Same Pool, Different Answer: Stable Best-of-N Selection for Vision-Language Models

Pengfei Zheng

Independent Researcher, China
pengfeizheng21@gmail.com

Abstract. A common way to improve vision-language model outputs is Best-of-N (BoN) selection: generate N candidate answers, score each one, return the best. When the candidate pool and compute budget are held fixed, the selected answer should be reproducible. We show it often is not. Slightly adjusting the softmax temperature, toggling a probability-truncation option, or enabling dropout noise during the scoring pass is enough to change the winning answer on 41–47% of test items — even though the candidate pool is identical. This failure is specific to *likelihood-based* selection, where candidates are ranked by policy likelihoods or policy/reference likelihood ratios; a deterministic reward-model or rule-based (e.g. RLVR) reranker assigns identical scores to identical text and is stable on a fixed pool by construction.

Standard scoring conflates answer quality with two confounding factors: when token-level scores are summed, low-value suffixes can change the total without improving answer quality, and score magnitudes shift whenever the model is updated. We propose DSPA, which separates two concerns into a test-time selector and a training-time filter. At test time, we score each candidate by comparing it token-by-token against a fixed reference model and averaging the result, which removes the additive length accumulation of summed scores and keeps all scores on a fixed reference scale; this is the component responsible for selection stability. At training time, we filter preference pairs to prevent shortcuts based on answer length, verbatim prompt copying, or hallucinated objects; this component improves the factual quality of the candidate pool, not the stability of selection. We evaluate stability by replaying the scoring step on the same frozen pool under controlled perturbations (a protocol we call SRP), where the anchored selector reduces the answer-flip rate from 44.3% to 24.9%. Manual inspection shows roughly 70% of remaining flips are near-ties that differ only in phrasing. On identical pools, the anchored selector also reduces object hallucination (POPE) from 9.6% to 6.2%; the stability gain holds at 13B scale, while the utility gain transfers to Qwen2.5-VL-7B-Instruct.

Keywords: Vision-Language Models · Best-of-N Sampling · Test-time Selection · Reproducibility · Hallucination Mitigation

1 Introduction

Imagine two teams deploying the same vision-language model with the same compute budget. Both generate eight candidate responses to each user query and return the one that scores highest — a strategy known as Best-of-N (BoN) selection. Same candidate pool, same compute. They should pick the same answer. They often do not.

Under small changes to the scoring procedure — a temperature shift of ± 0.05, a minor change in how probability tails are handled, or simply enabling dropout noise during the scoring pass — the standard baseline selector flips its winner on 41–47% of items (Table 2, Fig. 2). Our fix, an anchored selector (Sec. 4.1), lowers this residual flip-rate to 24.9%, but the flips that remain are not all harmless. Among the anchored selector's residual flips, we annotated 500 cases and found that roughly 29% involve factually non-equivalent winners — for instance, one candidate correctly denies the presence of an object while the other hallucinates it. Combining this conditional proportion with the 24.9% residual flip-rate yields an estimated factual-content change probability of about 7.2% (6–8%).

This matters beyond our particular setup. The "generate several, pick the best" pattern appears in rejection sampling for alignment [2, 30], pass@k selection in code generation [3], and self-consistency voting [37]. We state our scope at the outset: the instability we study is specific to *likelihood-based* selection, in which candidates are scored by the policy's own likelihoods or by policy/reference likelihood ratios — the common choice when one wants Best-of-N without training or serving a separate reward model. A deterministic reward-model or rule-based (e.g. RLVR) reranker assigns the same score to the same text regardless of softmax temperature or probability truncation, so on a fixed pool it is stable by construction; that regime is complementary and outside our scope. Most prior work has studied what goes wrong when the pool grows large — the selector starts gaming the scoring function, a failure called reward hacking [9, 18]. The failure we study is more basic: the pool is small, it never changes, and the likelihood-based scoring step alone still cannot agree with itself.

The instability has two sources, both rooted in how standard scoring works. When token-level scores are summed over the full sequence, response length can affect the total mechanically: a 40-token answer that correctly says "The cat is on the couch" and then rambles for 20 more tokens can change the relative ranking against the concise 20-token version, not because it is better, but because the additional tokens alter the sum. Separately, whenever the model is fine-tuned, all score magnitudes shift, and the gap between the top two candidates can narrow or reverse unpredictably. We adopt length normalization (dividing by token count) as our primary baseline; it removes the crudest accumulation but still scores on the policy's own scale, which shifts with model updates. Our ablations (Table 4) confirm that a fixed reference with per-token averaging is substantially more stable than either alone.

On identical N=8 candidate pools, switching only the selection rule reduces object hallucination (POPE [23]) from 9.6% to 6.2% (Table 1). Which selection rule you use changes whether the output is correct.

