed-basis key-and-value Palu-style projection and LLaVA-OneVision (Li et al., 2024a); Table 3 names the cells. The Qwen option-order cell was declared the pilot; the other five groups were confirmatory, with a two-part rule fixed before scoring (pooled lift LCB > 0 and enrichment ≥ 2; at least three of five groups with LCB > 0). It repeats,

Table 3: Repeatability of the disagreement set D (pre-registered; rules frozen before scoring). For two realizations of the same inputs: base rate $P(D_2)$, hit rate $P(D_2 \mid D_1)$, miss rate $P(D_2 \mid \neg D_1)$, lift = hit − miss with its bootstrap lower bound, Cohen's κ. Confirmatory groups only; the option-order pilot is excluded. obj/ma/cf = object_existence / moving_attribute / counterfactual_inference; $A/B/C$ = original option order / shuffled / re-sampled frames. The pooled CI is an input-level percentile bootstrap over the 2,392 pairs; a question-clustered bootstrap over the 798 distinct questions they rest on (the three STAR cells re-use the same 200 inputs and MVBench questions recur across cells) gives lift $[+0.44, +0.56]$ and κ $[0.44, 0.56]$, indistinguishable from it.

group	n	base	hit	miss	lift [LB]	κ
frames re-sampled (A/C; Qwen $\{15, 30\}$; obj, ma)	398	7.5%	66.7%	2.7%	+0.64 [+0.46]	0.64
second rank pair (Qwen $\{11, 30\}$; obj, cf)	398	18.6%	53.3%	8.4%	+0.45 [+0.34]	0.48
real video (STAR; three view pairs, see text)	600	11.7%	51.5%	6.7%	+0.45 [+0.32]	0.44
second compressor (Palu-style $\{20, 24\}$; obj, ma, cf)	598	7.7%	46.7%	5.6%	+0.41 [+0.23]	0.33
second architecture (LLaVA-OV $\{15, 30\}$; obj, ma)	398	14.8%	63.5%	5.7%	+0.58 [+0.45]	0.59
pooled confirmatory	2392	11.7%	**55.9%**	5.8%	**+0.50** [+0.44]	0.50

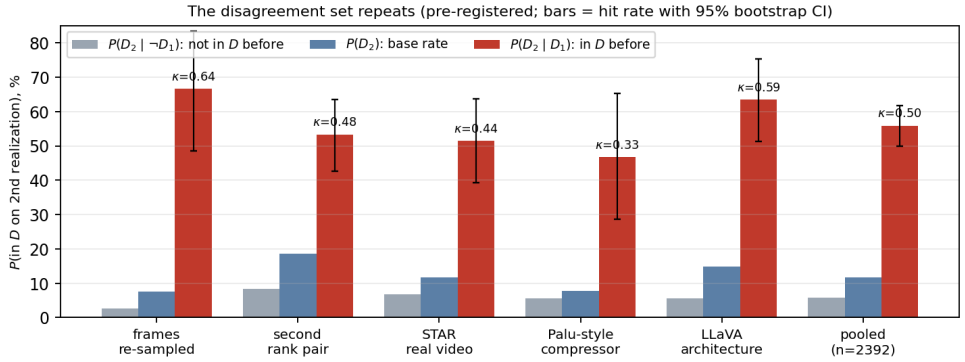

Figure 3: Repeatability of D by confirmatory group (Table 3): the probability that an input is in D on a second realization given that it was ($P(D_2 \mid D_1)$, with 95% bootstrap CI) or was not ($P(D_2 \mid \neg D_1)$) in D on the first, against the base rate. κ above each hit bar.

