Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA38553

Pricing Audits in Verifier-Guided Test-Time Scaling: Proxy Reliability and Budgeted Candidate Selection

Hanwen Zhang

College of Computer Science, Sichuan University, Chengdu, China
hanwen_zhang@stu.scu.edu.cn

Abstract. In verifier-guided test-time scaling, a cheap proxy ranks candidates and an audit obtains the target score of the selected output. The best split of a budget between generation and audits depends on proxy reliability among still-competitive candidates. We study this reliability through decision-granularity correlation: the proxy–target association within a prompt's top-ranked shortlist. We connect classical range restriction to an audit-efficiency threshold for linear proxy-assisted estimation, then evaluate candidate selection under a shared generation-and-audit budget. PickScore–HPS correlation falls from 0.84 within full diffusion prompt pools to 0.32 in their top 10%, and an API judge agrees with the target on 53.4% of near-ties. On text, an approximate correlation diagnostic accepts GSM8K and abstains on ARC. Across five diffusion models, we separate ranking gains from allocation gains. On 80 new Stable Diffusion XL prompts, reallocating a random-audit policy improves the composite preference score by 0.01196, while screening retains only 0.00076 over tuned random auditing (95% interval includes zero); higher audit prices favor screening.

Keywords: Test-Time scaling, Proxy reliability, Audit cost, Budget allocation, Candidate selection

1. Introduction

Test-time scaling improves language reasoning and diffusion generation by spending extra inference computation on sampling and verification [1, 2]. In verifier-guided selection, a generator proposes candidates, a cheap proxy ranks them, and an audit obtains the target score of the candidates it inspects. Under a fixed budget these operations compete: every additional candidate costs generation and proxy scoring, and every audit costs a target evaluation. How the budget should be divided therefore depends on how much the proxy ranking is worth.

That worth is decided among the strongest candidates, because selection acts only on the top of the proxy ranking. Proxy quality, however, is usually summarized by agreement over whole candidate pools. Selecting on the proxy restricts the range of the scores that remain and can weaken proxy–target association exactly where decisions are made, a pattern related to reward-model overoptimization [3] and to statistical range restriction [4]. A proxy that orders a broad pool reliably may thus barely separate its top candidates, so pool-level agreement can overstate the benefit of screening.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.BA38553

We study decision-granularity correlation, the proxy–target correlation within a prompt's top-ranked shortlist, and make three contributions. First, a Gaussian truncation analysis shows how shortlist correlation shrinks with the retained fraction, and we link it to the correlation required for an audit-efficiency gain in ideal linear proxy-assisted estimation, with a lower-bound diagnostic that accepts or abstains, applied to diffusion and text (GSM8K, ARC) pools. Second, we evaluate selection under a shared generation-and-audit budget against two random controls: one matches candidate and audit counts, isolating the gain from proxy ordering, and one optimizes its own allocation, measuring what survives reallocation. Third, we apply both analyses to five diffusion models and 80 held-out SDXL prompts with frozen policies, finding that most of the matched-count screening gain is recovered by reallocating random audits, while higher audit prices favor screening.

2. Related work

Test-time scaling. Snell et al. [1] show that allocating inference compute across sampling and verification can outperform scaling parameters for LLM reasoning, and Ma et al. [2] use verifiers to search over noises and candidates in diffusion models; we instead price audits separately and split a fixed budget between generating and auditing candidates.

Learned proxies and their reliability. Preference models such as ImageReward [5], PickScore [6] and HPSv2 [7] serve as text-to-image rewards and evaluators, and optimizing hard against a learned reward eventually degrades the true objective [3]. Reliability is usually reported over whole pools; we measure it within the decision region, with the target as an operational audit.

Range restriction and proxy-assisted estimation. Psychometrics corrects correlations for selection on a predictor [4], and prediction-powered inference [8] uses model predictions to reduce the variance of population estimates. Our linear certificate follows the same control-variate logic, but an estimation gain does not imply a better selected candidate. Because selecting and evaluating on the same data biases performance estimates [9], we separate development and held-out prompts.