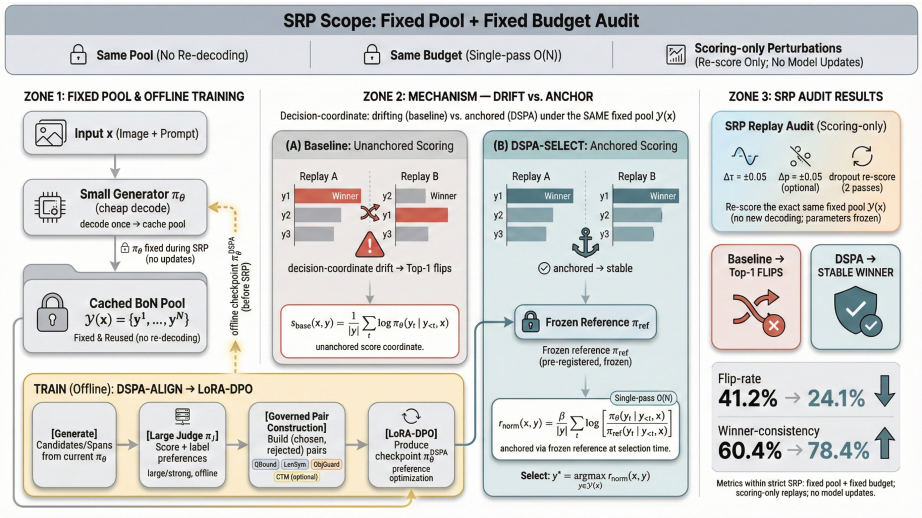

Fig. 1: Overview. *Problem:* on a fixed candidate pool, small scoring perturbations often change which answer wins (41–47% of items). *Solution:* we filter preference pairs during training (Sec. 4.2) and, at selection time, score each candidate token-by-token against a fixed reference, average, and return the top candidate in one pass (Sec. 4.1). *Audit:* SRP replays only the scoring step on the same frozen pool. Numeric values are from the dropout family (n=200); Table 2 averages all three families (n=500).

Our approach. We fix the scoring step itself with a method we call DSPA, which has two parts addressing different failure modes: a test-time *anchored selector* that scores each candidate token-by-token against a fixed reference model and averages the result (Sec. 4.1), and a training-time *filter* that removes length, verbatim prompt-copying, and object-hallucination shortcuts from the candidate pool (Sec. 4.2).

To evaluate stability without conflating it with extra computation, we replay only the scoring step while keeping the candidate pool and budget fixed, a protocol we call SRP (Same-pool/Same-budget Replay). It gives us two clean metrics: how often does the winner change (flip-rate), and how often does each replay agree with the original winner (agreement rate).

Contributions. (i) On an identical cached pool, scoring-only perturbations flip the BoN winner on 41–47% of items, and changing only the selection rule also shifts hallucination outcomes (POPE 9.6%→6.2%; Table 1). (ii) Using SRP, we show a log-ratio selector is stable exactly when the reference is frozen and sees the full input *and* scores are averaged per token rather than summed (removing either returns flip-rate to near-baseline; Table 4). (iii) We pair this selector with a preference-construction recipe that reduces length, copying, and hallucination confounds in the pool.

2 Related Work

Inference-time alignment and reward hacking. Sampling multiple candidates and scoring them is central to instruction-following alignment [2,30] and code generation [3]. As the pool grows, the selector can exploit scoring artifacts — reward hacking [9, 18] — and regularized BoN variants constrain how far the selected distribution deviates from the original model [1, 16, 17, 35]. We ask a different question: is the winner even reproducible when N and the pool are held fixed?

Replay-style evaluation. Recent work caches candidate pools to compare selectors under imperfect verifiers [5, 14, 38], and self-consistency aggregation highlights sensitivity to protocol choices [24,37]. SRP enforces a stricter boundary — same pool, same budget, pre-specified perturbations — and measures stability rather than comparing algorithms.

Reference-based scoring: same formula, different goals. The per-token log-ratio between two models has appeared in dialogue reranking for diversity (MMI [21]), generation-time steering away from degenerate text (contrastive decoding [4, 22, 25]), and preference learning (DPO [31]). These uses share the arithmetic but target different objectives, and none defines a stability target over a fixed pool.

When the same formula is repurposed for stable *selection*, two things go wrong. First, if scores are summed over all tokens (as in MMI), response length affects the total mechanically: a near-neutral suffix can shift the ranking against a more correct but shorter answer without improving it. Second, if the reference is the model's own distribution (per-token PMI) or is periodically updated (some DPO variants), the reference drifts with the model, and the same pool can rank differently after a minor update. Per-token averaging addresses the first; a pre-committed fixed reference addresses the second. Table 4 confirms that removing either one returns the flip-rate to near-baseline levels.

Hallucination benchmarks [7, 23, 41] and aligned VLMs [31, 39, 40] provide our factuality baselines; decoding-time strategies [15, 20] can be applied after our selector.

3 The SRP Audit Protocol

Before describing our method, we define the evaluation boundary that governs all primary claims.

Motivation. BoN selection has three steps: generate a candidate pool, score each candidate, return the highest-scoring one. The question is whether step (3) is reliable — whether the same pool always yields the same winner. If we regenerated the pool each time we tested, we could not tell whether a different winner came from a different pool or from scoring instability. Freezing the pool isolates the scoring step, the same way one tests a bathroom scale by stepping on it twice rather than switching people between measurements.