in every group (Table 3, Figure 3). Pooled over 2,392 input pairs, an input in D on one realization is in D on the other with probability 55.9% against a base rate of 11.7%: lift +0.50 (95% CI $[+0.44, +0.56]$, unchanged under a question-clustered bootstrap), κ=0.50, and all five groups clear the bar. The weakest cells are counterfactual_inference under every compressor (κ 0.19–0.33) and STAR under option order (0.24). The most economical alternative explanation is that D is simply the low-margin inputs and that low margin is itself stable: it does not hold. Splitting the pairs by the dense model's top-2 margin, the lift survives in every quartile and is largest, +0.77 with κ=0.80, where the margin is highest (Appendix B). Two limits belong here. Membership in D is partly readable from that margin (AUC 0.613); it is the *sign* that is not (§5). And D is less stable than the model's own correctness, which repeats at κ=0.735 under the same two axes against D's 0.50. Under the secondary definition of D, disagreement with the dense answer, the pooled figures are stronger (base 22.5%, hit 72.2%, κ=0.66).

The signed benefit repeats too, and has held-out value. Repeatability of D says which inputs respond; a policy also needs the sign. The rescued set—inputs with $\tau > 0$—reproduces under both nuisance axes on every task where it is powered, with test–retest κ per task in Table 4. To make that a number a deployment could use, a profiling policy—upgrade input i to r_{hi} iff its profiled $\tau_i > 0$—is scored as a genuine cross-view quantity, profiling on one realization and evaluating on the other, against a budget-matched baseline that upgrades the same fraction without per-input information: $H_{\text{xview}} = \frac{|S|}{N}(\overline{\tau^{\text{eval}}}_S - \overline{\tau^{\text{eval}}})$. Pooled over three tasks it is $+\mathbf{2.43}$pp (95% CI $[+1.25, +3.72]$), clearing the pre-registered bar of +1.5 with a positive lower bound, and about 39% of the in-view clairvoyant ceiling. Per-task values, including the higher-rescue pair $\{11, 30\}$ at which three of the

Table 4: Held-out policy value of profiling, per task: H_{xview} (profile one realization, evaluate another; LB = bootstrap lower bound) at the rank pair where the task is powered, and test–retest κ of the *rescued* set ($\tau > 0$; a different quantity from the disagreement-set κ of Table 3) under two nuisance axes. The κ columns are at $\{15, 30\}$ for all tasks except where a pair is given; at $\{11, 30\}$ the option-shuffle κ is 0.70 (object_existence) and 0.36 (counterfactual). N: 200 (moving_attribute, counterfactual, STAR), 198 (object_existence). Pooled $\{15, 30\}$: +2.4pp, 95% CI $[+1.25, +3.72]$; the two tasks underpowered at that pair still give $+1.41$ ($[+0.23, +2.76]$).

task	H_{xview} (LB), pair	κ option	κ frames
moving_attribute	+4.3 (+1.9), $\{15, 30\}$	0.51 (excess CI excl. 0)	0.69 (+0.49)
object_existence	+5.6 (+2.9), $\{11, 30\}$	0.62 (excess CI incl. 0; 3% rescue)	0.56 (+0.28)
counterfactual_inference	+6.1 (+3.2), $\{11, 30\}$	0.15 (excess CI incl. 0)	—
STAR (real video)	+5.0 (+2.3), $\{11, 30\}$	0.60 at $\{11, 30\}$	0.43 (+0.16)

Table 5: Cross-question transfer on CLEVRER (300 videos; Z_v=1 if any profiling question is in D): held-out $P(D)$, $P(D \mid Z{=}1)$, $P(D \mid Z{=}0)$, lift with 95% video-bootstrap CI; H_{xq} = held-out gain of upgrading a video iff its profiled mean $\tau > 0$, against a budget-matched random upgrade; oracle = upgrade by held-out benefit, best over budgets.

