FROM SET DIVERSITY TO OUTPUT DIVERSITY: PROGRESSIVE SELECTION FOR INFERENCE-TIME ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time alignment often samples several candidates and prunes them progressively: a reward model scores a preview of each candidate partway through generation, and only the best few, the survivors, are finished. Such selection is known to reduce diversity. Existing diversity methods for selection make the kept set diverse, but when a user receives one sample per request, what matters is output diversity, not set diversity. We characterise how the two relate and find that one candidate, the *champion*, whose preview scores best, largely decides whether a more diverse set changes what the user receives. Selectors that keep the champion turn only 6% to 22% of their gain in set diversity into output diversity, compared with 26% to 66% for selectors free to drop it. This *champion constraint* binds on 49% to 72% of requests, a share we prove rises with preview quality and falls with pool size. We propose champion-free selection and instantiate it with exact max-sum diversification and with density-aware selection, a single sort that penalises crowding in the candidate pool. On four text-to-image models, both selectors raise output diversity over reward-only pruning at the same budget while keeping 86% to 96% of its reward gain; requiring them to keep the champion removes more than half of that increase. Density also delivers rare modes 7% to 14% more often than max-sum, as our analysis predicts, and blinded raters prefer its variety in 72.5% of comparisons on SD3.5 and 70.3% on FLUX.2 Klein.

Code: `https://anonymous.4open.science/r/progressive-selection-B85A`

1 INTRODUCTION

Inference-time alignment improves a frozen generator without retraining it: a reward model scores candidates and only the ones it prefers are kept, through best-of-N, progressive pruning (Guimarães & Perona, 2026) or resampling of intermediate states (Wu et al., 2023; Singhal et al., 2025; Li et al., 2025b;a). We study progressive pruning. A request starts N candidate trajectories from different seeds and, partway through, decodes each into a *preview*, the model's current estimate of its final image. The reward model scores the previews, only the k best candidates (the *survivors*) are finished, and the request delivers the survivor whose finished image has the highest *final reward*. The same pattern drives best-of-N decoding with early rejection and process-reward search in language models (Sun et al., 2024; Snell et al., 2025), test-time scaling of video generators (Liu et al., 2025) and reward-guided design of molecules and biological sequences (Li et al., 2025b). In each case a request returns one output, so the diversity a user experiences is how that output varies across requests, and selecting by reward reduces it. Reward-only pruning, which keeps the best previews, lowers it on all four generators (backbones) we study: on SD3.5, eight requests yield 2.75 distinct content modes (clusters of similar images) instead of 3.13 without pruning.

Pruning already keeps a set of survivors, so a natural remedy is to make that set diverse. Maximal marginal relevance (MMR), max-sum diversification and determinantal point processes (DPPs) choose a subset that trades quality against spread (Carbonell & Goldstein, 1998; Gollapudi & Sharma, 2009; Kulesza & Taskar, 2012), and group inference and joint sampling diversify a jointly produced set of images (Parmar et al., 2026; Corso et al., 2024). This *set diversity*, the spread among the kept candidates, is what users see when all are returned. Here only one survivor is returned, and

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

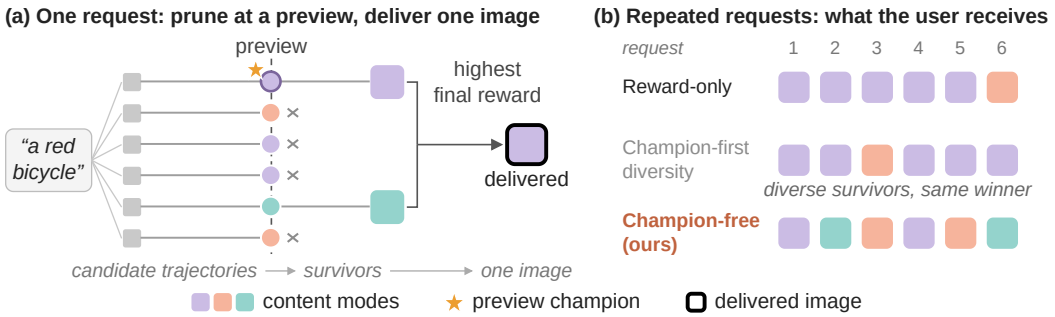

Figure 1: Set diversity does not guarantee output diversity. (a) Each request prunes its candidates at a preview and returns only the survivor with the highest final reward. (b) Across requests, reward-only pruning keeps returning the same kind of image. Adding diverse companions to the preview champion (★), the candidate with the best preview, changes little: it still wins whenever it is also the final best. A selector that may drop it changes the delivered image.

what matters is *output diversity*, the variation of that one image across requests. To our knowledge, set-diversity objectives have not been used to choose the survivors of reward-guided pruning for single-image delivery. Yet pruning is an attractive place to diversify: it reuses candidates already paid for and needs no gradients or sampler changes, so it suits few-step and autoregressive generators, which leave few or no denoising steps to steer. The usual rules, however, always keep the best-scoring candidate, so diverse survivors can still produce the same winner (Figure 1).

We identify the reason, which we call the *champion constraint*. Call the candidate with the highest preview reward the *preview champion*. Common diversification rules start from it and add diverse companions, so they never remove it; we call them *champion-first*. On many requests, however, the preview champion also has the highest final reward. On such a request, which we call *locked*, the champion beats any companion in the final comparison, so the delivered image is fixed however strongly the rule favours diversity and whatever its objective, solver or features. Better previews lock more requests; with a perfect preview, every request is locked. The constraint yields a design principle for any pipeline that prunes on partial outputs and returns one output, including language-model search and agent loops: a diversity-aware selector must be free to drop the champion.

We build on this principle with *champion-free diversity-aware progressive selection*: the selector scores candidates by preview reward and by the geometry of the pool in a feature space, and never reserves a slot for the champion. Exact max-sum searches all k-subsets for the best trade-off between reward and spread. Density-aware selection (*density* for short) scores each candidate by its preview reward minus the log of how many close neighbours it has in the whole pool, and keeps the top scores. Because crowding counts the whole pool, every rare candidate can outrank the champion, and rare candidates can fill all k slots. In a simple model where rare content has frequency p and scores slightly below common content, reward-only pruning and max-sum deliver it only when all N candidates are rare (probability p^N), whereas density delivers it whenever k are (order p^k).

- We identify the champion constraint, which decides whether diversification reaches the delivered image. It holds for any objective, solver, feature space and generator and binds on 49% to 72% of main-benchmark requests; we prove that this share rises with preview quality and falls with pool size, and a parameter-free model predicts it.
- We derive the design principle that a diversity-aware selector must be free to drop the champion, and instantiate it with exact max-sum, bringing set-diversity selection into single-output inference-time alignment. Requiring the champion, with everything else fixed, removes more than half of the gain in output diversity for every selector and backbone.
- We propose density-aware selection, a single sort over a pool-level crowding score with provable access to rare modes, which adds at most 2.2% latency. On four backbones it raises output diversity at a fixed budget over reward-only pruning, best-of-N, Feynman–Kac steering, champion-first rules and k-DPP sampling. It matches exact max-sum on mean diversity while delivering rare modes more often, and blinded raters see the added variety on SD3.5 and Klein.

2 RELATED WORK

Inference-time alignment. Best-of-N sampling and progressive pruning select among independent samples of a frozen generator (Guimarães & Perona, 2026). Sequential Monte Carlo, Feynman–Kac steering, SVDD and DSearch instead resample or search intermediate states under a reward (Wu et al., 2023; Singhal et al., 2025; Li et al., 2025b;a), and verifier-guided search over initial noises scales inference compute beyond denoising steps (Ma et al., 2025). Reward over-optimisation and the resulting loss of diversity motivate corrections in training and at inference (Gao et al., 2023; Zhang et al., 2024; Kirk et al., 2024; Jena et al., 2025; Sinha et al., 2026), including VASR, which counteracts the loss of particle lineages during resampling (Shekhar et al., 2026). We address a different stage of the pipeline: which candidates survive a pruning cut when a single image is delivered.

Set diversification. Maximal marginal relevance, max-sum diversification and determinantal point processes select subsets that balance quality and spread (Carbonell & Goldstein, 1998; Gollapudi & Sharma, 2009; Kulesza & Taskar, 2012), and diverse beam search diversifies language-model decoding (Vijayakumar et al., 2018). Scaling group inference (SGI) brings such objectives to text-to-image generation to return a diverse *group* of images for one request (Parmar et al., 2026).

Quality-diversity search. Novelty search rewards solutions unlike an archive of past ones (Lehman & Stanley, 2011a), and novelty search with local competition and MAP-Elites keep the best solution in each niche of a behaviour space (Lehman & Stanley, 2011b; Mouret & Clune, 2015; Pugh et al., 2016). These methods score novelty against an archive accumulated over generations; density scores crowding once, against the current pool, with no archive. In that literature, favouring the current best is known to limit exploration (Lehman & Stanley, 2011a); we show that with a single delivered output, keeping the best freezes the output on every locked request, and we predict and measure how often requests are locked.

Diversity-aware sampling. Particle Guidance, EDDY and CADS change the sampling dynamics (Corso et al., 2024; Vinograd et al., 2026; Sadat et al., 2024), and DOG-Trim repels parallel trajectories before reward-only pruning (Yu et al., 2025); SAIL, DivIn and contrastive noise optimization change the initial noise (Jeon et al., 2025; Li et al., 2026; Kim et al., 2026). These methods act before or during generation and are compatible with ours, which changes only which candidates are allowed to finish; Section 6.2 combines density with CADS.

3 SET DIVERSITY AND OUTPUT DIVERSITY

Progressive selection. For a prompt c, one request draws N independent candidate trajectories from the frozen generator. At a checkpoint partway through, the reward model scores each candidate's *preview*, the model's current estimate of the final image (for diffusion, the predicted clean image), and a feature encoder embeds it. A selector keeps k of the N candidates; the survivors finish their own trajectories with the unmodified sampler, and the request returns the survivor with the highest final reward. Like best-of-N and reward-only pruning, we keep this rule: any softer rule pays to finish every survivor and then gives up reward (Appendix A.6). *Native* sampling, without selection, is our $1\times$ reference. Reward-only pruning keeps the k candidates with the highest preview reward. Cost counts denoising steps over all candidates: with the checkpoint at the midpoint, keeping 2 of 6 costs $6 \times \frac{1}{2} + 2 \times \frac{1}{2} = 4$ full trajectories, as best-of-4 does; we write it $6 \to 2$.

Output diversity. For a *policy* A, a generator together with its selection rule, let Y be the image it delivers for prompt c and Y' the image it delivers for an independent second request. We define the output diversity of A as the expected perceptual distance between the two,

$$D(A) = \mathbb{E}_c \, \mathbb{E}_{Y,Y' \overset{\text{iid}}{\sim} P_A(\cdot|c)} \big[d(Y, Y') \big], \tag{1}$$

with d the DreamSim distance, a learned perceptual distance (Fu et al., 2023). We average d over the 28 pairs of eight requests per prompt. Quality is the mean reward of the delivered images.

Only the winner matters. Output diversity is not set diversity, the diversity of the survivors. To see why, note that DreamSim is a cosine distance between unit embeddings; let z_i be those of the

completed survivors. To cover any final rule, let w_i be the probability that the rule delivers survivor i, with $m = \sum_i w_i z_i$ the weighted mean and $W = \sum_{i,j} w_i w_j (1 - z_i^\top z_j)$ the weighted spread. The total-variance identity splits the output diversity $D_c(A)$ of a single prompt c into two terms,

$$D_c(A) = \underbrace{\mathbb{E}\, W}_{\text{within a request}} + \underbrace{\mathbb{E}\, \|m - \mathbb{E}m\|^2}_{\text{across requests}}. \tag{2}$$

When the final rule returns the survivor with the highest reward, one survivor has $w_i = 1$ and the first term vanishes. All output diversity then comes from how the winner varies across requests. Diverse survivors help only if they change which image wins, and a selector should be judged by the images it delivers rather than by the spread of the set it keeps. We call the share of a selector's gain in set diversity that reaches output diversity its *conversion*.

4 When diversification fails: the champion constraint

Consider one request with a pool of N candidates at the checkpoint. Candidate i has preview reward r_i and final reward R_i, the reward it would reach if finished. We scale the preview reward as $v_i = r_i/\sigma_r$, where the fixed scale σ_r is the typical within-request spread of preview rewards on development prompts. The *preview champion* is $a = \arg\max_i v_i$ and the *final winner* is $a^\star = \arg\max_i R_i$. A selector keeps a set S of k candidates, and the request delivers $Y(S) = \arg\max_{i \in S} R_i$. We call a selector *champion-preserving* if $a \in S$ on every request, and we call a request *locked* if $a = a^\star$.

Lemma 1 (Champion constraint)**.** *On a locked request, every champion-preserving selector delivers the preview champion, whatever its other survivors. Hence two champion-preserving selectors keeping S_1 and S_2 from one pool deliver different images only on unlocked requests:* $\Pr\{Y(S_1) \neq Y(S_2)\} \leq \Pr\{a \neq a^\star\}$.

On a locked request (Figure 2a) the kept set contains the pool's final winner, so that candidate is delivered. The statement makes no assumption about the diversity objective, its solver, the features or the strength of the diversity term. Its force comes from the *locked share* $L = \Pr\{a = a^\star\}$, the fraction of requests that are locked, which is 49% to 72% across our backbones on the main benchmark. Reward-only pruning keeps the champion too, so on all of those requests every champion-preserving rule delivers exactly what reward-only pruning delivers, however much diversity it adds to the survivors. In a Gaussian model the locked share depends only on preview quality and pool size.

Proposition 1 (Locked-share law)**.** *Let the preview–final pairs (v_i, R_i) be i.i.d. Gaussian with correlation $\rho \in [0, 1)$. The locked share $L(N, \rho)$ does not depend on the means and scales of v and R, increases strictly in ρ, decreases strictly in N, equals $1/N$ at $\rho = 0$ and tends to 1 as $\rho \to 1$.*

Better previews therefore make the constraint bind more often, and a perfect preview makes it bind on every request (Section 6.3). A larger pool lowers the share only slowly: for ρ between 0.6 and 0.7, each doubling of N from 8 to 128 multiplies L by 0.77 to 0.84. Appendix A.2 gives the proof and its scope. Dropping the champion helps density more in larger pools (Appendix G).

Which rules keep the champion? Maximal marginal relevance and greedy max-sum start from the preview champion and add diverse companions; these champion-first rules are champion-preserving by construction. So is any rule that reserves a slot for the best preview, whatever its objective or solver. Exact max-sum, DPP sampling and density (Section 5) can drop the champion.

Two survivors isolate the constraint. Write the max-sum objective as

$$J(S) = \frac{1}{k} \sum_{i \in S} v_i + \lambda \frac{1}{\binom{k}{2}} \sum_{i < j \in S} \frac{d_{ij}}{\bar{d}}, \qquad |S| = k, \tag{3}$$

where d_{ij} is the distance between the previews' features (Section 5), $\bar{d}$ a fixed typical distance and λ the diversity strength. J is the survivors' mean scaled reward plus λ times their mean pairwise distance in units of $\bar{d}$; greedy max-sum starts from a and repeatedly adds the candidate that most raises J. For $k = 2$, champion-first greedy max-sum returns the maximiser of J over all pairs that contain a (Appendix A.1). Comparing greedy max-sum with exact max-sum at $k = 2$ therefore changes only one thing, the requirement to keep the champion, without any search approximation. Section 6.3 uses this comparison and the same requirement imposed on the density score.