As a concrete case, on a "Is there a dog in this image?" query a temperature shift of 0.05 — small enough that most practitioners would consider it negligible — can flip the winner from "No, I see a cat on the rug" to "Yes, there appears to be a small dog," on the very same eight-candidate pool.

Definition. SRP (Same-pool/Same-budget Replay) replays only the scoring step while keeping everything else fixed. It controls four variables:

1. **Same pool.** The candidate set $\mathcal{Y}(x) = \{y^{(1)}, \ldots, y^{(N)}\}$ is identical across all replays; no new decoding occurs.
2. **Same budget.** Each replay scores each candidate exactly once ($O(N)$ per item).
3. **Pre-specified perturbations.** Only lightweight scoring perturbations are applied, from families defined before the experiment (full pseudocode in supplementary Alg. S1).
4. **Pre-specified reference.** The reference model π_{ref} is frozen and identified by its checkpoint hash before any SRP runs, preventing retroactive adjustments.

Ties are broken deterministically by smallest candidate index in the frozen pool manifest.

What perturbations, and why? Each perturbation family represents a realistic source of scoring variation that practitioners encounter across deployments.

Temperature jitters (± 0.05 around the native τ_0=0.7), the magnitude routinely produced by different calibration runs or framework defaults [11]. Because temperature rescales logits before the per-position softmax, a ± 0.05 change is *not* a global monotone rescaling of the length-normalized scores, so near-tied candidates can swap order.

Score-time probability truncation (± 0.05 around p_0=0.9): a top-p filter applied at *scoring* time (not generation), keeping the most probable tokens up to cumulative mass p and re-distributing the rest, which frameworks handle differently [12]. Like temperature it re-normalizes per position, so it need not preserve near-tie ordering; we always include the realized token before re-normalizing to avoid zero probabilities.

Dropout-enabled scoring (2 replays, distinct fixed seeds). Inference normally disables dropout, but misconfigurations happen; we use it as a simple, reproducible stochastic perturbation, not a model of hardware noise [8].

All perturbations apply symmetrically to both policy and reference. An asymmetric variant (perturbing only the policy) gave equivalent results, ruling out correlated noise cancellation as the source of the stability gain (supplementary material).

Metrics. We report two metrics per perturbation family.

Flip-rate: the fraction of test items where at least one replay in the family changes the winner.

$$\text{Flip-rate}_t = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{1}\Big[\exists\, p \in \mathcal{P}_t : y_p^*(x) \neq y_0^*(x)\Big].$$

A flip-rate of 44% means that for 44% of test items, at least one perturbation changed the winner.

Agreement with original winner: the average per-replay match rate.

$$\text{Agreement}_t = \frac{1}{|\mathcal{X}|\,|\mathcal{P}_t|} \sum_{x \in \mathcal{X}} \sum_{p \in \mathcal{P}_t} \mathbf{1}\big[y_p^*(x) = y_0^*(x)\big].$$

With one replay per family, agreement $= 1 -$ flip-rate. With two replays, a single discordant replay suffices for a flip but only partially reduces agreement, so we report both. Wilson interval definitions and selected paired tests are reported in the supplementary material.

Budget accounting. SRP caches candidate pools; replays only rescore. All primary selectors run in $O(N)$ time with a single pass per candidate. The anchored selector additionally evaluates the fixed reference, increasing wall-clock *scoring* time by approximately 1.8× per candidate. This 1.8× is a scoring-stage figure, not a doubling of end-to-end Best-of-N cost: candidates are first produced by autoregressive decoding (the dominant cost), whereas scoring is teacher-forced and can be run in a single parallel pass. In our N=8, ∼100-token setting, we estimate, from the decoding/scoring time decomposition, that the extra reference pass adds roughly +5–10% to end-to-end wall-clock rather than 1.8×. The selector remains single-pass $O(N)$; we exclude pairwise methods such as minimum Bayes risk (MBR) decoding, which require $O(N^2)$ comparisons [6].

4 Method

At test time, we score each candidate against a fixed reference model and average across tokens; this is the component responsible for selection stability. At training time, we filter preference pairs to remove systematic biases from the candidate pool; this component improves the factual quality of the pool rather than the stability of selection. We call the combination DSPA, and we isolate the two effects separately: selection stability in Sec. 5.5 (varying the selector while training is held fixed) and pool factuality in Sec. 5.4 (varying the filters while the selector is held fixed).

The core idea is short enough to say in one sentence: instead of measuring raw probability (which conflates quality with length and model state), we measure how much the fine-tuned model prefers each candidate *relative to a fixed reference*.

4.1 Test-time selection: the anchored selector

We want a score where a longer answer does not automatically win, and where the reference point stays fixed even when the policy changes.

To satisfy the first requirement, we score each token individually and *average* rather than sum, so that appending uninformative tokens dilutes the score instead of inflating it. For the second, we compare each token's probability under the fine-tuned model (the *policy*, π_θ) against a *fixed* reference model (π_{ref}) — one that is frozen before evaluation and never updated. This reference sees the full input — image, prompt, and all preceding tokens — so it provides a meaningful comparison, not a context-free prior that ignores the image.