3. Budgeted candidate selection

3.1. Candidates and audit costs

For each prompt p, a generator provides K candidates with proxy scores s and target scores y. A policy generates $n \le K$ candidates, audits $m \le n$, and returns the best audited candidate by target score (or the top-proxy candidate if $m = 0$). With generation, proxy and audit costs c_g, c_s and c_a, a budget B requires:

$$C(n,m) = n(c_g + c_s) + mc_a \le B \tag{1}$$

The policies are zero-audit selection Z, top-m auditing T and uniform random auditing R; a score cube of prompt–candidate–scorer values supports offline evaluation, revealing targets only for audited candidates (Figure 1).

3.2. Local reliability and random controls

Proxy and target means are subtracted within each retained prompt pool and the residuals pooled and correlated; the decision region is the top-q candidates under the proxy.

Let $Q_F(a)$ be the expected returned target score of family F at allocation a = (n,m), averaging over candidate sampling, audit randomization, then prompts, and let development prompts choose a

DOI: 10.54254/2755-2721/2026.BA38553

separate feasible allocation a_F^* per family. The matched-count and budget-tuned gains, and their difference, which isolates the random policy's allocation gain, are:

$$\Delta_{match} = Q_T(a_T^*) - Q_R(a_T^*) \tag{2}$$

$$\Delta_{tuned} = Q_T(a_T^*) - Q_R(a_R^*) \tag{3}$$

$$\Delta_{match} - \Delta_{tuned} = Q_R(a_R^*) - Q_R(a_T^*) \tag{4}$$

For m > 0, sampling n candidates and auditing m uniformly is equivalent to sampling m directly, so random expected quality depends only on m and unaudited candidates only add cost; more audits cannot lower the expected maximum, and at m = n the policies tie.

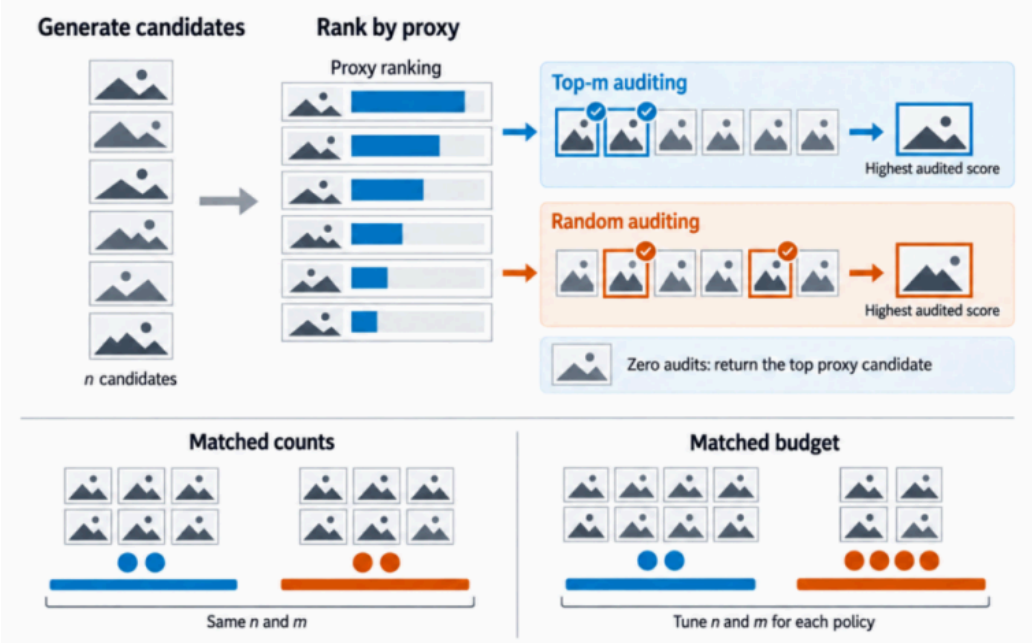

Figure 1. Candidate selection under an audit budget (schematic thumbnails)

4. Reliability and estimation efficiency

4.1. Gaussian shortlist correlation

Let (X,Y) be standard bivariate Gaussian with correlation ρ, $|\rho| < 1$, and retain $X > c_q = \Phi^{-1}(1 - q)$, with φ and Φ the standard normal density and distribution functions. Define the following quantities and the resulting retained correlation:

$$\lambda_q = \frac{\phi(c_q)}{q}, \nu_q = 1 + c_q\lambda_q - \lambda_q^2 \tag{5}$$

$$\rho_q = \frac{\rho\sqrt{\nu_q}}{\sqrt{\rho^2\nu_q + 1 - \rho^2}}, \rho_q \to 0 \; as \; q \to 0 \tag{6}$$

Equation (6) follows by writing $Y = \rho X + \sqrt{(1 - \rho^2)}\varepsilon$ and truncating X. On real data the top-10% correlation is 0.32 versus a Gaussian prediction of 0.54, so the observed decline is even steeper.