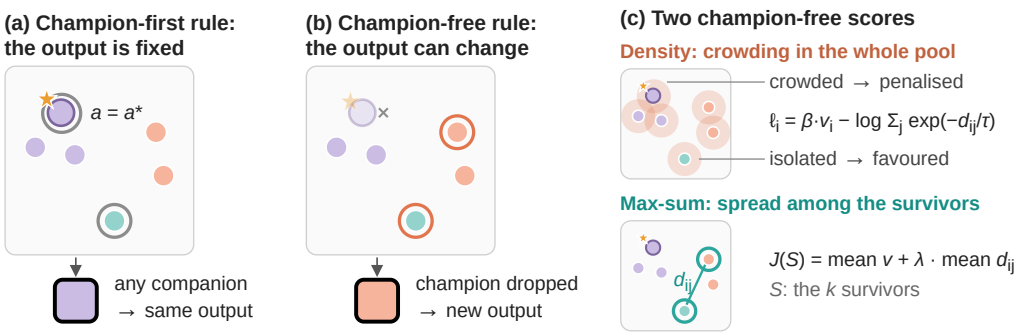

Figure 2: The champion constraint and the two champion-free selectors. Dots: candidates in feature space; colours: content modes. (a) When the preview champion (★) also wins on final reward, any selector that keeps it delivers it. (b) A selector that may drop it can deliver a different mode. (c) Density penalises crowding in the whole pool; max-sum rewards spread among survivors.

5 Champion-free diversity-aware selection

Following Section 4, we score candidates by reward and by the geometry of the pool and never reserve a slot for the champion. We use two scores. Both use the scaled preview reward $v_i = r_i/\sigma_r$ of Section 4, with r_i the preview ImageReward, and unit DINOv2 features of the previews (Oquab et al., 2024), with cosine distances d_{ij}.

Exact max-sum. It maximises $J(S)$ of Eq. (3) over all $\binom{N}{k}$ subsets (15 for $6 \to 2$) and, unlike greedy max-sum, can drop the champion when that trades a little reward for much more spread.

Density-aware selection. Density scores each candidate on its own, combining its promise with how crowded its neighbourhood is in the pool C of all N candidates,

$$\ell_i = \beta\, v_i - \log n_i, \qquad n_i = \sum_{j \in C} \exp(-d_{ij}/\tau), \tag{4}$$

and keeps the k candidates with the largest ℓ_i. The crowding n_i counts close neighbours within kernel width τ, including candidates about to be discarded, so it describes the pool, not the kept set. A candidate in a crowded mode is penalised even if it is the preview champion, and a candidate from a sparse mode is favoured. The reward weight β sets the trade-off: small β favours diversity, and $\beta \to \infty$ recovers reward-only pruning. The score has the form of fitness sharing (Goldberg & Richardson, 1987); what is new is its role as a champion-free selector for single-output delivery.

Why pool-level crowding helps rare modes. Max-sum rewards spread *among the survivors*, which one distant candidate next to a strong one already supplies; the strong one then wins the final comparison. Density instead lowers the score of every candidate in a crowded mode, so every rare candidate can outrank the champion and rare candidates can fill every slot. A model with one common mode and rare content makes this precise.

Proposition 2 (Rare-content access, informal)**.** *Let each of the N candidates independently be rare with probability p, in one rare mode or several, and otherwise come from one common mode. Candidates in one mode share their features and rewards, candidates in different modes are equally far apart with kernel similarity $b \in [0, 1)$ in Eq. (4), previews rank the modes as final rewards do, and the common mode's preview reward exceeds each rare mode's by at most $\Delta > 0$. Reward-only pruning and exact max-sum deliver a rare image with probability p^N. If $\Delta < (\sigma_r/\beta) \log \frac{N-k+bk}{k+b(N-k)}$, density delivers one with probability $\binom{N}{k}p^k + O(p^{k+1})$ as $p \to 0$.*

The condition says that the crowding penalty on the common mode outweighs its reward lead. Appendix A.5 gives the proofs. A block kernel ($b = 0$) gives the margin $\log \frac{N-k}{k}$, which needs $N > 2k$; for rare candidates from different modes, $\Delta < (\sigma_r/\beta) \log \frac{N-k+bk}{1+b(N-1)}$ suffices. The proposition locates density's advantage in requests holding at least k rare candidates, which density turns into k rare survivors; Section 6.4 finds most of the advantage there.

Table 1: Champion-free selection raises output diversity at a fixed budget. IR: ImageReward; DS: output diversity (Eq. 1); ΔDS: DS minus reward-only pruning ($\times 10^3$), grey where its paired 95% interval covers zero (intervals: Appendix D.1); main benchmark, four-trajectory budget. Our selectors at development strengths, greedy MMR and k-DPP as close to density's ImageReward as they reach.

	Klein (4-step)			SD3.5			FLUX.1-dev			Infinity (AR)		
Method	IR↑	DS↑	ΔDS	IR↑	DS↑	ΔDS	IR↑	DS↑	ΔDS	IR↑	DS↑	ΔDS
Native sampling (1×)	1.206	0.180	+12.9	1.161	0.249	+26.5	1.158	0.229	+20.1	1.090	0.127	+6.6
Best-of-4	1.405	0.169	+1.5	1.414	0.232	+9.1	1.410	0.214	+5.1	1.302	0.122	+1.2
Reward-only pruning	1.412	0.167	—	1.433	0.223	—	1.441	0.209	—	1.338	0.121	—
Greedy MMR	1.387	0.172	+4.9	1.415	0.230	+6.7	1.429	0.215	+5.6	1.331	0.121	+0.4
k-DPP	1.383	0.172	+5.3	1.406	0.230	+6.9	1.429	0.213	+4.4	1.324	0.122	+1.6
Max-sum (ours)†	1.387	0.178	+11.3	1.421	0.232	+9.2	1.429	0.220	+11.4	1.310	0.125	+4.9
Density (ours)†	1.383	0.180	+12.5	1.406	0.241	+18.4	1.429	0.219	+10.1	1.324	0.124	+3.6

† On all four backbones, IR within human preference tolerance of reward-only pruning (< 0.050 IR; Appendix B.4).

Cost. Previews are already decoded and scored for reward-only pruning. Density adds one feature-encoder pass and $O(N^2)$ arithmetic, raising measured latency by 2.2% on FLUX.2 Klein and 0.7% on SD3.5; selection itself takes at most 0.13 ms per request for pools of up to 16 candidates. Its cost does not depend on k, whereas exact max-sum enumerates all $\binom{N}{k}$ subsets.

6 Experiments

6.1 Setup

Models and budget. We use three flow-matching diffusion models, the four-step distilled FLUX.2 Klein 4B (Klein; Black Forest Labs, 2025; 2026), SD3.5 Medium (Esser et al., 2024) and FLUX.1-dev (Black Forest Labs, 2024), at 512×512, and the bitwise autoregressive (AR) Infinity-2B (Han et al., 2025), which generates an image coarse to fine over 13 scales. The diffusion models prune $6 \to 2$ at the midpoint, which costs four full trajectories; Infinity keeps 3 of 6 after 11 of its 13 scales, costing 3.93. ImageReward (Xu et al., 2023) scores previews and chooses the delivered image; PickScore (reported as PickScore/26) and CLIP (Kirstain et al., 2023; Radford et al., 2021) are further judges.

Diversity measures. Our main measure is output diversity, Eq. (1), over eight independent requests per prompt. For content modes, the 64 native-sampling images of each prompt form eight fixed DreamSim clusters; *distinct modes* counts the clusters among the eight delivered images, and the *rare-mode share* is the fraction of delivered images in clusters with at most 8 of the 64.

Prompt banks and statistics. The main benchmark has 200 PartiPrompts (Yu et al., 2022) on all four backbones. A second bank of 200 disjoint PartiPrompts, on Klein and SD3.5, repeats the champion interventions and the mode comparisons. We measure quality loss on a human scale: on held-out ImageRewardDB pairs, people prefer the higher-scoring image only 51% of the time when the gap is 0.050 ImageReward, 0.0023 PickScore/26 or 0.0044 CLIP; in our analysis, telling such a preference from chance would take about 20,000 paired judgements. We call this gap the judge's *tolerance* (Appendix B.4). One rule chose the strengths β and λ per backbone on separate development prompts, keeping each judge's estimated loss below 70% of its tolerance; these are the *development strengths*. Intervals are 95% paired prompt bootstraps, Bonferroni-corrected where several comparisons are tested jointly. Equal-quality comparisons mix two neighbouring strengths per request to match a target mean ImageReward and evaluate the mixture exactly (Appendix C.3).

6.2 Champion-free selection raises output diversity

In Table 1, reward-only pruning raises ImageReward over native sampling but lowers output diversity on every backbone. Both champion-free selectors recover between a third and nearly all of that

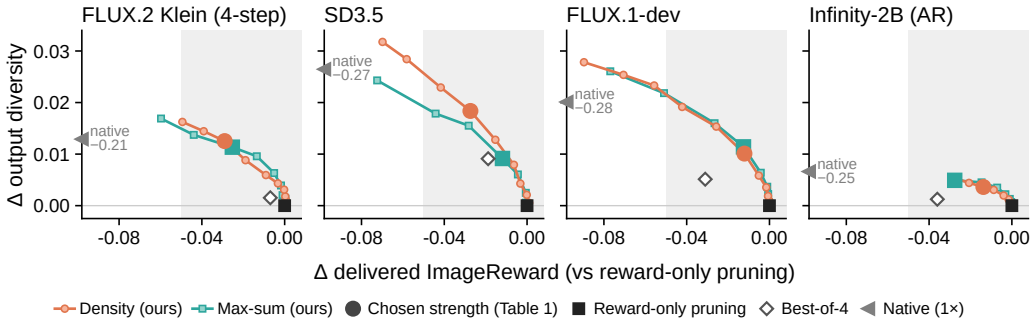

Figure 3: On every backbone our selectors beat best-of-4 in output diversity at equal or higher reward. Differences from reward-only pruning (origin); curves: each selector's strengths; large markers: development strengths; arrows: native sampling, off scale, labelled with its ImageReward difference. Shading: within the 0.050 ImageReward tolerance of pruning (Appendix B.4). PickScore: Appendix D.2.

loss at the same budget, with intervals excluding zero, and keep most of the reward gain. They give up at most 0.029 ImageReward, within the 0.050 tolerance (Appendix B.4). Because ImageReward also selects, we check two judges that do not: PickScore and CLIP lose at most 0.002 and 0.001, within their tolerances. Only the corrected one-sided bound on density's PickScore loss exceeds its tolerance, at 128% and 103% on SD3.5 and FLUX.1-dev, and even at that bound people would prefer pruning's image in at most 51.3% of pairs. With PickScore selecting instead, both selectors again raise output diversity and distinct modes on all three diffusion backbones, while ImageReward, now an independent judge, falls by at most 0.032 (Appendix D.2).

Figure 3 shows the full trade-offs. On every backbone both curves pass above best-of-4, with more output diversity at equal or higher reward. Selecting at the preview matters: applied to best-of-4's finished images, density delivers rare modes less often at matched reward (Appendix E.4).

The classical set-diversity selectors recover less. Greedy MMR keeps the champion by construction and cannot give up much reward: on three backbones even its most diverse setting keeps more reward than density. Against that setting, or at matched reward on FLUX.1-dev, density delivers more DreamSim and more distinct modes on all four backbones. The k-DPP, which samples survivor sets favouring high reward and large feature volume, drops the champion only by chance and also lies below density at matched reward on all four. All sixteen paired intervals (two baselines, two measures, four backbones) exclude zero. Four-particle Feynman–Kac steering at the same budget also trails density: it has less output diversity on all four backbones and lower reward on three.

In content modes, reward-only pruning covers fewer distinct modes than native sampling on every backbone. Density brings them back to about the native level or above, with its gain over pruning excluding zero on all four backbones (Figure 5 shows one prompt). These trends replicate on two further prompt suites, DIMCIM and general prompts (Appendix D.3).

CADS (Sadat et al., 2024) instead diversifies the candidates by noising the prompt conditioning early in sampling, and it combines with density: on SD3.5, density adds DreamSim on CADS candidates at matched reward. Pruning CADS candidates alone delivers more DreamSim than density on native candidates at the same reward, but cannot reach the reward of pruning native candidates (Appendix E.5).

6.3 Dropping the champion is what matters

To test whether dropping the champion is what matters, we require each selector to keep it. Required density keeps the champion and the $k-1$ best remaining scores; required max-sum optimises Eq. (3) over subsets that contain the champion, which for two survivors is exactly greedy max-sum (Section 4). Candidates, previews, strengths and final rule stay the same. Required density involves no greedy step or subset search, so its loss is the cost of the champion alone.

Table 2: Requiring the champion cuts conversion, the output-diversity gain over the set-diversity gain (%) at density's development ImageReward, main benchmark. Free/required: may drop / must keep the champion. L: locked share (%). *Cannot reach density's ImageReward; shown at its most diverse strength.

		Density		Max-sum	
Backbone	L	free	required	free	required
Klein (4-step)	49	66	22*	33	16*
SD3.5	51	61	17*	28	12*
FLUX.1-dev	60	58	15	31	12
Infinity (AR)	72	44	8*	26	6*

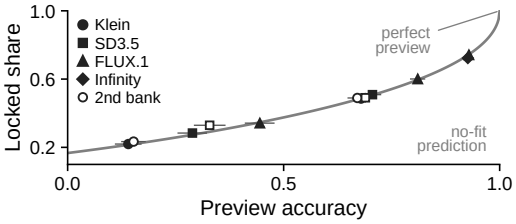

Figure 4: Better previews lock more requests: locked share against preview accuracy (preview–final rank correlation, converted to a Gaussian correlation) at up to three checkpoints per backbone. Open: second bank; bars: 95% intervals. Curve: Proposition 1 at $N = 6$, no fitted parameter.

Table 2 compares the conversion of free and required versions at density's development ImageReward, where set diversity is the mean DreamSim distance among survivors. Requiring the champion leaves the survivor set nearly as diverse or more (Figure 11), but conversion falls to less than half for both selectors on every backbone. On a separate sixteen-candidate bank, the gap holds for every pool of 6 to 16 candidates and 2 to 4 survivors on Klein and SD3.5, largest at $k = 2$ (Appendix G).

At the second bank's development strengths, requiring the champion lowers output diversity and the mode count for both selectors and backbones while raising ImageReward. All four output-diversity differences exclude zero after correction, as do all eight on the main benchmark. On the main benchmark the requirement removes more than half of each selector's gain over pruning, while gaining less than half the 0.050 tolerance in ImageReward. About half the second-bank requests are locked, and preview quality predicts this share (Figure 4): Proposition 1, evaluated at the measured rank correlation, matches eleven of twelve measured shares within 0.02 with no fitted parameter. On a language model, no champion-preserving selector changes the response on a locked request, and the prediction fits there too (Appendix H).