Given input x (image + prompt) and candidate $y = (y_1, \ldots, y_{|y|})$:

$$r_{\text{norm}}(y \mid x) \;=\; \frac{1}{|y|} \sum_{t=1}^{|y|} \log \frac{\pi_\theta(y_t \mid y_{<t}, x)}{\pi_{\text{ref}}(y_t \mid y_{<t}, x)}. \tag{1}$$

At each position t, the log-ratio $\log[\pi_\theta(y_t)/\pi_{\text{ref}}(y_t)]$ measures how much more the policy prefers that particular token than the reference does. Averaging over all $|y|$ tokens yields a per-token score that does not mechanically increase with answer length. Selection is a single $O(N)$ pass: return $\text{argmax}_{y \in \mathcal{Y}(x)} r_{\text{norm}}(y \mid x)$.

To see why averaging matters, appending 20 low-value tokens (mean log-ratio 0.02) to a correct 20-token answer (mean 0.15) raises the *summed* score from 3.0 to 3.4 — so the padded answer wins — but lowers the *averaged* score from 0.15 to 0.085, so the concise answer wins. The fixed reference plays a complementary role, and two natural alternatives are worse. Using the policy's own *context-free* distribution as the denominator — per-token PMI, i.e. $\pi_\theta(y_t \mid y_{<t})$ with the image and prompt removed — keeps the score well-defined and non-zero, but measures each candidate against a prior that ignores the image. Using a periodically updated copy of the policy makes the denominator co-move with the policy, so the same pool can rank differently after a minor update. Pre-committing the reference checkpoint before evaluation avoids both: the reference side of the ratio neither ignores the input nor drifts with the policy. (The numerator still changes across policy snapshots, so the score is not literally frozen; what is frozen is the baseline against which all candidates are measured.)

Table 4 isolates each constraint. Removing per-token averaging (using sequence-level sums) substantially increases the flip-rate; replacing the fixed reference with the policy's own distribution or a periodically updated copy also degrades stability. Only the combination achieves the full gain. A complete selector taxonomy is in supplementary Tab. S5.

4.2 Training-time filtering

A stable selector is necessary but not sufficient — it still picks from whatever pool it is given. During preference training, vision-language models pick up shortcuts that systematically corrupt the pool, and we address the three most damaging ones.

Prompt copying. Vision-language models sometimes copy large verbatim chunks of the prompt (especially on caption-style questions), which looks superficially correct but reflects no understanding; we exclude any candidate whose longest contiguous verbatim overlap with the input exceeds K=12 tokens.

Answer length. Annotators and model judges consistently prefer longer answers even when length adds no information [19,29,33], teaching the model that "longer is better"; we constrain the preferred/rejected length ratio to the band 0.8–1.2.

Object-existence hallucination. If the preferred candidate in a "Is there a [object]?" pair incorrectly says yes, the model learns to affirm object existence regardless of image content — a shortcut that directly produces hallucinations. For prompts matching object-existence patterns (matched against training-set labels only; no evaluation annotations used), we keep a pair only when the preferred candidate does not hallucinate the triggered object.

After filtering, candidates are segmented into sub-sentences, scored by a large vision-language judge (LLaVA-NeXT-34B [27]), and assembled into (preferred, rejected) pairs with at most two pairs per input. An optional fourth filter — masking copy-prone tails in caption prompts — is enabled in the full system; its benefit depends on prompt type (supplementary Tab. S4, block B). The policy is fine-tuned via LoRA [13]-DPO [31].

4.3 The interaction between selector and pool

Training-time filtering improves the pool but does not by itself stabilize ranking: if the selector still has a length bias, a verbose candidate can win under one scoring configuration and lose under another. Conversely, the anchored selector stabilizes ranking but cannot recover factual accuracy from a pool filled with biased candidates. The two components therefore target different axes — the selector drives stability, the filters drive pool factuality — and two existing results make this split concrete. Table 4 varies only the selector (training held fixed) and moves the flip-rate from 40.8% to 24.1%, identifying the selector as the dominant lever for stability. In the other direction, a separately aligned model (RLAIF-V [40]), which changes the pool but uses a standard length-normalized selector, reduces the flip-rate only from 44.3% to 35.1% — far short of the anchored selector's 24.9%. This separately-aligned-model comparison is consistent with training alone being insufficient to match the anchored selector, although it is not a controlled ablation of the training filters; separately, removing all training filters raises object hallucination fourfold (Table 3). We therefore attribute selection stability to the anchored selector and treat the training filters as a pool-factuality mechanism; we do not claim the filters themselves improve flip-rate or winner consistency.

5 Experiments

All stability results follow SRP: candidate pools are generated once with fixed seeds, re-scored under perturbations, and selected in a single $O(N)$ pass. No new decoding occurs during evaluation.

5.1 Setup

Models. We use LLaVA-1.5 7B [26] and train with LoRA [13]-DPO using preferences scored by LLaVA-NeXT-34B [27]. The fixed reference is the matched-size pre-fine-tuning checkpoint (LLaVA-1.5 7B before alignment). Pools use N=8 candidates unless otherwise noted. Unless otherwise stated, SRP stability is evaluated on a fixed 500-item subset sampled once from the LLaVA-Instruct validation split; the same subset is used for all selectors and perturbation families.