pair	base	hit	miss	lift [CI]	H_{xq} [CI]	oracle
$\{4, 30\}$ (frozen primary)	48.1%	49.2%	40.9%	+8.3 $[+2.0, +14.2]$	+0.19 $[-0.97, +1.34]$	+8.3
$\{15, 30\}$ (operating point)	10.2%	11.8%	9.5%	+2.3 $[-0.8, +5.4]$	+0.45 $[-0.05, +0.99]$	+4.4

four tasks are individually powered, are in Table 4. It reproduces on a second architecture, LLaVA-OneVision (LB +3.6/+2.1pp), and on the Palu-style projection at about half the value and below the pre-registered bar ($+1.38$, $[+0.57, +2.35]$, carried by one task). Across compressors, though, the benefiting identities do not transfer: cross-basis $H = +0.01$ ($[-0.32, +0.42]$), with cross-basis κ at chance against same-basis ceilings. A profiling table is therefore compressor-specific, and is re-profiled rather than copied when the compressor changes. Replacing gold labels by the dense model's own answer recovers $\approx$85% of the value (+2.1pp, $[+0.9, +3.4]$), extending profiling to unlabeled repeated workloads at the cost of one dense pass. Table 4 collects the per-task evidence.

The response does not transfer across questions on the same video. Repeatability across realizations leaves one recurrence untested: whether profiling a video on some questions informs its *other* questions, the case in which an answer cache cannot stand in for a rank profile. The answer is no (Table 5). We test it on CLEVRER: 300 untouched videos, three profiling and $\approx$8 held-out questions each, 13,192 forwards, the video as the bootstrap unit, with the rules and the pair $\{4, 30\}$ committed before any forward. A profiling disagreement raises the held-out disagreement rate only marginally, enrichment $1.02\times$ at that pair and $1.15\times$ at $\{15, 30\}$. The enrichment bar of 2 is unattainable here, because 87% of videos already carry a disagreement, so the lift is the reportable quantity. The profile-derived policy then earns $+0.19$pp ($[-0.97, +1.34]$) on unseen questions, against a video-level oracle of $+8.3$pp. Set beside Table 3, the two measurements identify the object: D is a property of the video–question pair, repeatable under nuisance change of that pair and not carried by the video alone.

8 Implications for deployment

The evidence supports a deployment pattern rather than a new method, and we state it as such: three phases, no learned predictor, pseudocode in Appendix C. Its held-out value is an existence proof that the repeatability of §7 can be cashed. **Profile.** For a workload whose inputs recur as new realizations—benchmark serving under re-sampled frames, a corpus re-queried over time—run the two serving ranks once and record D and, where labels exist, the signed τ. Labels are needed only on D (§6); without labels, the dense answer stands in at a measured discount. (The per-task fallback needs the ladder sweep's labels once per task—the 25–49% union—a larger budget than the pair's D, at 7–19% in Table 3.) **Store.** A response profile $x \mapsto \tau(x)$, specific to the compressor it was measured on. **Reuse.** On later occurrences, serve inputs with $\tau > 0$ at r_{hi} and the rest at

r_{lo}; inputs never profiled take the task's calibrated depth. Task-level calibration is the fallback for the unprofiled, not a layer on top of the profiled: at the operating budget the hierarchical test of §4 leaves the task prior nothing to add. Cost is dominated by serving. At $T{=}32$ the two ranks cost $c_{15}{=}14{,}992$ and $c_{30}{=}21{,}532$ visual-key parameters per head, $51\times$ and $36\times$ below the uncompressed $765{,}952$; upgrading the 6.5% of inputs profiling flags averages $15{,}418$, about $50\times$. Spending that same budget uniformly returns only $\approx+0.3$pp, against the policy's $+2.7$pp over serving everyone at r_{lo}. Upgrading everyone gains $+3.8$pp for $+44\%$ cost, of which the policy captures 71% for 6.5% of the increment.

Where the output can be cached, cache it. A recurrence exact enough to look up $\tau(x)$ is exact enough to look up the answer, and because reshuffling and re-sampling leave the gold answer unchanged, answer caches score on exactly the cross-view data. Pooled over seven view pairs ($n{=}1{,}396$) the profiling policy scores 73.3% and a cache of the r_{hi} answer 73.4% at zero forwards, against 70.6% for everyone at r_{lo} and 74.1% for everyone at r_{hi}; a cache of gold on D reaches 77.0% and of a paid dense answer 83.0%. The pattern therefore targets workloads whose outputs must be recomputed, and §7 fixes the recurring unit there as the context–query pair. What the cross-view measurement establishes is the exploitable value of repeatability.