The effect holds at equal quality. Keeping the champion also changes reward, so we compare both versions at matched ImageReward, at 25% and 50% of the reward range both reach. The free version delivers more output diversity, intervals above zero, in 15 of 16 selector–backbone comparisons on the main benchmark and 7 of 8 on the second bank, at both levels. Near reward-only pruning the gap narrows but still amounts to 18% to 28% of the free version's gain, positive when pooled. An exact per-request decomposition places the gap on the locked requests (Appendix F.4).

The replacement matters too. Dropping the champion opens the locked requests; the replacement decides their output. A control that drops the champion on density's requests but fills the slots randomly, weighted by preview reward, changes winners at least as often, yet trails density in DreamSim at matched ImageReward on all four backbones, intervals excluding zero (Appendix F.5).

Noisy previews are not the cause. An oracle ranking candidates by final reward locks every request, so every champion-preserving rule delivers the pool's best final image (Lemma 1), yet the free selectors still raise delivered DreamSim on every diffusion backbone of both banks. Dropping the champion is a source of diversity in its own right, not a correction of noisy previews (Appendix F.6).

6.4 What density adds: rare modes

Density matches exact max-sum on mean diversity: pooled over backbones, their DreamSim difference at matched ImageReward has intervals covering zero on both banks. They differ in which content reaches the user: density's rare-mode share is higher by 1.6 and 3.1 percentage points on the two banks (7% and 14% relative), with intervals excluding zero, and it delivers more distinct modes. Its PickScore stays within a quarter of that judge's tolerance of max-sum (Appendix D.5).

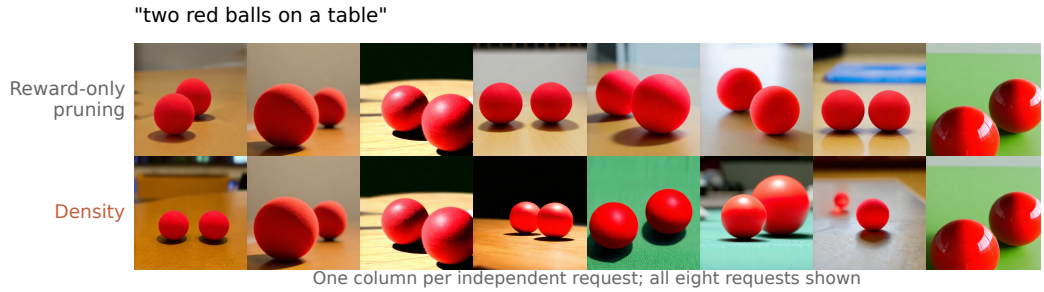

Figure 5: One SD3.5 prompt of the blinded study (95th percentile of DreamSim gains), from the same candidates and budget; both raters judged density more varied and the quality equal.

On the three diffusion backbones, which keep two survivors, requests with at least two rare candidates carry about three quarters of density's advantage on both banks (Figure 6), as Proposition 2 predicts. The rest comes from single rare candidates that win the final comparison.

On the sixteen-candidate bank, with two survivors, density's rare-mode advantage grows from 0.4 points at six candidates to 2.3 at 16, with an interval above zero, and greedy max-sum falls further behind in DreamSim. Only champion-free rules use the extra candidates. The rare-mode advantage is positive under 119 of 120 cluster definitions and holds in OpenCLIP space and without clusters; the distinct-mode difference depends more on the definition (Appendix D.6).

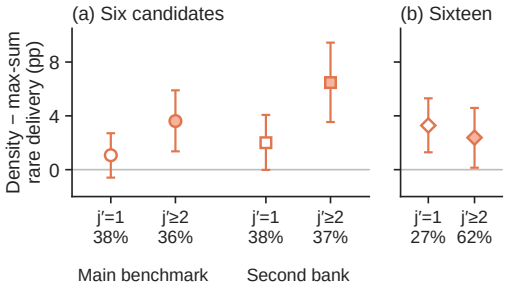

Figure 6: Density minus max-sum in rare-mode delivery (points, 95% intervals) at matched ImageReward, by the number j' of rare pool candidates, with each group's share of requests. (a) Six, (b) sixteen candidates. Weighted by share, most of the advantage comes from $j' \geq 2$ (Proposition 2).

6.5 Human evaluation

Two raters who are not authors compared eight-image strips from density and from reward-only pruning on the same 100 prompts for SD3.5 and for Klein. Blind to method and row order, they judged variety and quality, with "same" allowed. Counting a tie as half, they preferred density's variety in 72.5% [65.5%, 79.0%] of comparisons on SD3.5 and 70.3% [62.5%, 77.5%] on Klein (Cohen's $\kappa = 0.77$ and 0.71). Quality preference was 49.25% [47.75%, 50.5%] and 50.0% [49.25%, 50.75%]; both raters judged quality equal on 94 and 98 of 100 prompts. On both backbones, variety judgements track DreamSim, from about 40% where it favours pruning to about 95% on the largest gains (Appendix I, Figure 5).

7 Conclusion

Set diversity is not output diversity: with one image per request, diversity reaches the user only through each request's winner, which champion-preserving selectors cannot change on locked requests, 49% to 72% of all requests. Freeing the champion removes this constraint: both champion-free selectors raise output diversity on four backbones at a fixed budget, and density, a single sort, brings rare modes to the user when few candidates survive. Returning several images per request adds the within-request term of Eq. (2), which max-sum's spread term targets. The constraint needs only a preview ranking and a final reward, so it binds wherever candidates are pruned on partial evaluations and one best survivor is kept, as it does on a language model (Appendix H). Loops that keep their best candidates, such as self-training on top-reward samples (Gulcehre et al., 2023; Sessa et al., 2025) and evolutionary program search, where elitism plays the champion's role (Romera-Paredes et al., 2024; Novikov et al., 2025), are natural next targets.

REPRODUCIBILITY STATEMENT

Code: https://anonymous.4open.science/r/progressive-selection-B85A. This anonymous release contains the code and the inputs needed to rerun the experiments: prompts, model settings, selection strengths and calibration constants. Model versions and sampler settings are listed in Appendix B.

ETHICS STATEMENT

Reward models and visual representations may encode preference and demographic biases, and greater perceptual distance does not imply more inclusive or accurate generation. Both blinded studies (Section 6.5, Appendix I) were rated by the same two colleagues, who are not authors of this paper and had no part in designing the selectors or choosing the strengths. They were told what the task was and what the ratings would be used for, and that they could stop at any time; they took part voluntarily and without compensation. Raters saw only generated images and their prompts. No personal data was recorded, and the ratings carry no information that identifies a rater.

AI USE STATEMENT

AI tools assisted literature discovery, hypothesis refinement, planning, implementation, analysis and drafting. The authors take full responsibility for the paper's text, citations, mathematics, code and empirical claims.

REFERENCES

Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026. Model weights: https://huggingface.co/black-forest-labs/FLUX.2-klein-4B.

Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

Gabriele Corso, Yilun Xu, Valentin de Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle Guidance: non-I.I.D. Diverse Sampling with Diffusion Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.13102.

Bradley Efron. Tweedie's Formula and Selection Bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A Diversity Evaluation Metric for Machine Learning. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2210.02410.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning New Dimensions of Human Visual Similarity using Synthetic Data. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.09344.

Leo Gao, John Schulman, and Jacob Hilton. Scaling Laws for Reward Model Overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

David E. Goldberg and Jon Richardson. Genetic Algorithms with Sharing for Multimodal Function Optimization. In *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, 1987.

Sreenivas Gollapudi and Aneesh Sharma. An Axiomatic Approach for Result Diversification. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 381–390, 2009.

Rogério Guimarães and Pietro Perona. Inference-Time Scaling of Diffusion Models via Progressive Seed Pruning. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2607.21591.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling Bitwise AutoRegressive Modeling for High-Resolution Image Synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15733–15744, 2025.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Rohit Jena, Ali Taghibakhshi, Sahil Jain, Gerald Shen, Nima Tajbakhsh, and Arash Vahdat. Elucidating Optimal Reward-Diversity Tradeoffs in Text-to-Image Diffusion Models. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 232–242, 2025. doi: 10.1109/WACV61041.2025.00033.

Dongjae Jeon, Dueun Kim, and Albert No. Understanding and mitigating memorization in generative models via sharpness of probability landscapes. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27091–27112, 2025. URL https://proceedings.mlr.press/v267/jeon25a.html. Introduces Sharpness-Aware Initialization (SAIL) and the 500-prompt hybrid general prompt set.

Byungjun Kim, Soobin Um, and Jong Chul Ye. Diverse text-to-image generation via contrastive noise optimization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.03813.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the Effects of RLHF on LLM Generalisation and Diversity. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06452.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An Open Dataset of User Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.01569.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, 2023.

Alex Kulesza and Ben Taskar. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

Joel Lehman and Kenneth O. Stanley. Abandoning Objectives: Evolution Through the Search for Novelty Alone. *Evolutionary Computation*, 19(2):189–223, 2011a. doi: 10.1162/EVCO_a_00025.

Joel Lehman and Kenneth O. Stanley. Evolving a Diversity of Virtual Creatures through Novelty Search and Local Competition. In *Genetic and Evolutionary Computation Conference*, pp. 211–218, 2011b. doi: 10.1145/2001576.2001606.

Xiang Li, Dianbo Liu, and Kenji Kawaguchi. Initialization is half the battle: Generating diverse images from a guidance potential posterior. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zXJjT0slV7. Diversity-inducing Initialization (DivIn).

Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Shuiwang Ji. Dynamic Search for Inference-Time Alignment in Diffusion Models, 2025a. URL https://arxiv.org/abs/2503.02039.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-Free Guidance in Continuous and Discrete Diffusion Models with Soft Value-based Decoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://arxiv.org/abs/2408.08252.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-T1: Test-time Scaling for Video Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18671–18681, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Liu_Video-T1_Test-time_Scaling_for_Video_Generation_ICCV_2025_paper.html.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling Inference Time Compute for Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025. doi: 10.1109/CVPR52734.2025.00241. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ma_Scaling_Inference_Time_Compute_for_Diffusion_Models_CVPR_2025_paper.html.

Jean-Baptiste Mouret and Jeff Clune. Illuminating Search Spaces by Mapping Elites, 2015. URL https://arxiv.org/abs/1504.04909.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Gaurav Parmar, Or Patashnik, Daniil Ostashev, Kuan-Chieh Wang, Kfir Aberman, Srinivasa Narasimhan, and Jun-Yan Zhu. Scaling Group Inference for Diverse and High-Quality Generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.15773.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI*, 3:40, 2016. doi: 10.3389/frobt.2016.00040.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical Discoveries from Program Search with Large Language Models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zMoNrajk2X.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot-Desenonges, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abram Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. BOND: Aligning LLMs with Best-of-N Distillation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.14622.

Shivanshu Shekhar, Sagnik Mukherjee, Jia Yi Zhang, and Tong Zhang. Vasr: Variance-aware systematic resampling for reward-guided diffusion, 2026. URL https://arxiv.org/abs/2604.06779.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A General Framework for Inference-time Scaling and Steering of Diffusion Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55810–55827, 2025. URL https://proceedings.mlr.press/v267/singhal25b.html.

Abhijeet Sinha, Sundari Elango, and Dianbo Liu. Expected Return Causes Outcome-Level Mode Collapse in Reinforcement Learning and How to Fix It with Inverse Probability Scaling. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Vr0kjNERRf.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter L. Bartlett, and Andrea Zanette. Fast Best-of-N Decoding via Speculative Rejection.

In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2410.20290.

Revant Teotia, Candace Ross, Karen Ullrich, Sumit Chopra, Adriana Romero-Soriano, Melissa Hall, and Matthew Muckley. DIMCIM: A Quantitative Evaluation Framework for Default-mode Diversity and Generalization in Text-to-Image Generative Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16431–16440, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Teotia_DIMCIM_A_Quantitative_Evaluation_Framework_for_Default-mode_Diversity_and_Generalization_ICCV_2025_paper.html.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse Beam Search for Improved Description of Complex Scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.12340. URL https://ojs.aaai.org/index.php/AAAI/article/view/12340.

Gal Vinograd, Idan Achituve, and Ethan Fetaya. Diverse Sampling in Diffusion Models with Marginal Preserving Particle Guidance, 2026. URL https://arxiv.org/abs/2605.06553.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and Asymptotically Exact Conditional Sampling in Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.17775.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2304.05977.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=AFDcYJKhND.

Qingtao Yu, Changlin Song, Minghao Sun, Zhengyang Yu, Vinay Kumar Verma, Soumya Roy, Sumit Negi, Hongdong Li, and Dylan Campbell. Rethinking Test Time Scaling for Flow-Matching Generative Models, 2025. URL https://arxiv.org/abs/2511.22242.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Ziyi Zhang, Sen Zhang, Yibing Zhan, Yong Luo, Yonggang Wen, and Dacheng Tao. Confronting Reward Overoptimization for Diffusion Models: A Perspective of Inductive and Primacy Biases. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60396–60413, 2024. URL https://proceedings.mlr.press/v235/zhang24ch.html.

Guide to the appendix. Appendix A proves the results of Sections 3 to 5 and explains why every main result delivers by argmax (Appendix A.6). Appendices B and C give the implementation, the human-scale quality tolerances and the evaluation protocol. The rest follows Section 6: full results under other judges, prompt suites, diversity measures and mode definitions (Appendix D) and further baselines (Appendix E) for Section 6.2; the champion interventions (Appendix F), larger pools (Appendix G) and a language-model replication (Appendix H) for Sections 6.3 and 6.4; the human study (Appendix I) and qualitative examples (Appendix J).

A PROOFS AND THEORY

This appendix supports the analysis of Sections 3 to 5. It proves Lemma 1, the exactness of greedy max-sum at $k = 2$ and the locked-share law of Proposition 1, derives the decomposition of output diversity in Eq. (2), and shows by example that set diversity does not order selectors by output diversity. It then states the rare-mode result of Proposition 2 with its exact conditions and proves it, and closes with the choice of final rule.

A.1 THE CHAMPION CONSTRAINT

We use the notation of Section 4: one request, a pool of N candidates, preview rewards v_i, final rewards R_i, preview champion a and final winner $a^\star$. Two selectors are compared on the same pool, with the same previews and the same native continuations; their internal randomness may differ. Ties are broken by one fixed rule, shared by all selectors.

Proof of Lemma 1. Suppose the request is locked, so $a = a^\star$. A champion-preserving selector keeps a set S with $a \in S$, so S contains the final winner of the whole pool. Hence $Y(S) = \arg\max_{i \in S} R_i = a$, whichever companions S contains. Two champion-preserving selectors therefore deliver the same candidate on every locked request, which gives $\mathbf{1}\{Y(S_1) \neq Y(S_2)\} \leq \mathbf{1}\{a \neq a^\star\}$ request by request. Taking expectations proves the bound. □

The bound concerns the identity of the winner. It does not say that every unlocked request changes the winner, nor that a new winner shows new content. If previews were perfect, every request would be locked, and every champion-preserving selector would return the final winner of the whole pool; Appendix F.6 uses this case.

Greedy max-sum is exact at $k = 2$. Section 4 uses the following fact: for $k = 2$, champion-first greedy max-sum returns the maximiser of J in Eq. (3) over all pairs that contain a.