Baselines. For LLaVA-1.5 7B, all multi-sample selectors operate on identical cached pools so that differences reflect only the scoring rule. We compare four baselines, all under the same SRP protocol: **Greedy** (N=1, a single-output reference); **BoN + length-norm**, the standard baseline scoring each candidate by its mean per-token log-probability under the policy alone (no reference); **ϵ-margin tie-break**, applied on top of length-norm, returning the shorter of the top two when they differ by less than ϵ=0.02 (targeting near-ties without changing the scoring function); and **RLAIF-V 7B** [40], a separately aligned model showing what alignment training alone achieves.

Perturbations. As defined in Sec. 3: temperature jitters ($\pm$0.05), score-time probability truncation ($\pm$0.05), and dropout (2 seeds), applied symmetrically to policy and reference. These define a controlled local perturbation neighborhood; we do not claim they reproduce any particular precision or framework change.

Baseline scope. We restrict comparisons to single-pass $O(N)$ selectors on a cached pool using only the policy (and optionally a fixed reference). Methods based on deterministic reward models are outside our perturbation scope because their scores are unaffected by the likelihood-level perturbations considered here, not because they change the pool. Methods that alter the *sampling* distribution rather than re-ranking a fixed pool — including KL-constrained [17] and Soft-BoN [35] variants — are outside scope because they change the candidate pool itself. Pairwise methods such as MBR ($O(N^2)$) are excluded on budget grounds [6].

Metrics. Hallucination is measured in two ways. *Benchmark-grounded* (POPE [23]): binary "Is this object in the image?" questions evaluated against ground-truth labels; the hallucination rate is the fraction of incorrect "yes" answers. *Judge-estimated*: a large VLM judge assesses overall factuality; reported with a "judge-est." note and *not* directly comparable to POPE scores.

Table 1: Task utility of the selected output. LLaVA-1.5 7B; all multi-sample selectors use identical cached pools (same seeds, N=8), so differences reflect only the selection rule. Greedy (N=1) is a single-output reference. POPE: fraction of incorrect "yes" answers on binary object-existence questions. ϵ-margin tie-break: return the shorter candidate when the top-two length-norm gap < 0.02. CI conventions are described in the supplementary material.

Selector	VQAv2↑	OK-VQA↑	TextVQA↑	MME↑	MMMU↑	POPE halluc. rate↓
Greedy (N=1)	78.5	59.2	58.1	1510.5	35.1	10.5
BoN + length-norm	79.4	60.8	59.6	1532.8	36.4	9.6
BoN + ϵ-margin tie-break	79.1	60.5	59.3	1528.4	36.2	9.8
BoN + anchored selector (ours)	**79.7**	**61.2**	**60.1**	**1541.0**	**36.8**	**6.2**

We also report stability (flip-rate and agreement with original winner; Sec. 3), AMBER accuracy [36], pairwise win rate vs. runner-up, human preference (forced-choice on 100 items, 2 annotators, 200 total judgments; supplementary material), and standard benchmarks (VQAv2 [10], OK-VQA [28], TextVQA [32], MME [7], MMMU [41]).

5.2 Task utility

Table 1 shows that the anchored selector does not sacrifice standard benchmark accuracy. For LLaVA-1.5 7B, all multi-sample selectors operate on identical pools; greedy (N=1) is shown as a single-output reference. VQAv2 gains are modest — unsurprising given the fixed pool ceiling.

The ϵ-margin tie-break yields no POPE improvement (9.8% vs. 9.6% for length-norm). Resolving near-ties alone, however, does not improve factuality: tie-breaking without changing the scoring function turns out to be insufficient. The notable result is POPE hallucination: the anchored selector cuts the rate from 9.6% to 6.2%, a 35% relative reduction on the same candidates.

5.3 Stability under scoring perturbations

Table 2 reports stability under SRP. The baseline BoN selector flips its winner on 41–47% of items (average 44.3%); the anchored selector reduces this to 24–26% (average 24.9%), and agreement with the original winner rises from 55–60% to 77–78%.

The ϵ-margin tie-break helps modestly (37.1%), but it does not change how scores are computed and does not improve factuality (POPE 9.8% vs. 6.2%). RLAIF-V, a separately aligned model with a standard length-normalized selector, lands between the baseline and the anchored selector on flip-rate, consistent with training alone being insufficient to match the selector (Sec. 5.5). As a cost-effectiveness trade-off, the anchored selector reaches 24.9% flip-rate (POPE 6.2%) for an estimated +5–10% end-to-end overhead (not a doubling; Sec. 3), versus 37.1% for the near-free tie-break and 35.1% for training-only RLAIF-V.

Table 2: Stability under SRP (averaged across perturbation families). Per-family breakdown in Fig. 2; full table in supplementary Tab. S3. n=500. CI conventions are described in the supplementary material.