DOI: 10.54254/2755-2721/2026.BA38553

4.2. Scope of the correlation-based certificate

For N independent audited units with finite positive variances and known μ_X = E[X], estimate μ_Y = E[Y] with a coefficient fixed independently of the units:

$$\widehat{\mu}_{\beta} = \overline{Y} - \beta(\overline{X} - \mu_X) \tag{7}$$

For r = Corr(X,Y), the population-optimal β* = Cov(X,Y)/Var(X) yields:

$$Var(\widehat{\mu}_{\beta^*}) = \frac{\sigma_Y^2}{N}(1 - r^2) \tag{8}$$

The ideal variance ratio is $1/(1 - r^2)$; a ratio above $F_{min} > 1$ requires the following (for a positively oriented proxy, ratios of 1.5 and 2 need r > 0.577 and r > 0.707):

$$r^2 > 1 - \frac{1}{F_{min}} \tag{9}$$

A one-sided lower bound L_r with coverage at least $1 - \alpha$ supports acceptance when:

$$(max\{L_r, 0\})^2 > 1 - \frac{1}{F_{min}} \tag{10}$$

If $r^2 \leq 1 - 1/F_{min}$, acceptance requires a coverage failure (probability at most α under valid coverage); Fisher-based calls below are approximate. The ratio ignores fitting and calibration costs, and since the diagnostic and the (14,9) policy of Section 6 concern different regions, no predictive link to selection gains is claimed.

5. Allocation procedure and experimental design

5.1. Development and held-out evaluation

Development prompts fix normalization and each family's best feasible allocation; held-out comparisons use shared candidate prefixes and paired prompt differences.

The large-pool dataset, Stack, has 16 prompts × 288 SDXL candidates, with PickScore, ImageReward and CLIPScore [10] as proxies and HPS as target; diagnostics retain the top 10% (top 25% in 24-candidate pools). Text targets are answer correctness on GSM8K (80 questions × 24 Qwen2.5 solutions) [11, 12] and ARC-Challenge (60 questions) [13], with an LM judge as proxy.

Image judges (local Qwen3-VL-8B-Instruct; API Qwen3.6-Plus) compare ImageReward top-eight shortlists in both orders, scoring inconsistent pairs 0.5; near-ties have HPS gaps below 0.05.

5.2. Budget-allocation study

Five models (TinySD, BK-SDM-small, SDXL-base-1.0, SD 1.5, DreamShaper-7) share 40 prompts × 24 candidates [14-16], generated with DPM-Solver++ (Karras, 30 steps, guidance 7) [17, 18] at 512^2 (SDXL: 1024^2); ImageReward-v1.0 is the proxy, and the target averages normalized PickScore and HPSv2 (v2.1):

$$y_i = \frac{1}{2}\sum_{k\in\{Pick,HPS\}} \frac{r_{i,k} - a_{D,k}}{b_{D,k}} \tag{11}$$

DOI: 10.54254/2755-2721/2026.BA38553

Here $a_{D,k}$ and $b_{D,k}$ are scorer k's minimum and positive range on same-model development prompts D, $r_{i,k}$ is a raw score; test scores are not clipped.

Nominal costs are C(n,m;λ) = 31n + λm (30 generation and one proxy unit per image, λ per audit), with B ∈ {256,512,1024}, λ ∈ {1,8,32,128} and n ≤ 24; the coarse menu uses m ∈ {0,2,4,8,12,24}, m ≤ n, and the integer menu every feasible m.

Five-model evaluation is leave-one-prompt-out (integer menu as sensitivity analysis). The separate test freezes policies and normalization on 40 development prompts and generates 1,920 new SDXL images for 80 GenEval prompts [19]; the prespecified primary comparison is Equation (4) at B = 512, λ = 8. Results average 512 permutations over three seeds, with paired 95% prompt-bootstrap intervals.