Proof. With $k = 2$, every pair that contains a has the form $\{a, j\}$ and scores $J(\{a, j\}) = (v_a + v_j)/2 + \lambda d_{aj}/\bar{d}$. The term $v_a/2$ does not depend on j. Maximising over j is therefore maximising $v_j/2 + \lambda d_{aj}/\bar{d}$, which is exactly the greedy step that adds the best companion to $\{a\}$. □

Coefficient alignment. Eq. (3) is the mean-based score that our code evaluates: mean scaled reward plus λ times the mean pairwise distance divided by $\bar{d}$, the development median of preview distances. Multiplying by k gives the sum form $\sum_{i \in S} v_i + \eta \sum_{i<j \in S} d_{ij}$ with $\eta = 2\lambda/((k-1)\bar{d})$, so $\eta = 2\lambda/\bar{d}$ when $k = 2$. The champion-first greedy rule scores each partial set with the same mean-based objective, so the greedy and exact rules optimise the same function at every k. We checked that greedy and constrained exact selection coincide on every request and strength at $k = 2$. At $k = 3$ (Infinity) we enumerate the constrained and unconstrained optima directly.

A reward floor for champion-preserving rules. Keeping the champion also limits how much reward a selector can give up. For $k \geq 2$, let $R^{(-a)}_{(k-1)}$ be the $(k-1)$-th smallest final reward among the candidates other than a. Then

$$\min_{S:\,|S|=k,\ a \in S} \max_{i \in S} R_i = \max\{R_a,\ R^{(-a)}_{(k-1)}\}.$$

Every feasible set contains $k-1$ companions of a, and the best of them has at least this order statistic; keeping the $k-1$ weakest companions attains it. Averaged over the main benchmark,

the floor is 1.3691, 1.3957, 1.4113 and 1.3250 ImageReward on Klein, SD3.5, FLUX.1-dev and Infinity-2B. Density delivers 1.3828, 1.4059, 1.4291 and 1.3243 at its development strengths. On Infinity, therefore, no champion-preserving rule can give up as much reward as density does, so no reward-matched required rule exists there.

A.2 THE LOCKED-SHARE LAW

Throughout, $\gamma = \rho/\sqrt{1-\rho^2}$, which ranges over $[0, \infty)$ as ρ ranges over $[0, 1)$.

Proof of Proposition 1. The proof has three parts: invariance, monotonicity in ρ with the two limits, and the decrease in N.

Invariance. Write $v_i = \mu_v + \sigma_v Z_i$ and $R_i = \mu_R + \sigma_R(\rho Z_i + \sqrt{1-\rho^2}\,\varepsilon_i)$ with Z_i, ε_i i.i.d. standard normal. The maximisers of v and of R do not change under increasing affine maps, so L does not depend on the means or scales, and we take $\mu = 0$ and $\sigma = 1$. By exchangeability, $L = N \Pr\{Z_1 = \max_j Z_j,\ R_1 = \max_j R_j\}$. The event $\{Z_1 = \max_j Z_j\}$ with $Z_1 = x$ has density $\varphi(x)\Phi(x)^{N-1}$, and given it the other previews are independent standard normals truncated to $(-\infty, x)$. What remains is the probability $\pi_N(x, \rho)$ that $\rho x + \sqrt{1-\rho^2}\,\varepsilon_1$ exceeds every $\rho Z_j + \sqrt{1-\rho^2}\,\varepsilon_j$, the probability that a candidate with preview x has the highest final reward when the other $N-1$ previews lie below x. Hence

$$L(N, \rho) = N \int \varphi(x)\, \Phi(x)^{N-1} \pi_N(x, \rho)\, dx,$$

a fixed function of N and ρ.

Monotonicity in ρ. Condition on the previews Z and let $a = \arg\max_i Z_i$. For $\rho < 1$, $\arg\max_i R_i = \arg\max_i(\gamma Z_i + \varepsilon_i)$, so

$$\Pr\{a^\star = a \mid Z\} = \Pr\big\{\varepsilon_j - \varepsilon_a \le \gamma\,(Z_a - Z_j) \text{ for all } j \ne a \;\big|\; Z\big\}.$$

Almost surely every threshold $\gamma(Z_a - Z_j)$ is positive and increases strictly with γ, and the differences $\varepsilon_j - \varepsilon_a$ have a positive joint density, so the conditional probability increases strictly in γ, hence in ρ; averaging over Z keeps the strict increase. At $\gamma = 0$ the probability is $1/N$ by the symmetry of the ε_j, and as $\gamma \to \infty$ every threshold diverges and the probability tends to one.

Decrease in N. Let $u_1 \le \cdots \le u_{N-1}$ be the gaps between the largest preview and the others, $u_j = Z_{(N)} - Z_{(N-j)}$. Conditioning on the previews and on $\varepsilon_a = e$, the display above gives

$$L(N, \rho) = \mathbb{E}\,\psi_N(u_1, \ldots, u_{N-1}), \qquad \psi_N(u) = \int \varphi(e) \prod_{j=1}^{N-1} \Phi(e + \gamma u_j)\, de. \tag{5}$$

Since $\gamma \ge 0$, ψ_N is nondecreasing in every gap, and dropping a factor cannot decrease it because $\Phi \le 1$. It therefore suffices to couple pools of sizes N and $N+1$ so that the first $N-1$ gaps of the larger pool are no larger than those of the smaller one. We do this in four steps. (i) Let $H(x) = -\log(1 - \Phi(x))$ be the cumulative hazard of the standard normal and $g = H^{-1}$. If $E \sim \mathrm{Exp}(1)$, then $g(E)$ is standard normal; since g is increasing, the order statistics of a normal sample are the images under g of those of an $\mathrm{Exp}(1)$ sample. (ii) By Rényi's representation, the spacing between the m-th and $(m+1)$-th largest of N i.i.d. $\mathrm{Exp}(1)$ variables is Y_m/m, with $Y_1, Y_2, \ldots$ i.i.d. $\mathrm{Exp}(1)$, and the smallest is Y_N/N. The maximum can therefore be generated as $M_N = \sum_{m=1}^{N} Y_m/m$, and its gaps to the next $N-1$ order statistics are $s_j = \sum_{m=1}^{j} Y_m/m$, which do not depend on N. (iii) Building both pools from the same Y_m gives $M_{N+1} = M_N + Y_{N+1}/(N+1) \ge M_N$ with shared $s_1, \ldots, s_{N-1}$, and normal gaps $u_j^{(N)} = g(M_N) - g(M_N - s_j)$. (iv) The normal hazard rate $\varphi/(1-\Phi)$ is increasing, so H is convex and g is concave, and for a concave g the increment $g(t) - g(t-s)$ over an interval of fixed length $s \ge 0$ is nonincreasing in t. Hence $u_j^{(N+1)} \le u_j^{(N)}$ for every $j \le N-1$ on every sample path, and

$$\psi_{N+1}\big(u_1^{(N+1)}, \ldots, u_N^{(N+1)}\big) \le \psi_N\big(u_1^{(N+1)}, \ldots, u_{N-1}^{(N+1)}\big) \le \psi_N\big(u_1^{(N)}, \ldots, u_{N-1}^{(N)}\big).$$

Taking expectations in Eq. (5) gives $L(N+1, \rho) \le L(N, \rho)$. The first inequality is strict on every sample path, because the retained integrand is positive for every e and the dropped factor $\Phi(e + \gamma u_N^{(N+1)})$ is below one; the decrease is therefore strict for every $\rho \in [0, 1)$. □

Scope of the decrease in N. The same argument covers a more general model: $a^\star = \arg\max_i(\gamma v_i + \varepsilon_i)$ with $\gamma \geq 0$, previews v_i i.i.d. from a continuous law F with nondecreasing hazard rate (every log-concave density qualifies), and noises ε_i i.i.d. from any law G, independent of the previews. Replacing H by $-\log(1-F)$ and φ, Φ by dG, G, the weak inequality $L(N+1) \leq L(N)$ holds. It is strict whenever the noise is non-degenerate: choose $e_1 < e_2$ with $G(e_1) > 0$ and $\Pr\{\varepsilon \geq e_2\} > 0$. On the event that γ times the range of the previews is below $e_2 - e_1$, which has positive probability, the dropped factor is at most $1 - \Pr\{\varepsilon \geq e_2\}$ for every $e \leq e_1$, while each retained factor is at least $G(e) > 0$ for G-almost every such e. Some shape condition on the previews is necessary. With Pareto previews of index one, $\gamma = 1$ and Gaussian noise, the locked share rises from 0.81 at $N = 3$ to 0.84 at $N = 5$ (2×10^6 Monte Carlo draws each), because a larger pool makes an isolated top candidate more likely; a bimodal preview reward with a rare high mode behaves likewise.

Size of the effects. Table 3 gives $L(N, \rho)$ from 10^6 Monte Carlo draws per entry; the $N = 2$ column agrees with the closed form $\frac{1}{2} + \arcsin(\rho)/\pi$ to within 0.001, the Monte Carlo error. For ρ between 0.6 and 0.7, each doubling of N multiplies L by 0.73 to 0.84, and by 0.77 to 0.84 from $N = 8$ on; the factor rises slowly with N, so the decrease is slow. Appendix F.3 compares the law with measured locked shares, across checkpoints and across pool sizes.

Table 3: Locked share $L(N, \rho)$ when preview and final rewards are jointly Gaussian with correlation ρ (Proposition 1; Monte Carlo, 10^6 draws per entry).

ρ	N=2	4	6	8	12	16	32	64	128
0.3	0.597	0.367	0.278	0.230	0.177	0.147	0.096	0.063	0.042
0.4	0.632	0.411	0.325	0.276	0.220	0.188	0.130	0.091	0.066
0.5	0.667	0.460	0.376	0.326	0.270	0.237	0.174	0.131	0.099
0.6	0.704	0.514	0.435	0.386	0.330	0.296	0.231	0.181	0.145
0.65	0.725	0.545	0.467	0.419	0.364	0.331	0.265	0.214	0.174
0.7	0.747	0.577	0.502	0.457	0.402	0.370	0.303	0.251	0.211
0.8	0.795	0.652	0.586	0.547	0.497	0.465	0.402	0.350	0.308
0.9	0.856	0.750	0.700	0.669	0.629	0.603	0.550	0.503	0.465
0.95	0.899	0.822	0.784	0.761	0.729	0.710	0.668	0.630	0.598

A.3 Within- and between-request diversity

Let S be the completed survivors of one request, and let the final rule deliver survivor i with probability $w_i(\mathsf{S})$. Give the final image y_i of survivor i a unit feature z_i and use the cosine distance $d(y_i, y_j) = 1 - z_i^\top z_j$. DreamSim distance is one minus the cosine similarity of DreamSim embeddings, so everything below applies to it exactly. Write $m(\mathsf{S}) = \sum_i w_i z_i$ and $W(\mathsf{S}) = \sum_{i,j} w_i w_j (1 - z_i^\top z_j)$. Then, for a fixed prompt, Eq. (2) reads

$$D_c(A) = \mathbb{E}\, W(\mathsf{S}) + \mathbb{E}\left\|m(\mathsf{S}) - \mathbb{E}\, m(\mathsf{S})\right\|^2.$$

Proof. Two requests have independent survivor states S and S'. Given both, the expected distance between their deliveries is $\sum_{i,j} w_i w'_j (1 - z_i^\top z'_j) = 1 - m(\mathsf{S})^\top m(\mathsf{S}')$, so $D_c(A) = 1 - \|\mathbb{E}m\|^2$. Because the weights sum to one, $W(\mathsf{S}) = 1 - \|m(\mathsf{S})\|^2$. Hence $1 - \|\mathbb{E}m\|^2 = \mathbb{E}(1 - \|m\|^2) + \big(\mathbb{E}\|m\|^2 - \|\mathbb{E}m\|^2\big)$, which is Eq. (2). □

Three final rules illustrate the first term. Under argmax one weight equals one, so $W = 0$ and all output diversity is variation of the winner across requests. Uniform delivery over k survivors gives $W = \frac{k-1}{k} G(\mathsf{S})$, where G is the mean distance between distinct survivors. With two survivors at distance d_{12}, a reward gap Δ and a softmax of inverse temperature θ, $W = d_{12}/(2\cosh^2(\theta\Delta/2))$. A distant companion with lower reward therefore adds diversity only under a soft final rule, and not at all under argmax. The identity also holds for request-level mixtures of policies once the mixture component is included in S; Appendix C.3 uses this to evaluate mixtures exactly.

Under argmax delivery $m(\mathsf{S}) = z_Y$, so $D_c(A) = 1 - \|\mu_A\|^2$ with $\mu_A = \mathbb{E}\, z_{Y_A}$. Compare a champion-preserving selector A with reward-only pruning A_0 on the same pools, and let $\delta = \mu_{A_0} - \mu_A$. By Lemma 1 the two deliver the same image on every locked request, so $\delta = (1 - L_c)\,\mathbb{E}[z_{Y_{A_0}} - z_{Y_A} \mid a \neq a^\star]$, with L_c the locked share of prompt c, and exactly $D_c(A) - D_c(A_0) = 2\langle \mu_{A_0}, \delta\rangle - \|\delta\|^2$. The gain therefore depends on the mean displacement of the delivered feature on unlocked requests, not on how far individual deliveries move.

A.4 Set diversity does not order output diversity

Draw six candidates independently from two modes A and B with probability $\frac{1}{2}$ each, keep two, and deliver by final reward with $R_A = 1 > R_B = 0$. Features are orthogonal across modes, so the cross-mode distance is one. Selector 1 keeps one candidate of each mode whenever both are present. Its set diversity is $62/64$, yet it delivers B only when all six candidates are B, so its output diversity is $2 \cdot \frac{1}{64} \cdot \frac{63}{64} = 126/4096$. Selector 2 keeps two candidates of the majority mode and breaks a 3:3 tie by the mode of the first candidate. Its set diversity is zero, but it delivers A and B equally often, so its output diversity is $1/2$. Set diversity ranks Selector 1 first, and output diversity ranks Selector 2 first. The two selectors deliver different rewards, so the example shows a reversal of order, not a better trade-off.

A.5 Rare modes in a two-mode model

We now make Proposition 2 precise. Candidates carry mode labels $h(i)$, and distances are block distances, $d_0(i, j) = \mathbf{1}\{h(i) \neq h(j)\}$. In this geometry max-sum always keeps the champion.

Proposition 3 (Max-sum keeps the champion under block distance)**.** *Let a be the unique maximiser of preview reward. For every k and every $\lambda \geq 0$, every maximiser of J in Eq. (3) under d_0 contains a. Consequently, on a locked request exact max-sum delivers the same image as reward-only pruning at every strength, and the same holds over several cuts if each cut has block geometry.*

Proof. Suppose an optimal set S excludes a. If S contains a candidate b from a's mode, replacing b by a leaves every distance unchanged and raises the reward term. If a's mode is absent from S, replacing any member by a gives distance one to all others, so no distance falls and the reward term rises. Both cases contradict optimality. Applying the argument at each cut proves the multi-cut statement; the partition into modes may differ between cuts. □

The conclusion survives small departures from block geometry. If $|d - d_0| \leq \epsilon$ everywhere, a scaled preview gap $v_a - v_i$ to every other candidate larger than $2\eta(k-1)\epsilon$ still forces a into every optimum, with η the sum-form coefficient. It does not survive general geometry. Take three unit vectors at angles $0°$, $60°$ and $-60°$ with scaled preview rewards 1, 0.99 and 0.98. The champion is at cosine distance 0.5 from each of the others, which are 1.5 apart. For $k = 2$ the pair without the champion wins as soon as $\eta > 0.02$. In real features exact max-sum does drop the champion, on 19.9% to 38.0% of requests of the second bank (Appendix F.1).