Model	Flip-rate (avg%) ↓	Agreement with orig. winner (avg%) ↑
BoN + length-norm (LLaVA-1.5 7B)	44.3	57.2
BoN + ϵ-margin tie-break	37.1	63.2
RLAIF-V 7B (length-norm) [40]	35.1	65.6
Anchored selector (ours)	**24.9**	**77.5**

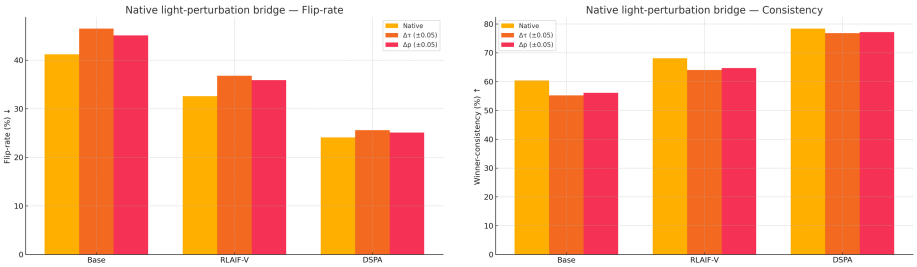

Fig. 2: Stability per perturbation family. Left: flip-rate ↓; right: agreement with the original winner ↑. Bar groups correspond to the compared selectors: length-normalized baseline, RLAIF-V, and the anchored selector. Within each group, bars correspond to the three SRP perturbation families. The in-plot label "Native" denotes the dropout replay family with native score-time τ and p but dropout-enabled scoring seeds; the other two bars denote score-time temperature perturbation $\Delta\tau$ (±0.05) and score-time top-p perturbation Δp (±0.05). Relative to the length-normalized baseline, the anchored selector improves the across-family averages by 19.4 pp in flip-rate and 20.3 pp in agreement. Exact per-family values are reported in supplementary Tab. S3.

Looking at individual flip cases, we see three recurring patterns. In some, the baseline flip is factuality-damaging: the perturbed winner hallucinates. In others, the anchored selector holds steady under the same perturbation. The most common pattern, however, is a benign near-tie — both candidates convey the same information in slightly different words. Representative flip categories are summarized in supplementary Tab. S1; near-ties account for the majority of residual instability (Sec. 6).

5.4 Ablation: training-time filters

Table 3 ablates the training filters while keeping the anchored selector. Removing any one filter causes a moderate increase in judge-estimated object hallucination (6.2%→6.9–7.6%). Removing all three, however, produces a disproportionate fourfold jump to 23.6%.

Table 3: Training-filter ablation (summary). All rows use the anchored selector. "All three primary filters off" refers to the length-matching, prompt-copy, and object-existence filters of Sec. 4.2; the optional fourth filter (caption-tail masking) remains enabled in all rows, as in the full system. Object hallucination rate and overall hallucination rate: judge-estimated (not directly comparable to POPE). Additional single-off summaries and pool-size comparisons are reported in supplementary Tab. S4.

Setting	Object halluc. rate↓ (judge-est.)	Overall halluc. rate↓ (judge-est.)	AMBER accuracy↑	Pairwise win rate↑	Human pref. (wins/200)↑
DSPA (full: filters + anchored selector)	**6.2**	**29.7**	**81.7**	**48.2**	**192**
RLAIF-V 7B (anchored selector)	9.6	43.9	80.1	46.8	–
All three primary filters off	23.6	36.6	77.5	45.8	182

This suggests a compounding effect. Without length matching, the model favors verbose candidates. Verbosity creates more opportunity for prompt copying. And copying-heavy candidates can mask hallucinated objects that the object-existence filter would have caught. Once all three filters are gone, the biases feed off each other. Pool-size ablations (supplementary Tab. S4, block C) tell a consistent story: for DSPA, moving from greedy to N=8 cuts object hallucination from 10.5% to 6.2%; for RLAIF-V, the drop is smaller (12.1%→9.6%).

5.5 Which constraint drives stability?

Table 4 holds the training procedure constant and varies the selector. We test sequence-level summation (reintroduces length bias), self-anchoring against the policy's own *context-free* distribution (per-token PMI: the denominator $\pi_\theta(y_t \mid y_{<t})$ drops the image and prompt, so it is well-defined but ignores the visual input), a co-moving reference that is periodically updated (score magnitudes drift), and the full anchored selector.

Sequence-level summation brings the flip-rate back to 40.8% — nearly as bad as the unanchored baseline. Self-anchoring and a co-moving reference fare somewhat better (35.4% and 33.6%) but are still far from the 24.1% achieved by the full design. Both constraints matter; neither alone is enough.

5.6 Pool size and length

Varying the pool size N (supplementary Fig. S2) shows that, with the anchored selector, larger N reduces hallucination and improves the pairwise win rate *without* inflating response length, whereas the unanchored baseline produces steadily longer outputs (the length-accumulation bias of Sec. 4.1). This is a quality-vs-N analysis, not stability-vs-N: because changing N also changes the compute budget, it falls outside SRP's same-pool/same-budget boundary, and we restrict stability claims to the fixed N=8 setting.

Table 4: Selector constraint ablation. Each row relaxes one constraint of the anchored selector while keeping training-time filtering fixed. n=200; dropout family only.