Three costs must be kept apart: *labels* (only the disagreement union, §6), *forwards* (every candidate rank on every profiled input, once) and *statistical samples* (unchanged). Amortization repays the second: the one-time profiling pass costs $c_{15} + c_{30}$ per input, an overhead of 47% at 5 repetitions, 9% at 25 and 2% at 100, and breaks even after $M^* \approx 6$ repetitions against serving every input at r_{hi}; a dense pseudo-label pass (§7) moves M^* to $\approx$130, the price of dropping the label requirement.

9 What we claim, what we do not, and how to check

We claim that the rank a video-LLM key cache needs is set by a sparse, repeatable set of inputs whose allocation-relevant part the tested static signals do not recover, and that the policy this implies—measure once, label only disagreements, reuse the response—has positive held-out value at matched budget against a policy carrying no per-input information. We do not claim five things. That it beats caching the answer where the answer can be cached (§8). That a video's profile informs its other questions (§7). A temporal mechanism for the mode-wise win. That no a-priori signal for D exists; the interval is wide and a neighbouring label reaches 0.61–0.65. Or that the substrate result is matched-storage on all six tasks; it is on five. Scope is stated rather than hidden: everything is low-rank compression of the visual *key* cache of video LLMs, with no claim relative to quantization (Salfati, 2026) or to text-only settings. The substrate result is Qwen2-VL only and at five frames; the structural results replicate on Qwen2.5-VL (§5); the repeatability and policy results add STAR real video, a second compressor family and LLaVA-OneVision; counterfactual_inference is the weakest task throughout; the cross-question test is one query-agnostic profile on one synthetic benchmark (CLEVRER), so it bounds what that profile recovers rather than every query-agnostic signal; and nuisance axes beyond option order and frame sampling are untested. Two practices underpin the numbers: pre-registration of arms, budgets and decision rules before observation, and a compression assertion, which is a spot check of the configuration plus the degradation itself rather than a per-cell reconstruction test. The frozen files, a provenance map naming each number's source run and the scoring scripts are in the supplement.

Reproducibility statement

Every number in this paper traces to a committed run or summary file, and the supplement carries those files, the scoring scripts, the cached per-view predictions and a provenance map naming each number's source run. The nine structural accounts, the repeatability test, the policy bar and the second-backbone replication were pre-registered with their arms, budgets and decision rules frozen before their numbers were seen, and the pre-registrations ship alongside the results.

USE OF AI ASSISTANCE

An AI assistant (Claude, Anthropic) was used throughout this work under the authors' direction: to implement and debug the experiment harness, to write the scripts that generate tables from the committed run records, to run the frozen analyses, to check claims in this manuscript against those records, to build the anonymised supplement, and to draft and edit prose, including this section. All experimental designs, pre-registration rules, verdicts and decisions about what the paper claims were made by the authors, who read and are responsible for every sentence and every number. No experimental result was produced or altered by an AI system without a committed record.

REFERENCES

Ali Abbasi, Chayne Thrash, Haoran Qin, Hamed Pirsiavash, and Soheil Kolouri. IO-SVD: Input-output whitened SVD for adaptive-rank LLM compression. *arXiv preprint arXiv:2605.15626*, 2026.

Luca Benfenati, Matteo Risso, Andrea Vannozzi, Ahmet Caner Yüzügüler, Lukas Cavigelli, Enrico Macii, Daniele Jahier Pagliari, and Alessio Burrello. Don't be so Stief! learning KV cache low-rank approximation over the Stiefel manifold. *arXiv preprint arXiv:2601.21686*, 2026.

Priyansh Bhatnagar, Ashkan Moradifirouzabadi, Se-Hyun Yang, SeungJae Lee, Jungwook Choi, and Mingu Kang. STAR-KV: Low-rank KV cache compression via soft thresholding for adaptive rank control. *arXiv preprint arXiv:2606.08382*, 2026.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: Compressing KV-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*, 2024.