Density behaves differently because its crowding term counts the whole pool. Consider N independent candidates, each from a rare mode B with probability p and from a common mode A otherwise. Rewards are constant within a mode, $r_A - r_B = \Delta > 0$, and previews rank candidates as final rewards do. Candidates of one mode share their features, and candidates of different modes have kernel similarity $b \in [0, 1)$, so a candidate in a mode with m members of the pool has crowding $m + b(N - m)$; a block kernel, $b = 0$, makes the crowding the size of the mode. One cut keeps k candidates, and argmax delivers one of them. In reward units, the density score is $r_i - \kappa \log\big(m_{h(i)} + b(N - m_{h(i)})\big)$, with m_h the number of pool candidates in mode h and crowding weight $\kappa = \sigma_r/\beta$.

Proposition 4 (Capacity threshold for a single rare mode)**.** *Suppose the pool is mixed and holds j candidates from B, with $0 < j < N$. Density delivers B if and only if $j \geq k$ and $\Delta < \kappa \log \frac{N-j+bj}{j+b(N-j)}$. A pool made only of B delivers B under every rule. If $\Delta < \kappa \log \frac{N-k+bk}{k+b(N-k)}$, which requires $N > 2k$, then as $p \to 0$ the probability q_B that a rule delivers B satisfies*

$$q_B^{\text{density}} = \tbinom{N}{k} p^k + O(p^{k+1}), \qquad q_B^{\text{reward}} = q_B^{\text{max-sum}} = p^N. \tag{6}$$

Proof. Any surviving A wins the final argmax, so B is delivered exactly when all k survivors are B. This is impossible when $j < k$. When $j \geq k$, all survivors are B exactly when every B outscores every A, that is, when $r_B - \kappa \log(j + b(N-j)) > r_A - \kappa \log(N - j + bj)$, which is the stated condition. Summing over the binomial number of B candidates gives the exact law

$$q_B^{\text{density}} = p^N + \sum_{j=k}^{N-1} \tbinom{N}{j} p^j (1-p)^{N-j} \, \mathbf{1}\Big\{\kappa \log \tfrac{N-j+bj}{j+b(N-j)} > \Delta\Big\}.$$

The margin $\log \frac{N-j+bj}{j+b(N-j)}$ decreases in j, so $j = k$ is the easiest eligible count. It qualifies exactly when $\Delta < \kappa \log \frac{N-k+bk}{k+b(N-k)}$, which needs $N > 2k$, and it contributes the leading term $\binom{N}{k} p^k$. Reward-only pruning keeps an A whenever one exists. So does max-sum: replacing a B in an all-B set by an available A raises the reward term and raises every distance to that candidate from zero to one. Both rules therefore deliver B only from an all-B pool, which has probability p^N. □

Three remarks delimit the result. First, leakage $b > 0$ shrinks the margin, $N > 2k$ stays necessary, and as $b \to 1$ the margin vanishes. Second, the contrast between p^k and p^N is specific to argmax delivery. A softmax final rule with inverse temperature θ delivers a retained single B with probability $Np\, e^{-\theta\Delta}/(k-1+e^{-\theta\Delta}) + O(p^2)$, and any rule that retains that single B benefits, max-sum included when $\sigma_r \eta(k-1) > \Delta$. Third, when $N \leq 2k$, as for Infinity's $6 \to 3$ cut, a single rare mode cannot fill every slot, but rare candidates from different modes can (Proposition 5 below). Exhaustive enumeration confirms the exact law in 324 parameter cases, and 28,000 exact subset searches on random block instances confirm Proposition 3.

Rare content in several modes. Proposition 2 is stated for rare content, which may spread over several rare modes, as the rare-mode share of Section 6.1 does. Let the pool hold one common mode A and any number of rare modes, each candidate being rare independently with probability p, with block distances d_0 and cross-mode kernel similarity $b \in [0, 1)$ as above, rewards constant within a mode, and $0 < r_A - r_h \leq \Delta$ for every rare mode h.

Proposition 5 (Rare content; formal version of Proposition 2)**.** *Suppose the pool holds j' rare candidates, $0 < j' < N$, and at most c of them share a rare mode. Reward-only pruning and exact max-sum deliver a rare image only from an all-rare pool. Density delivers a rare image whenever $j' \geq k$ and $\Delta < \kappa \log \frac{N-j'+bj'}{c+b(N-c)}$. Hence, if $\Delta < \kappa \log \frac{N-k+bk}{k+b(N-k)}$, density delivers a rare image with probability $\binom{N}{k} p^k + O(p^{k+1})$ as $p \to 0$, against p^N for the other two rules. If every rare candidate lies in a mode of its own, $c = 1$ and $\Delta < \kappa \log \frac{N-k+bk}{1+b(N-1)}$ suffices, which needs only $N - k \geq 2$.*

Proof. Any surviving A wins the final argmax, so a rare image is delivered exactly when all k survivors are rare, which needs $j' \geq k$. A rare candidate in mode h has crowding $m_h + b(N - m_h) \leq c + b(N-c)$ and density score at least $r_A - \Delta - \kappa \log(c + b(N-c))$, while every A candidate scores $r_A - \kappa \log(N - j' + bj')$. Under the stated condition every rare candidate outscores every A candidate, so all k survivors are rare. Exact max-sum keeps a candidate from A whenever the pool is mixed: replacing a rare member of an all-rare set by an available A candidate raises the reward term and lowers no distance. Reward-only pruning keeps one as well. With exactly k rare candidates, $c \leq k$, so $\Delta < \kappa \log \frac{N-k+bk}{k+b(N-k)}$ covers every way of splitting them into modes; pools with fewer than k rare candidates never deliver one. This gives the leading term $\binom{N}{k} p^k$. □

For isolated rare candidates, exhaustive enumeration on pools of three to eight candidates with $k \in \{2, 3\}$ finds no counterexample: exact max-sum keeps the champion in all 2,071,872 instances, and density delivers a rare image in each of the 21,312 instances that meet the condition, for each of four cross-block kernel similarities. On real pools, density's rare-mode advantage over max-sum comes mainly from requests holding at least k rare candidates, as both versions predict (Figure 6 and Appendix D.5).

A.6 The final rule: why argmax, and what a soft rule changes

Why argmax. Three reasons fix the final rule before any experiment. First, it is the rule of the pipelines this paper studies. Best-of-N, reward-only pruning, our Feynman–Kac baseline (Ap-

pendix E.1) and best-of-N decoding in language models all return the candidate the reward ranks first, and the champion constraint is a property of these pipelines; the final rule is part of the problem, not a design choice of the method. Second, it is the only rule that does not waste the budget at the last step. When the final rule acts, every survivor has been finished and its cost paid; argmax delivers the highest final reward those survivors offer, and any other rule gives up expected reward for the same cost. Third, it isolates the effect under study. Under argmax one delivery weight equals one, so the within-request term of Eq. (2) vanishes and every change in output diversity comes from which image wins, that is, from selection. A soft rule adds a second lever that acts after selection, ignores the geometry of the pool and is available to every selector alike, reward-only pruning included; we therefore evaluate it as a baseline rather than build it into the method.

What depends on it. Every main result uses argmax; only the softening study of Appendix E.3 departs from it. Lemma 1 uses this rule: a locked request fixes the output because the final comparison picks the pool's final winner whenever it survives. Proposition 2 uses it as well, since a surviving common candidate wins that comparison and rare content is delivered only when every survivor is rare. Proposition 1 describes the locked share itself, which does not depend on the final rule; the rule decides what a locked request implies.

What a soft rule changes. Under a softmax final rule of inverse temperature θ, the champion of a locked request keeps the largest delivery weight among the survivors, but a weight below one, and its companions share the remaining mass. Output diversity then gains the within-request term of Eq. (2), which shrinks as the survivors' reward gap grows (Appendix A.3). Softening is thus a second route from set diversity to output diversity. It acts on every request, locked or not, and its reward cost depends only on the survivors' reward gaps, not on their geometry. The contrast of Proposition 2 between p^k and p^N is likewise specific to argmax: a soft rule also delivers a retained single rare candidate, and any rule that retains it benefits (Appendix A.5).

The measurements agree. Appendix E.3 lets each family tune its strength and θ jointly under a common quality budget. No softened configuration lies above the argmax curve of the same selector, and softening helps reward-only pruning far less than champion-free selection does: champion-free selection at the preview buys diversity with the geometry of the pool, softening buys it with reward alone. The within-request term matters by design only when several images are returned per request, the extension noted in Section 7.

B Implementation

This appendix gives what is needed to reimplement both selectors and to check the cost statements of Sections 5 and 6.1. It covers the backbones and samplers, the construction of previews, the features and calibration constants, the human-scale quality tolerances and how strengths were chosen, the schedule and its budget, the full algorithm and measured runtime.

B.1 Backbones and samplers

FLUX.2 Klein 4B (Black Forest Labs, 2025; 2026) is a four-step distilled flow model; we run it at 512×512 without classifier-free guidance. SD3.5 Medium (Esser et al., 2024) uses 32 FlowMatch Euler steps (Lipman et al., 2023) with classifier-free guidance 7 (Ho & Salimans, 2022) at 512×512 in BF16. FLUX.1-dev (Black Forest Labs, 2024) uses 32 steps with distilled guidance 3.5 at 512×512 in BF16. Infinity-2B (Han et al., 2025) generates 13 residual bit-token scales at 1024×1024 with classifier-free guidance 3, top-k 900 and top-p 0.97. For every request we cache eight native candidates, that is, eight initial noises or token seeds with their full native trajectories. All selectors use the first six. No candidate is cloned or modified: survivors continue under the unchanged native sampler, and the request returns the completed survivor with the highest final ImageReward (Xu et al., 2023). PickScore divided by 26 and CLIP ViT-L/14 image–text cosine (Kirstain et al., 2023; Radford et al., 2021) serve as additional quality judges and never enter selection.

B.2 PREDICTED-CLEAN PREVIEWS

Progressive selection needs an image-like observation before a trajectory is complete, and we call it a *preview*. A preview is decoded and scored only to choose survivors; it never replaces the candidate's latent or token state. The three flow backbones use a linear path on which the model predicts a velocity $v_\theta(x_\sigma, \sigma)$, and the preview is the clean-latent estimate

$$\widehat{x}_0(x_\sigma, \sigma) = x_\sigma - \sigma\, v_\theta(x_\sigma, \sigma), \tag{7}$$

the velocity form of Tweedie's posterior-mean denoiser (Efron, 2011). The velocity is the one the sampler computes anyway, so the preview costs no extra transformer call; decoding and scoring are real overhead and are included in the measured runtime. The survivor continues from the native post-update state $x_{\sigma'} = x_\sigma + (\sigma' - \sigma)v_\theta$, not from $\widehat{x}_0$. Infinity already maintains a coarse reconstruction. After a scale, each decoded residual is upsampled to the final latent grid and added to the accumulated code $h_{\le s}$, and the preview is the VAE decoding of $h_{\le s}$. Finer residuals that have not been generated contribute zero; no future token is sampled or imputed. Table 4 lists the construction per backbone. All metrics are computed on completed images, never on previews.

Table 4: Preview construction. Decoding is used only for scoring; survivors continue from their own native state.

Backbone	Checkpoint used by the cut	Preview passed to ImageReward and DINOv2
FLUX.2 Klein 4B	after step 2 of 4	VAE decoding of $x_\sigma - \sigma v_\theta$
SD3.5 Medium	after step 16 of 32	VAE decoding of $x_\sigma - \sigma v_\theta$
FLUX.1-dev	after step 16 of 32	VAE decoding of $x_\sigma - \sigma v_\theta$
Infinity-2B	after scale 11 of 13 (40×40 tokens)	VAE decoding of the accumulated residual code $h_{\le s}$

B.3 FEATURES AND CALIBRATION CONSTANTS

Each preview is embedded with DINOv2-L (Oquab et al., 2024); we use the normalised class-token feature and cosine distances $d_{ij} = 1 - f_i^\top f_j$. Three constants per backbone are estimated once on that backbone's development prompts, from eight-candidate pools at the cut. The reward scale σ_r is the mean over requests of the within-request standard deviation of preview ImageReward. The typical distance $\bar{d}$ is the median distance between distinct previews of a request, and the kernel width is $\tau = \bar{d}/4$ for the three flow backbones. On Infinity-2B, τ is a quarter of the median over requests of each request's median preview distance, 0.0260 against $\bar{d}/4 = 0.0264$. A final-reward scale σ_R, computed in the same way on completed images, is used only by the oracle diagnostic of Appendix F.6 and by density on the final images (Appendix E.4). All constants are carried unchanged to every evaluation set and are never refitted. Table 5 lists them together with the strengths.

B.4 QUALITY TOLERANCES AND STRENGTHS

Quality tolerances. To express quality losses on a human scale, we measure how often people prefer the higher-scoring image of a pair as a function of the score gap. ImageRewardDB (Xu et al., 2023) lists $55{,}247$ images with human rankings within each prompt. Removing $1{,}723$ empty files and the 275 prompts left with fewer than two images leaves $53{,}524$ images, which we score with the same ImageReward (model v1.0, package 1.5), PickScore/26 and CLIP ViT-L/14 code that selection uses. Every within-prompt pair is oriented by the model score, the human signal is the annotator's ranking with ties counted as one half, and intervals come from $2{,}000$ bootstrap resamples of prompts. We use only the validation and test splits: ImageReward was trained on the train split, and over all splits its agreement with the annotators rises to 73.0%. For each judge we fit a logistic link $P = \sigma(s\Delta)$ between the score gap Δ and the probability P that people prefer the higher-scoring image. The scale s is 0.804 [0.732, 0.881] for ImageReward on $14{,}521$ pairs from 878 prompts, 17.23 [15.39, 19.11] for PickScore/26 and 9.04 [7.26, 10.68] for CLIP on $14{,}532$ pairs; excluding ties, the higher-scoring image is preferred in 64.8%, 61.8% and 55.6% of pairs. A judge's *tolerance* is the gap at which the link predicts 51% preference, $\ln(0.51/0.49)/s$: 0.0498 ImageReward, 0.00232 PickScore/26 and 0.00442 CLIP. We fit the link rather than use the one implied by ImageReward's training loss, $\sigma(1.033\Delta)$, because ImageReward was trained on exactly

Table 5: Calibration constants and strengths at the cut. σ_r: preview reward scale; τ: kernel width; $\bar{d}$: typical preview distance used by max-sum; σ_R: final-reward scale (oracle and final-image rule only); β: density strength; λ: max-sum strength.

Backbone	σ_r	τ	$\bar{d}$	σ_R	β	λ
FLUX.2 Klein 4B	0.1851	0.0288	0.1152	0.1952	$1/32$	1.6
SD3.5 Medium	0.3442	0.0574	0.2297	0.2617	$1/16$	0.8
FLUX.1-dev	0.3237	0.0676	0.2704	0.3018	$1/4$	0.8
Infinity-2B	0.1802	0.0260	0.1056	0.1792	$1/32$	6.4

the within-prompt pairs that annotators did not tie, $155{,}684 - 18{,}792 = 136{,}892$, so that link carries no information about small gaps, where the tolerances lie. CLIP was not trained on preference pairs, so its scale is an empirical fit. Figure 7 shows the measured curve for ImageReward: over all pairs whose gap is at most δ, the interval of human preference contains 50% for every δ up to 0.2, four times the tolerance. A standard power analysis (80% power, $\alpha = 0.05$) gives a sense of scale: telling that preference from chance takes about $20{,}000$ paired judgements, and by the same analysis the largest ImageReward loss of our selectors on the main benchmark, 0.029 (predicted preference 50.6%), would take about $58{,}000$, against about 610 to $1{,}100$ for the gain of reward-only pruning over native sampling. These counts are approximate, since they treat a mean gap as the gap of one pair.