Selector variant	Flip-rate ↓	Agreement with orig. winner ↑
Seq-level log-ratio (no per-token avg)	40.8	58.6
Per-token, self-anchored (PMI; context-free denominator)	35.4	64.9
Per-token, co-moving reference	33.6	66.2
Anchored selector (fixed ref + per-token avg, ours)	**24.1**	**78.4**

5.7 Generalization

At 13B scale, the anchored selector reduces the average SRP flip-rate from 38.5% to 22.3%, consistent with the 7B results. Applying it to Qwen2.5-VL-7B-Instruct [34] without architecture-specific tuning reduces judge-estimated object hallucination from 38.4% to 8.7% (matched-family Qwen2.5-VL-32B-Instruct judge); the higher Qwen baseline (vs. 9.6% for LLaVA) reflects judge-estimated rates on open-ended responses, not the binary POPE of Table 1, so the two are not directly comparable though both reductions are large. We measure SRP flip-rate directly only on LLaVA (including 13B); the Qwen result is a cross-family transfer of the *utility* gain, not a stability measurement, so taken with the 13B result it is consistent with the instability arising from how likelihood scores are computed rather than from a model family — but our cross-family evidence is for utility, not flip-rate. Full results in supplementary Tab. S6–S7.

5.8 Reference robustness

Does stability depend on the reference checkpoint? On a 200-item subset, swapping to a different-family reference or a union pool leaves flip-rate and agreement within overlapping confidence intervals (all McNemar tests non-significant, $p > 0.5$); a stronger reference is slightly better, but on this subset we detect no significant difference, which bounds rather than establishes equivalence. A placebo shuffle gives ∼50% agreement (the null band), and a delete-k test shows the selector stays well above null as the pool shrinks (supplementary Fig. S1, Tab. S2–S3).

6 Discussion

The anchored selector brings the flip-rate from 44.3% down to 24.9%. As noted in the introduction, roughly 29% of remaining flips involve factually different answers — a residual risk of about 6–8%. Here we examine those residual flips in more detail.

Most remaining flips are harmless. We sampled **500** of the anchored selector's flip cases uniformly from all perturbation families (these are individual flips, not the same as the fixed 500-item SRP evaluation subset from Sec. 5) and had two independent annotators label each one. A *flip case* is an (item, perturbation family) pair where the winner changes; one item can contribute up to three cases. A flip is *semantically equivalent* if the two winners convey the same factual content and differ only in phrasing — "The cat is on the couch" versus "There is a cat sitting on the couch." Inter-annotator agreement was κ=0.82 (Cohen's κ).

71% of flips turned out to be semantically equivalent near-ties. Holding the observed average flip-rate fixed at 24.9%, the 29% factual non-equivalence proportion implies an estimated factual-change probability of roughly 7.2%; the Wilson 95% CI for that conditional proportion, propagated at the fixed flip-rate, is 6.3–8.2%. Because a single item can contribute up to three flip cases, these cases are not fully independent, so this interval should be read as conditional on the point flip-rate rather than as a joint confidence interval; a per-item cluster bootstrap is left to future work. Annotation protocol and per-category breakdowns are in the supplementary material.

The remaining ∼29% of flips do involve genuinely different answers, typically when candidates cluster near the decision boundary (score gap < 0.02). The anchored selector shifts the score-gap distribution toward larger margins (supplementary App. D), which is why most items are now stable.

Three sources of residual instability stand out: benign near-ties between semantically equivalent candidates (the most common, inflating the metric without affecting factuality); cross-family references, which compress score margins and widen the confidence interval; and a hard floor — with all training filters removed, the selector avoids length inflation but cannot conjure factual content that was never generated.

Limitations. SRP evaluates stability within a specific perturbation neighborhood — temperature jitters, probability truncation, dropout. A full precision switch or a major architecture change could behave differently, and we have not tested those. Our primary experiments use LLaVA-1.5 7B, with a 13B check and cross-architecture transfer to Qwen2.5-VL-7B-Instruct; we would like to see results on a wider range of models. The annotation study covers 500 flip cases with two annotators (κ=0.82), which is enough to establish the rough proportion of benign flips but not enough for fine-grained subcategory analysis. Two evidence gaps also remain by design. First, we isolate the stability contribution of the selector (Table 4) but do not separately report flip-rate or winner consistency for the training-filter ablations of Table 3; we therefore frame the filters as a pool-factuality mechanism rather than a stability mechanism, and a direct stability ablation of the filters is left to future work. Second, our stability claims are made at the fixed N=8 budget; how flip-rate and winner consistency scale with N — which also changes the compute budget — falls outside the same-pool/same-budget scope of SRP and we leave it unquantified.

7 Conclusion

Best-of-N selection is widely used, but its winner is fragile: 41–47% of items change their selected answer under small scoring perturbations on a fixed pool. Our audit protocol (SRP) holds pools and compute constant while replaying controlled perturbations. Scoring against a fixed reference and averaging per token reduces the average flip-rate to 24.9%, and roughly 71% of what remains are near-ties that differ only in phrasing. Training-time filtering plays a complementary role, improving the factual quality of the candidate pool rather than selection stability.