6. Results

6.1. Reliability within competitive regions

In Stack, within-prompt PickScore–HPS correlation falls from 0.84 to 0.32 in the top 10%, ImageReward from 0.59 to 0.15, and CLIPScore from 0.42 to −0.02 (Figure 2).

Text tasks contrast: GSM8K has decision-region correlation 0.854 (approximate lower bound 0.79), exceeding both thresholds, whereas ARC's 0.59 has lower bound 0.43, implying abstention.

Near-tie agreement with HPS is 53.4% [50.8%,56.3%] for the API judge (915 of 933 pairs) and 52.0% [50%,54%] for the local judge, without establishing judge equivalence.

6.2. Ranking gains and budget allocation

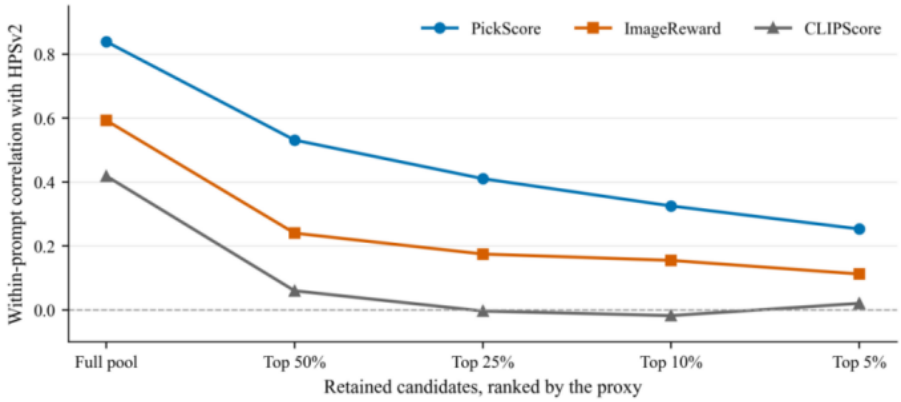

Figure 2. Proxy–target correlation across retained fractions (stack; target HPSv2)

At B = 512, λ = 8, coarse-menu screening beats zero-audit selection by 0.039–0.064 but tuned random by only 0.0006–0.0023; TinySD and SDXL audit 12 of 13, leaving little room for ranking.

Under the integer menu, four models select (n,m) = (14,9) and tuned random (13,13), costing 506 and 507 units, while TinySD fully audits under both and ties. Matched-count intervals are positive, but tuned-family intervals include zero (Figure 3a): screening buys one extra candidate, whereas tuned random fully audits a slightly smaller pool.

DOI: 10.54254/2755-2721/2026.BA38553

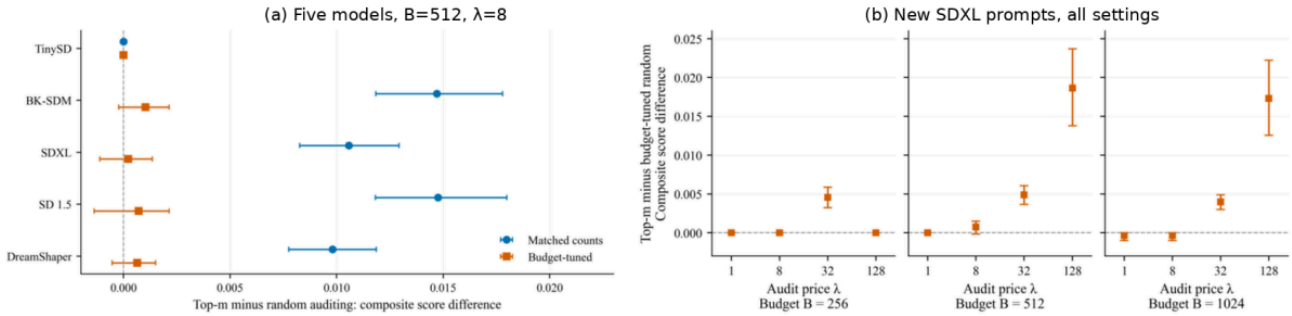

Figure 3. Screening gains over random controls: (a) five models; (b) 80 new SDXL prompts

6.3. Allocation gains on new prompts

With frozen allocations (14,9) for screening and (13,13) for tuned random, the primary allocation gain is 0.01196 (Table 1); screening gains 0.01272 over matched random but only 0.00076 over tuned random, whose interval includes zero, so neither superiority nor equivalence is established.