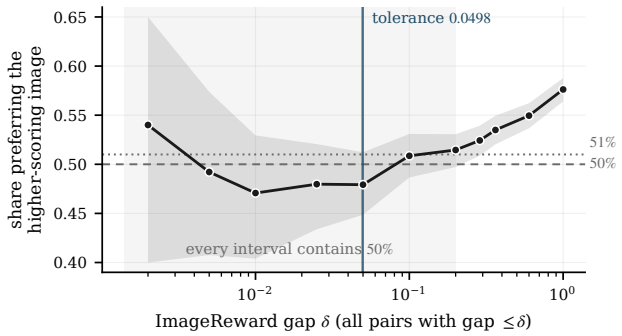

Figure 7: What an ImageReward gap is worth to people. For each gap δ, the share of held-out ImageRewardDB pairs with gap at most δ in which the annotator prefers the higher-scoring image (ties count one half; $14{,}521$ pairs from 878 prompts), with 95% prompt-bootstrap intervals. The vertical line is the ImageReward tolerance, the gap at which the fitted link predicts 51% preference (dotted); over the shaded range of gaps, up to 0.2, every interval contains 50%.

How strengths were chosen. One rule, applied per backbone and per selector on development data, fixed every strength before the main benchmark was generated. It takes the strongest setting whose estimated loss against same-schedule reward-only pruning stays below 70% of each tolerance on every development bank, whose one-sided 97.5% bound stays within the tolerance, and whose DreamSim gain has a positive lower bound. On Infinity we used one grid step weaker than the rule's choice ($1/32$ instead of $1/64$), because at the rule's choice the PickScore bound already used 99% of its tolerance on a development bank. On the main benchmark (Section 6.2) every estimated loss stays within its tolerance. The corrected bounds reported there are one-sided 98.75% bounds, Bonferroni-corrected over the four backbones; all stay within the tolerances except density's PickScore bounds on SD3.5 and FLUX.1-dev, at 128% and 103% of the tolerance. Figure 3 shows the full strength curves.

B.5 SCHEDULE AND BUDGET

We count cost as denoising steps summed over all candidates, in units of one native trajectory. The $6 \to 2$ schedule runs six candidates to the midpoint and two to the end, so it costs $6 \cdot \frac{1}{2} + 2 \cdot \frac{1}{2} = 4$ units, the same as best-of-4. Preview decoding, reward scoring and feature encoding are outside this count and inside the measured runtime. For Infinity, whose scales have very unequal cost, we use a per-scale multiply–accumulate proxy $c_s = t_s(12D^2 + 2D\sum_{u \le s} t_u)$ with t_s tokens at scale $s = 1, \dots, 13$ and width $D = 2048$. Its $6 \to 3$ cut after scale 11 then costs $(6\sum_{s \le 11} c_s + 3\sum_{s > 11} c_s)/\sum_s c_s = 3.93$ units.

Why a single cut at the midpoint. Flow-model previews become informative only late in the trajectory. The request-demeaned Pearson correlation between a candidate's preview reward and its final reward, measured on development prompts, is 0.11 on Klein and 0.35 on SD3.5 at the quarter checkpoint. At the midpoint it rises to 0.63 and 0.72 (FLUX.1-dev: 0.46 at the quarter checkpoint, 0.87 at the midpoint). In a Gaussian model of preview and final scores, halving a pool of six costs no final reward to leading order only when this correlation exceeds $\sqrt{\log 2 / \log 6} = 0.62$. A quarter-point cut therefore discards candidates on almost no information, while the midpoint preview is accurate enough to halve the pool about once. We therefore prune once, at the midpoint. On Infinity the preview after scale 11 already reaches 0.96, so we cut there, before the two finest scales. A later replay of five earlier banks per backbone (general, DrawBench, DIMCIM and open-ended prompts) under reward-only pruning supports this choice: $6 \to 2$ is within 0.0019 (Klein) and 0.0050 (SD3.5) ImageReward of the best of nine equal-cost schedules on average, against about 0.021 and 0.009 for the halving schedule $8 \to 4 \to 2$, and moving density to the reward-optimal schedule would raise ImageReward by about 0.004 but lower DreamSim by 0.0018 (Klein) and 0.0036 (SD3.5). Appendix E.3 lets every method choose its own schedule.

B.6 FULL ALGORITHM AND RUNTIME

Table 6 lists one request of either selector, including the champion-required variants used only in the interventions of Section 6.3.

Table 6: Champion-free progressive selection for one request. Step 4 differs between the two instances; step 4c is used only by the interventions.

Step	Inputs: prompt c; pool size N; survivors k; checkpoint t; constants $\sigma_r, \tau, \bar{d}$; strength β (density) or λ (max-sum)
1	Draw N native initial states once and run all N trajectories to checkpoint t with the native sampler.
2	For each candidate, form the preview (Eq. 7 or the accumulated code), decode it, and compute $v_i = r_i/\sigma_r$ from its ImageReward r_i and its unit DINOv2 feature f_i.
3	Compute $d_{ij} = 1 - f_i^\top f_j$ for all pairs in the pool.
4a	Density: $n_i = \sum_j \exp(-d_{ij}/\tau)$ over the whole pool, $j = i$ included; $\ell_i = \beta v_i - \log n_i$; keep the k largest ℓ_i (ties by candidate index).
4b	Exact max-sum: keep the maximiser of $J(S)$ in Eq. (3) over all $\binom{N}{k}$ subsets of size k.
4c	Champion required: keep $a = \arg\max_i v_i$; density adds the $k-1$ largest ℓ_i among the others, and max-sum maximises J over the subsets that contain a.
5	Continue the survivors along their own native trajectories to the end; never clone or restart a candidate.
6	Score the completed survivors with ImageReward and return the one with the highest final reward.

Density needs $O(N^2)$ distance arithmetic and one sort; exact max-sum enumerates $\binom{N}{k}$ subsets, which is 15 for $6 \to 2$. For pools of up to 16 candidates, either selection step takes at most 0.13 ms per request (Appendix G). Table 7 reports end-to-end wall-clock time and peak memory. Density costs 2.638 s per request against 2.582 s for reward-only pruning on Klein ($+2.2\%$), 8.975 s against 8.917 s on SD3.5 ($+0.7\%$) and 11.72 s against 11.68 s on Infinity-2B ($+0.3\%$); FLUX.1-dev was not timed. Feature encoding adds about 1.1 GiB of peak memory.

Table 7: Warm wall-clock time per request and peak GPU memory, one request at a time on an exclusive GPU. Klein and SD3.5: 30 development prompts; Infinity-2B: 12 prompts on an RTX 4090D. Times include prompt encoding, generation, preview decoding, feature encoding, reward scoring and image I/O. Infinity uses its $6 \to 3$ cut at $3.93\times$ under the multiply–accumulate proxy; its prototype advances one candidate per transformer call, so its pruning methods pay a per-call overhead that batching would remove.

		FLUX.2 Klein 4B		SD3.5 Medium		Infinity-2B
Method	Budget	s↓	GiB↓	s↓	GiB↓	s↓
Native sampling	$1\times$	0.705	20.01	2.310	17.67	2.37
Best-of-4	$4\times$	2.172	22.23	8.689	20.84	9.11
Reward-only pruning, $6 \to 2$	$4\times$	2.582	22.23	8.917	20.84	11.68
Density, $6 \to 2$	$4\times$	2.638	23.35	8.975	21.97	11.72

C Evaluation protocol

This appendix describes the prompt banks, the metrics, and how intervals and exact expectations are computed. Every number in Section 6 and in the other appendices follows these rules unless stated otherwise.

C.1 Prompt banks

Development prompts. Strengths, calibration constants and the schedule were chosen on development prompts only. For Klein and SD3.5 these were the first seed run of an earlier 200-prompt bank, 100 prompts from the general set described below and 100 self-authored open-ended prompts, and, while their strengths were being settled, further DrawBench and PartiPrompts samples. FLUX.1-dev used 60 and 100 DIMCIM prompts (Teotia et al., 2025), and Infinity-2B used three development banks. No development prompt appears in the main benchmark, in the second bank or in a further suite on which the same backbone is evaluated. The earlier bank was run a second time with independent seeds, and the large-pool replay of Appendix G, which pairs the two runs, is the one analysis that reuses development prompts.

Main benchmark. The main benchmark consists of 200 official PartiPrompts rows (Yu et al., 2022) that no earlier run had used. They were drawn with a fixed seed and checked for zero overlap, exact and punctuation-insensitive, against every earlier prompt list. The same prompts serve all four backbones. Each prompt receives eight independent requests, and each request caches eight native candidates. The benchmark was generated after all strengths had been fixed, and it provides the main results of Section 6.2. The mode definition of Section 6.4 was fixed before any outcome of this benchmark was examined. We later reused this benchmark to develop the mechanism analysis of Sections 4 and 6.3; the second bank then tests that analysis on prompts it had not used. The human study of Appendix I uses its SD3.5 and Klein images.

Second bank. The second bank holds 200 further PartiPrompts rows, disjoint from every earlier prompt, generated on Klein and SD3.5 with the same design. It was created to evaluate independently tuned baselines (Appendix E.3), whose configurations were fixed before generation. After its images existed, and before any of its outcomes were examined, we fixed the champion interventions, their strength grids, the bootstrap and four free-versus-required contrasts: density and exact max-sum at their development strengths on each backbone (Table 23). Apart from these contrasts and the tuned baselines, every analysis of the second bank was designed after its outcomes had been seen.

Further prompt suites. Two further suites test the main results outside PartiPrompts (Appendix D.3). *DIMCIM* holds 100 prompts of the coarse DIMCIM set (Teotia et al., 2025), up to four for each of 30 object categories; its intervals resample prompts within categories. *General* draws on the 500 general prompts assembled by Jeon et al. (2025) and also used by Li et al. (2026), a mix of MS-COCO captions (Lin et al., 2014), Lexica and Tuxemon prompts and GPT-4-generated compositional prompts; it holds the 300 of them that belong neither to the development prompts nor

to the earlier bank. No backbone's strength, setting or calibration constant was chosen on a suite it is evaluated on. Klein and SD3.5 were run on both suites before the development strengths were fixed, with the same generator, sampler settings and eight cached candidates per request as the main benchmark, and these banks had been analysed at earlier strengths; selection at the development strengths is an exact replay of their cached candidates. FLUX.1-dev and Infinity-2B were run on the general prompts with new seeds after all strengths had been fixed, using the generators of the main benchmark. They used the DIMCIM prompts for development and are therefore not evaluated on them.

C.2 METRICS

Quality is the mean over delivered images of ImageReward, PickScore/26 and CLIP image–text cosine. DreamSim (Fu et al., 2023) embeds an image with an ensemble of DINO, CLIP and OpenCLIP encoders tuned to human judgements of image similarity, and the DreamSim distance between two images is one minus the cosine similarity of their embeddings. Output diversity is this distance averaged over the 28 pairs of a prompt's eight delivered images and then over prompts, so higher values mean more diverse outputs. To measure content modes, we cluster, for each prompt, all 64 cached final images (8 requests $\times$ 8 candidates) into eight clusters by average-linkage clustering on DreamSim distance. The clusters are fitted once per prompt and shared by all policies; they are a representation-based proxy for content modes, not annotated categories. *Distinct modes* is the expected number of different clusters among the eight delivered images. *Rare-mode share* is the expected fraction of delivered images that fall in clusters holding at most one eighth of the prompt's final images. Native sampling delivers the first candidate of each request; its distinct modes and rare-mode share are exact expectations under uniform choice among the request's candidates. We also report the champion-exclusion rate (the share of requests whose survivors omit the preview champion; "Dropped" in the tables), the winner-change rate (the share whose delivered image differs from reward-only pruning's) and a DINO-free cosine distance (Appendix D.4).

C.3 INTERVALS, EXACT EVALUATION AND REWARD MATCHING

Prompt bootstrap. Intervals resample prompts 10,000 times within strata: PartiPrompts categories with weights proportional to their counts, or the general and open halves of the earlier bank. The eight requests of a prompt always move together, so the 28 pairwise distances are never treated as independent observations. Intervals are nominal two-sided 95% unless a caption says otherwise, and we call a difference *resolved* when its interval excludes zero. The four second-bank free-versus-required contrasts (Table 24) use two-sided 98.75% intervals, a Bonferroni correction for four comparisons. All intervals are conditional on the fixed policies and, where used, on fitted mixture weights.

Exact evaluation. Every candidate's preview reward, preview feature, final image, final reward and evaluation embeddings are cached. Deterministic selectors are therefore replayed without generating anything. For stochastic selectors we enumerate every survivor set with its probability, so reported values are exact expectations with no Monte Carlo error. Sampling k survivors without replacement from a pool C with positive weights w_i selects the unordered set S with probability

$$K(S) = \sum_{\sigma \in \mathrm{Perm}(S)} \prod_{l=1}^{k} \frac{w_{\sigma_l}}{\sum_{i \in C \setminus \{\sigma_1, \ldots, \sigma_{l-1}\}} w_i}. \tag{8}$$

A final rule $T(i \mid S)$ then gives each candidate a delivery probability $p_q(i) = \sum_{S \ni i} K(S)\, T(i \mid S)$ on request q. Two independent requests q and q' of the same prompt contribute $\sum_{i,j} p_q(i)\, p_{q'}(j)\, d(y_{qi}, y_{q'j})$ to expected output diversity. Distinct modes and rare-mode share are computed from the same delivery probabilities.

Reward matching. To compare two families at equal mean ImageReward, we sort each family's configurations by mean delivered reward. The two neighbouring configurations that bracket the target are mixed with the unique weight u that matches it. The mixture picks its configuration independently for each request, so its delivery law on request q is $(1-u)\, p_q^{(0)} + u\, p_q^{(1)}$, and its diversity includes the cross terms between the two configurations. Matched comparisons between selector

families never interpolate diversity scores linearly: for cosine features with mean embeddings μ_0 and μ_1, mixing adds $u(1-u)\|\mu_0-\mu_1\|^2$, which a straight line omits. Two exceptions are the best-of-N reference curve, which joins native sampling, best-of-4 and best-of-8 by straight lines in the plane of mean ImageReward and DreamSim, and the language-model study of Appendix H, which checks its linear interpolation against exact mixtures. Targets outside a family's reward range are left unmatched. "Matched at 25%, 50% or 75%" refers to positions within the reward range that two families share, ordered from low to high reward.

D FULL RESULTS

This appendix gives the main-benchmark scores behind Table 1 with their intervals, paired contrasts and mode counts, and repeats the comparison under other quality judges and on two prompt suites outside PartiPrompts. It then measures the delivered images in a representation that the selector never sees and with the Vendi score and LPIPS, and compares density with max-sum in content modes: at matched reward, where the difference comes from, and how it depends on the mode definition. Together these support the claims of Section 6.2 that both champion-free selectors raise output diversity on all four backbones, at a reward cost that the strength controls, and of Section 6.4 that density delivers rare modes more often than max-sum.