The SRP stability gain transfers from LLaVA-7B to LLaVA-13B, while the hallucination-reduction gain transfers to Qwen2.5-VL-7B-Instruct. Both SRP and the anchored selector are modality-agnostic: the scored input can be an image–question pair or a text-only prompt, and the selector needs only token probabilities from a policy and a frozen reference, so any system that selects among cached candidates can be audited and stabilized the same way. The training-time filtering is largely transferable as well — length matching and prompt-copy control apply to any modality — with only the object-existence filter being vision-specific, replaceable by a task-appropriate factuality or grounding check. We will release the SRP audit scripts, pool manifests, and item-level outputs at `https://github.com/PengfeiZheng-721/DSPA`; reference checkpoints are available from the author upon reasonable request.

References

1. Aminian, G., et al.: Best-of-n through the smoothing lens: KL divergence and regret analysis. CoRR **abs/2507.05913** (2025)
2. Bai, Y., et al.: Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073 (2022)
3. Chen, M., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021)
4. Chuang, Y.S., Xie, Y., Luo, H., Kim, Y., Glass, J.R., He, P.: DoLa: Decoding by contrasting layers improves factuality in large language models. In: International Conference on Learning Representations (ICLR) (2024), arXiv:2309.03883
5. Dorner, F.E., et al.: ROC-n-reroll: How verifier imperfection affects test-time scaling. CoRR **abs/2507.12399** (2025)
6. Freitag, M., Grangier, D., Tan, Q., Liang, B.: High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. Transactions of the Association for Computational Linguistics (2022), arXiv:2111.09388
7. Fu, C., Chen, P., Shen, Y., Qin, Y., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. CoRR (2023)
8. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning (ICML) (2016)
9. Gao, L., Schulman, J., Hilton, J.: Scaling laws for reward model overoptimization. In: ICML (2023)

10. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
11. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML) (2017), arXiv:1706.04599
12. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: ICLR (2020)
13. Hu, E.J., Shen, Y., Wallis, P., et al.: Lora: Low-rank adaptation of large language models. arXiv:2106.09685 (2021)
14. Huang, A., et al.: Is best-of-n the best of them? Coverage, scaling, and optimality in inference-time alignment. In: International Conference on Machine Learning (ICML) (2025)
15. Huang, Q., Dong, X., Zhang, P., Wang, B., He, C., Wang, J., Lin, D., Zhang, W., Yu, N.: Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In: CVPR (2024)
16. Ichihara, Y., et al.: Evaluation of best-of-n sampling strategies for language model alignment. Transactions on Machine Learning Research (2025)
17. Jinnai, Y., et al.: Regularized best-of-n sampling to mitigate reward hacking for language model alignment. CoRR **abs/2404.01054** (2024)
18. Khalaf, H., et al.: Inference-time reward hacking in large language models. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
19. Koehn, P., Knowles, R.: Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation (WMT). pp. 28–39 (2017). https://doi.org/10.18653/v1/W17-3204
20. Leng, S., Zhang, H., Chen, G., Li, X., Lu, S., Miao, C., Bing, L.: Mitigating object hallucinations in large vision-language models through visual contrastive decoding. CoRR **abs/2311.16922** (2023)
21. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: Proceedings of NAACL-HLT (2016)
22. Li, X.L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., Lewis, M.: Contrastive decoding: Open-ended text generation as optimization. arXiv preprint arXiv:2210.15097 (2023), main conference long paper at ACL 2023
23. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.: Evaluating object hallucination in large vision-language models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2023)
24. Lightman, H., et al.: Let's verify step by step. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
25. Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N.A., Choi, Y.: DExperts: Decoding-time controlled text generation with experts and anti-experts. arXiv preprint arXiv:2105.03023 (2021), aCL 2021 camera-ready
26. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. CoRR **abs/2310.03744** (2023)
27. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (2024), blog/Project page
28. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: CVPR (2019)
29. Murray, K., Chiang, D.: Correcting length bias in neural machine translation. In: Proceedings of the Third Conference on Machine Translation (WMT). pp. 212–223 (2018). https://doi.org/10.18653/v1/W18-64022

30. Ouyang, L., et al.: Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022)
31. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
32. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: CVPR (2019)
33. Srivastava, P., et al.: Robust reward modeling via causal rubrics. In: International Conference on Machine Learning (ICML) (2025)
34. Team, Q.: Qwen2.5-VL technical report. CoRR **abs/2502.13923** (2025)
35. Verdun, C.M., et al.: Soft best-of-n sampling for model alignment. In: IEEE International Symposium on Information Theory (ISIT) (2025)
36. Wang, J., Wang, Y., Xu, G., Zhang, J., Gu, Y., Jia, H., Yan, M., Zhang, J., Sang, J.: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. CoRR **abs/2311.07397** (2023)
37. Wang, X., Wei, J., et al.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
38. Yang, A.X., et al.: Bayesian reward models for LLM alignment. CoRR **abs/2402.13210** (2024)
39. Yu, T., Yao, Y., Zhang, H., He, T., Han, Y., Cui, G., Hu, J., Liu, Z., Zheng, H., Sun, M., Chua, T.: Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In: CVPR (2024)
40. Yu, T., Zhang, H., Yao, Y., He, T., Hu, J., Liu, Z., Chua, T., et al.: Rlaif-v: Open-source ai feedback leads to super gpt-4v trustworthiness. CoRR **abs/2405.17220** (2024)
41. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. CoRR **abs/2311.16502** (2023)