Alex Chen, Renato Geh, Aditya Grover, Guy Van den Broeck, and Daniel Mingyi Israel. The pitfalls of KV cache compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026a. ACL Anthology 2026.acl-long.1926.

Jian Chen, Zhuoran Wang, Jiayu Qin, Ming Li, Meng Wang, Changyou Chen, Yin Chen, Qizhen Weng, and Yirui Liu. KV-CoRE: Benchmarking data-dependent low-rank compressibility of KV-caches in LLMs. *arXiv preprint arXiv:2602.05929*, 2026b.

Anna Cordoba, Adam Puente Tercero, Nerea Angulo Hijo, Mar Linares Tercero, Julia Barrientos, Ainhoa Miranda, and Jesus Olivera. DepthWeave-KV: Token-adaptive cross-layer residual factorization for long-context KV cache compression. *arXiv preprint arXiv:2607.06523*, 2026.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2407.09141.

Lukas Haverbeck, Carmen Amo Alonso, Andres Felipe Posada-Moreno, Sebastian Trimpe, and Marco Pavone. The risk of KV cache compression. *arXiv preprint arXiv:2607.01520*, 2026.

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2401.18079.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500, 2009.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024c.

Bokai Lin, Zihao Zeng, Zipeng Xiao, Siqi Kou, Tianqi Hou, Xiaofeng Gao, Hao Zhang, and Zhijie Deng. MatryoshkaKV: Adaptive KV compression via trainable orthogonal projection. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.14731.

Zhuowen Liu, Longkun Hao, Shiyu Feng, Xiaowen Chang, Ruiqun Li, and Changqun Li. LACE-SVD: Loss-aware SVD with cumulative error correction for LLM compression. *arXiv preprint arXiv:2607.03057*, 2026.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. *International Conference on Machine Learning (ICML)*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Samuel Salfati. Quantization dominates rank reduction for KV-cache compression. *arXiv preprint arXiv:2604.11501*, 2026.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for KV cache compression. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 15332–15344, 2024. arXiv:2408.05646.

Sungbin Shin, Wonpyo Park, Jaeho Lee, and Namhoon Lee. Rethinking pruning large language models: Benefits and pitfalls of reconstruction error minimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. Adaptive KV-cache compression without manually setting budget. *arXiv preprint arXiv:2509.03136*, 2025.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. *arXiv preprint arXiv:2410.23317*, 2024.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations (ICLR)*, 2025.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. ThinK: Thinner key cache by query-driven pruning. In *International Conference on Learning Representations (ICLR)*, 2025. Spotlight; arXiv:2407.21018.

Bohan Yu and Yekun Chai. EvolKV: Evolutionary KV cache compression for LLM inference. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025. arXiv:2509.08315.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. DynamicKV: Task-aware adaptive KV cache compression for long context LLMs. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025. arXiv:2412.14838.

A SUBSTRATE VERIFICATION

Two checks and a provenance rule make this a compression effect rather than an accounting one. Engagement is verified by a spot check of the configuration (reconstruction error strictly positive and retained rank below full on the compressed modes; the temporal mode is uncompressed at five frames) and by the degradation itself—both methods fall below dense on every task—so the cell is not a silent no-op. The contrast is flat against mode-wise rather than Tucker against everything: a CP arm at $R{=}20$ in the same runs, at a comparable key ratio, lands within a point of Tucker over the same six tasks. Budget fairness is verified per sample, not on average. On five tasks the logged key ratio is $18.9\times$ (Tucker) against $18.7\times$ (SVD) on every sample, so the $+12.6$pp mean over those five is a matched-storage result. The sixth is not. On action_antonym the clips are short, so the Tucker core and channel factor outweigh SVD's rank-6 factors on every sample ($11.8\times$ against $16.6\times$, $\sim 40\%$ more parameters); its $+2.0$pp is descriptive only (Table 1). Every number traces to a saved run file.