D.1 ABSOLUTE SCORES AND PAIRED CONTRASTS

Table 8 lists absolute scores with marginal intervals, and Table 9 lists paired differences from same-schedule reward-only pruning. Both selectors raise DreamSim on every backbone with paired intervals above zero, density by 0.0036 to 0.0184 and max-sum by 0.0049 to 0.0114. The price is paid mainly in ImageReward, 0.012 to 0.029 for either selector; PickScore/26 falls by at most 0.0022, and CLIP changes by less than 0.001. At these strengths the two selectors spend different amounts of reward, so their diversity values should not be compared directly; Table 15 compares them at matched reward.

Table 10 gives the paired intervals behind the ΔDS column of Table 1 for every row; only best-of-4 on Klein and Infinity-2B and greedy MMR on Infinity-2B have intervals that cover zero.

Content modes. Reward-only pruning also removes content modes. Among eight delivered images, native sampling covers 2.95, 3.13, 3.10 and 3.15 distinct modes on Klein, SD3.5, FLUX.1-dev and Infinity-2B. Reward-only pruning covers 2.61, 2.75, 2.82 and 3.01. Density returns to 3.09, 3.11, 3.12 and 3.21, and max-sum reaches 2.90, 2.92, 3.04 and 3.26. The rare-mode share tells the same story on the three diffusion backbones. There, reward-only pruning lowers it from about 20.5% under native sampling to 18.0%–19.6%, and density raises it to 24.1%–25.3%, above native sampling. On Infinity pruning does not lower it (22.5% against 21.0%), and density raises it to 25.9%.

D.2 OTHER QUALITY JUDGES

PickScore selects. Table 1 and the strength rule use ImageReward both to select and to judge. To separate the two roles, we replay the main benchmark's cached candidates on the three diffusion backbones with PickScore in ImageReward's place: PickScore scores the previews, the champion is the candidate with the best preview PickScore, and the delivered survivor is the one with the highest final PickScore. Candidates, the $6 \to 2$ cut, the DINOv2 features, τ and $\bar{d}$ are unchanged, and ImageReward and CLIP judge the delivered images without entering selection. We read density and max-sum at the strengths the development rule selects when applied to this bank itself, so these strengths are not held out; the gains do not hinge on them, since every strength of either selector raises output diversity with an interval above zero except the few weakest, which sit next to PickScore-only pruning. The second paragraph below keeps ImageReward selection and uses PickScore only to match quality. Infinity-2B is omitted because its cache has no preview PickScore.

Table 11 gives the differences from PickScore-only pruning. Both selectors raise output diversity and distinct modes on all three backbones, with every interval above zero, and density brings DreamSim back to about the level of native sampling (0.182, 0.247, 0.226 against 0.180, 0.249, 0.229 on Klein, SD3.5 and FLUX.1-dev). The independent judges lose at most 0.032 ImageReward, 64% of

Table 8: Absolute delivered scores with marginal 95% prompt-bootstrap intervals at the development strengths. Intervals are recomputed from absolute per-prompt scores. Paired comparisons appear in Table 9.

Backbone	Method	ImageReward↑	PickScore/26↑	CLIP↑	DreamSim↑
FLUX.2 Klein 4B	Native sampling	1.2056 [1.1180, 1.2893]	0.8773 [0.8714, 0.8829]	0.2839 [0.2779, 0.2897]	0.1800 [0.1705, 0.1897]
	Reward-only pruning	1.4118 [1.3403, 1.4786]	0.8832 [0.8773, 0.8888]	0.2868 [0.2807, 0.2928]	0.1671 [0.1587, 0.1756]
	Max-sum	1.3865 [1.3140, 1.4541]	0.8823 [0.8763, 0.8879]	0.2863 [0.2803, 0.2923]	0.1785 [0.1694, 0.1878]
	Density (ours)	1.3828 [1.3107, 1.4512]	0.8818 [0.8760, 0.8875]	0.2871 [0.2811, 0.2931]	0.1796 [0.1711, 0.1882]
SD3.5 Medium	Native sampling	1.1607 [1.0781, 1.2387]	0.8663 [0.8611, 0.8715]	0.2833 [0.2781, 0.2886]	0.2495 [0.2369, 0.2621]
	Reward-only pruning	1.4332 [1.3690, 1.4933]	0.8741 [0.8689, 0.8791]	0.2869 [0.2814, 0.2924]	0.2230 [0.2122, 0.2338]
	Max-sum	1.4213 [1.3566, 1.4818]	0.8734 [0.8682, 0.8784]	0.2864 [0.2809, 0.2917]	0.2321 [0.2210, 0.2433]
	Density (ours)	1.4059 [1.3402, 1.4673]	0.8719 [0.8667, 0.8769]	0.2867 [0.2813, 0.2920]	0.2414 [0.2309, 0.2520]
FLUX.1-dev	Native sampling	1.1578 [1.0660, 1.2430]	0.8845 [0.8791, 0.8899]	0.2739 [0.2683, 0.2795]	0.2290 [0.2172, 0.2414]
	Reward-only pruning	1.4412 [1.3735, 1.5036]	0.8926 [0.8872, 0.8980]	0.2796 [0.2740, 0.2854]	0.2089 [0.1982, 0.2198]
	Max-sum	1.4287 [1.3615, 1.4907]	0.8915 [0.8861, 0.8968]	0.2789 [0.2733, 0.2846]	0.2203 [0.2096, 0.2314]
	Density (ours)	1.4291 [1.3615, 1.4915]	0.8910 [0.8856, 0.8962]	0.2788 [0.2732, 0.2846]	0.2190 [0.2084, 0.2298]
Infinity-2B	Native sampling	1.0896 [0.9742, 1.1972]	0.8759 [0.8700, 0.8817]	0.2705 [0.2652, 0.2757]	0.1271 [0.1171, 0.1382]
	Reward-only pruning	1.3381 [1.2437, 1.4242]	0.8817 [0.8759, 0.8874]	0.2756 [0.2701, 0.2810]	0.1205 [0.1111, 0.1309]
	Max-sum	1.3105 [1.2141, 1.3988]	0.8810 [0.8751, 0.8867]	0.2750 [0.2696, 0.2804]	0.1254 [0.1160, 0.1358]
	Density (ours)	1.3243 [1.2295, 1.4106]	0.8809 [0.8751, 0.8865]	0.2752 [0.2698, 0.2805]	0.1241 [0.1148, 0.1344]

Table 9: Paired differences underlying Table 1. Intervals are nominal two-sided 95% stratified prompt bootstraps; their lower endpoints are one-sided 97.5% lower bounds. Each rule uses its development-selected strength and the same-schedule pruning reference.

Backbone	Comparison	ΔImageReward	ΔPickScore/26	ΔCLIP	ΔDreamSim
FLUX.2 Klein 4B	Density − pruning	-0.0290 [-0.0358, -0.0225]	-0.0014 [-0.0020, -0.0008]	+0.0003 [-0.0003, +0.0009]	+0.0125 [+0.0102, +0.0150]
	Max-sum − pruning	-0.0253 [-0.0310, -0.0197]	-0.0010 [-0.0016, -0.0003]	-0.0005 [-0.0011, +0.0002]	+0.0113 [+0.0089, +0.0140]
SD3.5 Medium	Density − pruning	-0.0274 [-0.0343, -0.0207]	-0.0022 [-0.0029, -0.0015]	-0.0002 [-0.0009, +0.0004]	+0.0184 [+0.0139, +0.0234]
	Max-sum − pruning	-0.0119 [-0.0162, -0.0078]	-0.0007 [-0.0012, -0.0002]	-0.0005 [-0.0011, -0.0000]	+0.0092 [+0.0065, +0.0120]
FLUX.1-dev	Density − pruning	-0.0121 [-0.0155, -0.0084]	-0.0017 [-0.0023, -0.0011]	-0.0008 [-0.0013, -0.0003]	+0.0101 [+0.0072, +0.0131]
	Max-sum − pruning	-0.0126 [-0.0157, -0.0094]	-0.0011 [-0.0017, -0.0006]	-0.0008 [-0.0014, -0.0002]	+0.0114 [+0.0087, +0.0142]
Infinity-2B	Density − pruning	-0.0138 [-0.0167, -0.0111]	-0.0008 [-0.0012, -0.0005]	-0.0004 [-0.0008, -0.0001]	+0.0036 [+0.0025, +0.0048]
	Max-sum − pruning	-0.0276 [-0.0345, -0.0219]	-0.0008 [-0.0013, -0.0003]	-0.0006 [-0.0011, -0.0002]	+0.0049 [+0.0032, +0.0066]

its tolerance, and 0.0012 CLIP, and PickScore/26 itself moves by at most 0.0002. PickScore-only pruning gains only 0.0006 to 0.0007 PickScore/26 over native sampling, under a third of that judge's tolerance. The champion is dropped on 25% to 49% of requests.

Table 10: Paired differences behind the ΔDS column of Table 1: output diversity minus reward-only pruning ($\times 10^3$) with 95% stratified prompt-bootstrap intervals, main benchmark; max-sum and density at their development strengths. k-DPP, and greedy MMR on FLUX.1-dev, are read on their curves at density's ImageReward; elsewhere greedy MMR is at its most diverse setting.

Method	Klein (4-step)	SD3.5	FLUX.1-dev	Infinity (AR)
Native sampling	+12.9 [8.7, 17.2]	+26.5 [20.8, 32.4]	+20.1 [15.1, 25.5]	+6.6 [3.4, 10.0]
Best-of-4	+1.5 [−1.1, 4.1]	+9.1 [5.8, 12.5]	+5.1 [2.1, 8.2]	+1.2 [−0.3, 3.0]
Greedy MMR	+4.9 [3.0, 6.9]	+6.7 [4.3, 9.0]	+5.6 [2.8, 7.9]	+0.4 [−0.5, 1.3]
k-DPP	+5.3 [3.8, 6.9]	+6.9 [4.8, 9.2]	+4.4 [2.9, 5.9]	+1.6 [0.8, 2.4]
Max-sum (ours)	+11.3 [8.9, 14.0]	+9.2 [6.5, 12.0]	+11.4 [8.7, 14.2]	+4.9 [3.2, 6.6]
Density (ours)	+12.5 [10.2, 15.0]	+18.4 [13.9, 23.4]	+10.1 [7.2, 13.1]	+3.6 [2.5, 4.8]

Table 11: Selection and delivery by PickScore: on the same main-benchmark candidates, $6 \to 2$ cut and DINOv2 features, PickScore scores the previews and picks the delivered survivor, and the champion is the preview-PickScore maximiser; ImageReward and CLIP are then independent judges. Entries are differences from PickScore-only pruning at the strengths the development rule selects on this bank itself, so not held out (density $\beta = 1/128$, $1/32$, $1/16$ and max-sum $\lambda = 3.2$, 1.6, 3.2 on Klein, SD3.5 and FLUX.1-dev), with 95% stratified prompt-bootstrap intervals for ImageReward, DreamSim and distinct modes. Champion dropped: share of requests whose preview champion is not kept. Infinity-2B is omitted because its cache has no preview PickScore.

Backbone	Selector	ΔImageReward	ΔPickScore /26	ΔCLIP	ΔDreamSim	ΔDistinct modes	Champion dropped
Klein	Density	−0.032 [−0.046, −0.020]	−0.0002	−0.0006	+0.016 [0.013, 0.019]	+0.70 [0.55, 0.85]	49%
	Max-sum	−0.031 [−0.046, −0.017]	−0.0001	−0.0009	+0.014 [0.012, 0.017]	+0.41 [0.27, 0.54]	44%
SD3.5	Density	−0.027 [−0.042, −0.012]	−0.0001	−0.0001	+0.024 [0.019, 0.030]	+0.62 [0.48, 0.77]	42%
	Max-sum	−0.026 [−0.037, −0.015]	−0.0001	−0.0002	+0.011 [0.007, 0.014]	+0.26 [0.15, 0.36]	25%
FLUX.1-dev	Density	−0.022 [−0.034, −0.010]	−0.0001	−0.0008	+0.018 [0.015, 0.022]	+0.53 [0.41, 0.66]	34%
	Max-sum	−0.022 [−0.035, −0.009]	−0.0001	−0.0012	+0.017 [0.013, 0.021]	+0.46 [0.33, 0.60]	35%

PickScore matches quality. Strengths were chosen and curves matched with ImageReward, which also selects. Table 12 repeats the matched comparisons with PickScore as the quality axis. Against the best-of-N curve the result does not depend on the judge. At their own PickScore both champion-free selectors deliver more DreamSim than the curve through native sampling, best-of-4 and best-of-8: density by 0.0056 [0.0041, 0.0071] on the main benchmark and 0.0081 [0.0057, 0.0105] on the second bank, and max-sum by 0.0054 and 0.0044. Reward-only pruning lies on or below the curve under both judges. Density also delivers more DreamSim than reward sampling at every matched PickScore.

Between the two selectors, the judge decides how their difference shows. At matched ImageReward density delivers more rare modes (Table 15). At matched PickScore it instead keeps 0.005 to 0.016 more ImageReward on the main benchmark, with DreamSim 0.002 to 0.004 lower and the rare-mode share even; on the second bank its lead in distinct modes and rare-mode share remains (+2.8 points [1.4, 4.2] at the development strength). Neither selector dominates the other across both judges; under either judge both lie above the best-of-N curve, and reward-only pruning does not.

D.3 Two further prompt suites

Two further suites test the main results outside PartiPrompts: 100 DIMCIM prompts (Teotia et al., 2025), each asking for one object category, on Klein and SD3.5, and 300 general prompts on all

Table 12: Matching on PickScore instead of ImageReward. (a) Density minus max-sum at matched mean PickScore/26, at density's development strength (Dev.) and at 25, 50 and 75% of the PickScore range the two selectors share, pooled over backbones per prompt; the last column compares density with reward sampling at the same PickScore. Tail mass uses the cluster-free rarity of Table 17 ($h = m_p$, rarest $1/8$). †No max-sum strength reaches density's PickScore on Infinity-2B, so the density − max-sum columns of this row pool Klein, SD3.5 and FLUX.1-dev. (b) Delivered DreamSim minus the best-of-N curve (native, best-of-4, best-of-8, interpolated) at the method's own mean ImageReward or PickScore/26, at the development strengths. 95% stratified prompt-bootstrap intervals; bold intervals exclude zero.