Table 1. Frozen-policy SDXL results on 80 new prompts (B = 512, λ = 8)

Comparison	Difference	Paired 95% interval
Tuned random − matched random	0.01196	[0.01064, 0.01340]
Top-m − matched random	0.01272	[0.01090, 0.01458]
Top-m − tuned random	0.00076	[−0.00015, 0.00151]
Top-m − zero audit	0.05765	[0.04753, 0.06825]

At B = 512, screening beats tuned random by 0.00489 at λ = 32 and 0.01864 at λ = 128, with positive secondary intervals (Figure 3b). At B = 1024 and λ ≤ 8, full-pool auditing is affordable, so negative differences there cannot be attributed to screening alone.

6.4. Routing, calibration and generator capability

Routers for prompt-level audit value overfit (negative cross-validated R^2, in-sample 0.19–0.68) with no reliable gain over uniform auditing. Mixing six generator configurations lowers proxy selection by 0.028, a gap that configuration-specific calibration reduces. On a fixed-512px ladder, decision correlation and target spread both correlate at −0.94 (Spearman) with raw-HPS capability.

7. Discussion and conclusion

Matched-count and separately optimized controls measure proxy-ordering gains and the advantage left under a shared budget, which local correlation and the certificate cannot replace. Targets are learned, prompt pools are small, the held-out test covers one SDXL setup, and high budgets saturate pools. Under nominal costs, fully auditing a smaller pool can rival screening, whereas expensive audits favor it; broader generators, measured costs and human assessment remain to be tested.

References

[1] Snell, C., Lee, J., Xu, K., & Kumar, A. (2025). Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. *International Conference on Learning Representations*.

DOI: 10.54254/2755-2721/2026.BA38553

[2] Ma, N., Tong, S., Jia, H., Hu, H., Su, Y.-C., Zhang, M., et al. (2025). Scaling inference time compute for diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2523–2534.
[3] Gao, L., Schulman, J., & Hilton, J. (2023). Scaling laws for reward model overoptimization. *Proceedings of the 40th International Conference on Machine Learning*, 202, 10835–10866.
[4] Carretta, T. R., & Ree, M. J. (2022). Correction for range restriction: Lessons from 20 research scenarios. *Military Psychology, 34*, 551–569. https://doi.org/10.1080/08995605.2021.2022067
[5] Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., et al. (2023). ImageReward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems, 36*, 15903–15935.
[6] Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., & Levy, O. (2023). Pick-a-Pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems, 36*, 36652–36663.
[7] Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., & Li, H. (2023). Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. https://arxiv.org/abs/2306.09341
[8] Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I., & Zrnic, T. (2023). Prediction-powered inference. *Science, 382*, 669–674. https://doi.org/10.1126/science.adi6000
[9] Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11*, 2079–2107.
[10] Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., & Choi, Y. (2021). CLIPScore: A reference-free evaluation metric for image captioning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7514–7528. https://doi.org/10.18653/v1/2021.emnlp-main.595
[11] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., et al. (2021). Training verifiers to solve math word problems. *arXiv.org*. https://arxiv.org/abs/2110.14168.
[12] Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., et al. (2024). Qwen2.5 technical report. *arXiv.org*. https://doi.org/10.48550/arXiv.2412.15115
[13] Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv.org*. https://arxiv.org/abs/1803.05457.
[14] Kim, B.-K., Song, H.-K., Castells, T., & Choi, S. (2024). BK-SDM: A lightweight, fast, and cheap version of Stable Diffusion. *Computer Vision – ECCV 2024*, 381–399.
[15] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., et al. (2024). SDXL: Improving latent diffusion models for high-resolution image synthesis. *International Conference on Learning Representations*.
[16] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.
[17] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2025). DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research, 22*, 730–751. doi: 10.1007/s11633-025-1562-4.
[18] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems, 35*, 26565–26577.
[19] Ghosh, D., Hajishirzi, H., & Schmidt, L. (2023). GenEval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems, 36*, 52132–52152.