B THREE CHECKS ON THE CENTRAL MEASUREMENT

Computed from the committed prediction records, with no new inference. The AUC used here is validated on a positive, a negative and a tie control first; the tie control caught an ordinal-rank error in the first version.

How large is D against the floor the accuracy gap forces. Two ranks differing by $b - c$ correct answers must disagree on at least $|b - c|$ inputs, so $|D|/N$ has an arithmetic floor. Over the twelve confirmatory cells $|D|/N$ averages 0.117 against a floor of 0.039, a factor of about three; per cell the ratio runs from 1.0 (Palu counterfactual, where $b{=}c$ makes the floor zero and the ratio undefined, and Palu object_existence, where the floor is the whole of D) to 11.0.

The dense margin does not explain the repeatability. Predicting membership in D from the dense model's top-2 option-logit margin alone gives AUC 0.613 pooled, above chance and below the 0.755 the same features reach for fragility. But conditioning on it does not remove the repeatability:

dense-margin quartile	n	$P(D_2)$	$P(D_2 \mid D_1)$	$P(D_2 \mid \neg D_1)$	lift	κ
Q1 (lowest)	597	15.7%	45.8%	10.0%	+0.36	0.36
Q2	599	17.5%	55.3%	9.7%	+0.46	0.45
Q3	598	10.0%	65.5%	4.1%	+0.61	0.61
Q4 (highest)	598	3.3%	77.3%	0.5%	+0.77	0.80
pooled	2392	11.7%	55.9%	5.8%	+0.50	0.50

The lift is largest where the margin is largest, the opposite of what the margin explanation predicts.

The model's own test-retest agreement. Predicted letters are not comparable across option-shuffled views, so the comparable quantity is correctness. Dense correctness repeats at $\kappa{=}0.735$ pooled over the same twelve cells (1.000, 0.666 and 0.673 on the three frame-resample cells). D repeats at $\kappa{=}0.50$, so it is less stable than the model's own correctness; what a profiling policy needs is not that the two match but that D be far more predictable than its own base rate, 11.7% to 55.9%.

C THE POLICY IN PSEUDOCODE

The three phases of §8, stated as an algorithm.

Algorithm 1 Measure, then amortize (one compressor; a workload whose inputs recur as new *realizations*)

Require: serving pair $(r_{\mathrm{lo}}, r_{\mathrm{hi}})$; an input $x =$ a (video, question) whose later occurrences differ in realization (option order, sampled frames) *and whose output must be recomputed*—if the answer itself can be cached, cache it (§8); per-task calibrated depth $r_d^\star(t)$ from the labelled ladder sweep of §4, if one exists; labels $y(x)$ if available, else the dense answer $\hat{y}_{\mathrm{dense}}(x)$

1: **Profile (once, two forwards per input):** run r_{lo} and r_{hi}; set $D(x) = \mathbf{1}[\hat{y}_{r_{\mathrm{lo}}}(x) \neq \hat{y}_{r_{\mathrm{hi}}}(x)]$

2: label only $\{x : D(x) = 1\}$ (§6); for those set $\tau(x) = \mathbf{1}[\hat{y}_{r_{\mathrm{hi}}}(x) = y(x)] - \mathbf{1}[\hat{y}_{r_{\mathrm{lo}}}(x) = y(x)]$, else $\tau(x) = 0$

3: **Store:** table $T \leftarrow \{x \mapsto \tau(x)\}$, tagged with the compressor it was profiled on

4: **Reuse (each later realization of x; one table lookup): if** $x \in T$ and $\tau(x) > 0$ **then** serve at r_{hi} **elif** $x \in T$ **then** serve at r_{lo} **else** serve at $r_d^\star(t(x))$ (fallback for unprofiled inputs; not exercised in our experiments, where every evaluated input is profiled, and worth $\approx$0.2pp over a static rank by §4)

5: **Re-profile**, never copy T, when the compressor changes (§7; on the second compressor family the re-profiled value cleared zero but not our bar)