(a) Density − max-sum at matched mean PickScore

Bank	Matched at	ΔImageReward	ΔDreamSim	ΔDistinct modes	ΔRare-mode share (pp)	ΔTail mass, rarest $1/8$ (pp)	ΔDreamSim vs reward sampling
Main	Dev.†	**+0.015** [+0.010, +0.021]	−0.0021 [−0.0043, +0.0001]	+0.058 [−0.019, +0.137]	+0.9 [−0.2, +2.1]	−0.3 [−1.2, +0.5]	**+0.0065** [+0.0051, +0.0080]
	25%	**+0.016** [+0.011, +0.021]	−0.0016 [−0.0035, +0.0003]	+0.037 [−0.027, +0.102]	**+1.0** [+0.1, +1.9]	+0.1 [−0.7, +0.8]	**+0.0076** [+0.0059, +0.0094]
	50%	**+0.013** [+0.010, +0.017]	**−0.0037** [−0.0052, −0.0022]	−0.037 [−0.094, +0.021]	−0.1 [−1.0, +0.7]	−0.4 [−1.0, +0.2]	**+0.0045** [+0.0032, +0.0059]
	75%	**+0.005** [+0.004, +0.007]	**−0.0030** [−0.0040, −0.0021]	−0.030 [−0.070, +0.010]	−0.4 [−1.0, +0.2]	**−0.6** [−1.0, −0.1]	**+0.0024** [+0.0016, +0.0032]
Second	Dev.	+0.004 [−0.002, +0.011]	+0.0006 [−0.0021, +0.0033]	**+0.125** [+0.030, +0.224]	**+2.8** [+1.4, +4.2]	+0.3 [−0.9, +1.6]	**+0.0087** [+0.0065, +0.0110]
	25%	+0.003 [−0.004, +0.010]	+0.0014 [−0.0012, +0.0040]	**+0.125** [+0.036, +0.219]	**+2.5** [+1.1, +3.8]	+0.3 [−0.9, +1.4]	**+0.0095** [+0.0072, +0.0117]
	50%	**+0.004** [+0.000, +0.008]	−0.0005 [−0.0027, +0.0017]	+0.060 [−0.020, +0.141]	**+1.2** [+0.1, +2.3]	−0.0 [−1.0, +0.9]	**+0.0053** [+0.0036, +0.0070]
	75%	+0.002 [−0.001, +0.004]	**−0.0017** [−0.0033, −0.0000]	−0.010 [−0.074, +0.055]	+0.1 [−0.7, +1.0]	−0.7 [−1.5, +0.1]	**+0.0022** [+0.0008, +0.0036]

(b) DreamSim above the best-of-N curve at the method's own mean judge score

		Density		Max-sum		Reward-only pruning	
Bank	Backbone	ImageReward	PickScore	ImageReward	PickScore	ImageReward	PickScore
Main	Klein	**+0.0097** [+0.0066, +0.0128]	**+0.0094** [+0.0062, +0.0127]	**+0.0087** [+0.0059, +0.0116]	**+0.0094** [+0.0064, +0.0121]	−0.0013 [−0.0039, +0.0010]	−0.0008 [−0.0029, +0.0010]
	SD3.5	**+0.0087** [+0.0045, +0.0130]	**+0.0071** [+0.0028, +0.0115]	+0.0007 [−0.0028, +0.0041]	+0.0026 [−0.0014, +0.0066]	**−0.0073** [−0.0104, −0.0043]	**−0.0039** [−0.0075, −0.0002]
	FLUX.1-dev	**+0.0063** [+0.0031, +0.0095]	+0.0034 [−0.0001, +0.0069]	**+0.0076** [+0.0045, +0.0105]	**+0.0059** [+0.0025, +0.0093]	**−0.0030** [−0.0053, −0.0007]	**−0.0029** [−0.0054, −0.0005]
	Infinity-2B	**+0.0027** [+0.0012, +0.0043]	**+0.0025** [+0.0008, +0.0041]	**+0.0038** [+0.0020, +0.0056]	**+0.0038** [+0.0020, +0.0055]	−0.0007 [−0.0020, +0.0006]	−0.0006 [−0.0019, +0.0007]
	Pooled	**+0.0068** [+0.0053, +0.0084]	**+0.0056** [+0.0041, +0.0071]	**+0.0052** [+0.0039, +0.0064]	**+0.0054** [+0.0040, +0.0068]	**−0.0031** [−0.0043, −0.0019]	**−0.0021** [−0.0034, −0.0008]
Second	Klein	**+0.0081** [+0.0051, +0.0109]	**+0.0066** [+0.0040, +0.0093]	**+0.0060** [+0.0036, +0.0085]	**+0.0049** [+0.0024, +0.0075]	**−0.0025** [−0.0047, −0.0005]	−0.0013 [−0.0034, +0.0010]
	SD3.5	**+0.0134** [+0.0098, +0.0170]	**+0.0096** [+0.0056, +0.0136]	**+0.0069** [+0.0033, +0.0103]	**+0.0039** [+0.0002, +0.0077]	−0.0018 [−0.0048, +0.0012]	−0.0032 [−0.0068, +0.0004]
	Pooled	**+0.0107** [+0.0083, +0.0131]	**+0.0081** [+0.0057, +0.0105]	**+0.0064** [+0.0043, +0.0085]	**+0.0044** [+0.0022, +0.0067]	**−0.0022** [−0.0040, −0.0004]	**−0.0022** [−0.0044, −0.0000]

four backbones (Appendix C.1 gives their provenance). Table 13 evaluates both at the development strengths of Table 5.

The results of Sections 6.2 and 6.3 carry over. Reward-only pruning again lowers delivered DreamSim on every suite and backbone, most on SD3.5 (from 0.270 to 0.245 on DIMCIM). Both champion-free selectors raise it again, with paired intervals above zero in all twelve comparisons: density by 0.0022 to 0.0183 at an ImageReward cost of 0.009 to 0.032, and max-sum by 0.0044 to 0.0109 at a cost of 0.005 to 0.037. Both also deliver more distinct modes than reward-only pruning, again with all twelve intervals above zero, and density raises the rare-mode share, by 3.0 to 7.6 percentage points with every interval above zero. Requiring the preview champion lowers delivered DreamSim for both selectors on every suite and backbone, with all twelve intervals above zero, and 43.5% to 72.9% of requests are locked.

Against max-sum at matched ImageReward, density again matches mean DreamSim: pooled over backbones, the difference is +0.0004 [−0.0024, +0.0032] on DIMCIM and −0.0009 [−0.0020, +0.0001] on the general prompts. Its rare-mode advantage holds on both suites when pooled over their two and four backbones: +3.0 percentage points [+1.5, +4.4] on DIMCIM, where both backbones resolve it on their own, and a smaller +0.9 [+0.2, +1.5] on the general prompts, where of the four backbones only Infinity-2B resolves it on its own. Its estimate is positive on all six suite–backbone pairs. On DIMCIM density also delivers 0.16 [0.05, 0.27] more distinct modes; on the general prompts the distinct-mode difference is +0.02 [−0.02, +0.07].

Table 13: Two further prompt suites at the development strengths of Table 5: 100 DIMCIM prompts (Klein and SD3.5) and 300 general prompts (Klein, SD3.5, FLUX.1-dev and Infinity-2B). No backbone's strength or setting was chosen on a suite it is evaluated on, and every value is an exact replay of cached candidates. (a) Delivered ImageReward (IR) and DreamSim (DS), as in Table 1. (b) Paired DreamSim differences of each champion-free selector from reward-only pruning and from its champion-required version, both at the development strength, and density minus max-sum in rare-mode share (percentage points) at density's mean ImageReward. Brackets: 95% prompt-bootstrap intervals; DIMCIM resamples prompts within object categories.

(a) Absolute scores

	DIMCIM (100)				General (300)							
	Klein		SD3.5		Klein		SD3.5		FLUX.1-dev		Infinity-2B	
Method	IR↑	DS↑	IR↑	DS↑	IR↑	DS↑	IR↑	DS↑	IR↑	DS↑	IR↑	DS↑
Native sampling (1×)	1.032	0.195	0.979	0.270	0.988	0.166	0.877	0.243	0.838	0.232	1.008	0.120
Best-of-4	1.245	0.187	1.238	0.252	1.203	0.161	1.163	0.230	1.169	0.220	1.224	0.116
Reward-only pruning	1.251	0.185	1.242	0.245	1.215	0.162	1.181	0.225	1.208	0.216	1.265	0.116
Max-sum (ours)	1.214	0.195	1.232	0.255	1.189	0.169	1.176	0.232	1.199	0.222	1.237	0.120
Density (ours)	1.221	0.192	1.214	0.264	1.182	0.167	1.155	0.235	1.199	0.221	1.251	0.118

(b) Paired differences

		ΔDS against pruning		ΔDS free − required		ΔRare (pts)
Suite	Backbone	Density	Max-sum	Density	Max-sum	Density − max-sum
DIMCIM	Klein	+0.0076 [+0.0050, +0.0102]	+0.0109 [+0.0082, +0.0135]	+0.0053 [+0.0029, +0.0078]	+0.0063 [+0.0039, +0.0086]	+3.1 [+0.6, +5.5]
	SD3.5	+0.0183 [+0.0141, +0.0228]	+0.0093 [+0.0063, +0.0123]	+0.0109 [+0.0078, +0.0142]	+0.0037 [+0.0018, +0.0057]	+2.8 [+0.9, +4.7]
General	Klein	+0.0055 [+0.0043, +0.0069]	+0.0067 [+0.0050, +0.0084]	+0.0032 [+0.0021, +0.0043]	+0.0039 [+0.0026, +0.0053]	+0.4 [-1.0, +1.9]
	SD3.5	+0.0102 [+0.0080, +0.0126]	+0.0064 [+0.0046, +0.0082]	+0.0067 [+0.0050, +0.0084]	+0.0025 [+0.0015, +0.0035]	+1.3 [-0.1, +2.7]
	FLUX.1-dev	+0.0053 [+0.0039, +0.0068]	+0.0060 [+0.0043, +0.0078]	+0.0026 [+0.0017, +0.0035]	+0.0026 [+0.0014, +0.0038]	+0.5 [-0.5, +1.6]
	Infinity-2B	+0.0022 [+0.0014, +0.0029]	+0.0044 [+0.0032, +0.0056]	+0.0016 [+0.0010, +0.0022]	+0.0039 [+0.0028, +0.0050]	+1.2 [+0.0, +2.3]

D.4 DINO-FREE AND OTHER DIVERSITY MEASURES

DreamSim combines DINO, CLIP and OpenCLIP features, and DINOv2 also defines the selector's geometry, so the headline metric is not independent of the selector. We therefore re-embedded every cached final image of the main benchmark with OpenCLIP ViT-g-14 (LAION-2B), which enters neither the selector nor any quality judge, and recomputed output diversity as mean cosine distance in that space. We also score the same eight delivered images per prompt with two measures common in work on generative diversity: the Vendi score (Friedman & Dieng, 2023), the exponential of the entropy of the eigenvalues of the normalised similarity matrix, interpreted as the effective number of distinct images, here with a DreamSim cosine kernel; and LPIPS (Zhang et al., 2018), a learned perceptual distance on AlexNet features. Selection is unchanged: the same rewards, kernel, strengths and final rules are evaluated on the same prompts. Table 14 shows that both selectors raise output diversity over reward-only pruning under all four measures on every backbone, with all 32 intervals above zero. Reward-only pruning lowers output diversity in 15 of the 16 backbone–measure pairs; under LPIPS on Infinity-2B its interval covers zero. The measures have different scales, so we compare signs and orderings, not magnitudes. The Vendi score moves little in absolute terms because the eight images of one prompt lie close together in DreamSim space: native sampling reaches only 1.76 to 2.53 effective images out of eight.

Table 14: The same delivered images under four diversity measures on the main benchmark, at the development strengths. Selection is unchanged: the same DINOv2-L features, reward, strengths and selection rules. Only the measure of the eight delivered images changes: DreamSim distance; OpenCLIP ViT-g-14 cosine distance, which appears in neither the selector nor any scorer; the Vendi score with a DreamSim cosine kernel, the effective number of distinct images (native sampling: 1.76 to 2.53 of 8); and LPIPS (AlexNet, 512×512 inputs). For all four, higher means more diverse. Native sampling delivers the first candidate of each request, as in Table 1. Paired 95% stratified prompt-bootstrap intervals, same seeds and replicate count as the main benchmark.

Backbone	Contrast	ΔDreamSim	ΔViT-g-14	ΔVendi	ΔLPIPS
FLUX.2 Klein 4B	Reward-only pruning − native	−0.0129 [−0.0172, −0.0087]	−0.0104 [−0.0144, −0.0064]	−0.081 [−0.107, −0.056]	−0.0060 [−0.0112, −0.0011]
	Density − pruning	+0.0125 [+0.0102, +0.0150]	+0.0082 [+0.0058, +0.0108]	+0.076 [+0.062, +0.091]	+0.0089 [+0.0053, +0.0133]
	Max-sum − pruning	+0.0113 [+0.0089, +0.0140]	+0.0076 [+0.0052, +0.0101]	+0.070 [+0.056, +0.086]	+0.0057 [+0.0023, +0.0095]
SD3.5 Medium	Reward-only pruning − native	−0.0265 [−0.0324, −0.0208]	−0.0157 [−0.0201, −0.0113]	−0.182 [−0.222, −0.144]	−0.0148 [−0.0212, −0.0084]
	Density − pruning	+0.0184 [+0.0139, +0.0234]	+0.0093 [+0.0063, +0.0127]	+0.119 [+0.092, +0.148]	+0.0132 [+0.0080, +0.0183]
	Max-sum − pruning	+0.0092 [+0.0065, +0.0120]	+0.0032 [+0.0008, +0.0057]	+0.061 [+0.044, +0.080]	+0.0071 [+0.0032, +0.0110]
FLUX.1-dev	Reward-only pruning − native	−0.0201 [−0.0255, −0.0151]	−0.0120 [−0.0166, −0.0076]	−0.125 [−0.162, −0.092]	−0.0079 [−0.0122, −0.0037]
	Density − pruning	+0.0101 [+0.0072, +0.0131]	+0.0071 [+0.0047, +0.0097]	+0.066 [+0.047, +0.085]	+0.0080 [+0.0051, +0.0110]
	Max-sum − pruning	+0.0114 [+0.0087, +0.0142]	+0.0076 [+0.0052, +0.0101]	+0.070 [+0.053, +0.088]	+0.0085 [+0.0059, +0.0113]
Infinity-2B	Reward-only pruning − native	−0.0066 [−0.0100, −0.0034]	−0.0073 [−0.0105, −0.0043]	−0.040 [−0.061, −0.020]	−0.0024 [−0.0062, +0.0012]
	Density − pruning	+0.0036 [+0.0025, +0.0048]	+0.0029 [+0.0019, +0.0040]	+0.021 [+0.015, +0.028]	+0.0058 [+0.0038, +0.0078]
	Max-sum − pruning	+0.0049 [+0.0032, +0.0066]	+0.0029 [+0.0014, +0.0045]	+0.028 [+0.018, +0.038]	+0.0055 [+0.0035, +0.0076]

Table 15: Density minus max-sum at matched mean ImageReward, pooled over backbones per prompt, with 95% prompt-bootstrap intervals. The first two rows keep two of six candidates (three on Infinity). Rare modes are clusters holding at most one eighth of the prompt's native final images. In the 16-candidate pools, the rare-mode gain with two survivors is 1.9 points [0.2, 3.6] larger than with six candidates from the same pools.

Setting	ΔDreamSim	ΔDistinct modes	ΔRare-mode share (pp)
Main benchmark (4 backbones)	+0.0005 [−0.0011, +0.0021]	+0.085 [+0.023, +0.147]	+1.6 [+0.7, +2.5]
Second bank (2 backbones)	+0.0017 [−0.0010, +0.0043]	+0.142 [+0.049, +0.239]	+3.1 [+1.7, +4.5]
16-candidate pools, $k = 2$	+0.0005 [−0.0020, +0.0030]	+0.098 [−0.003, +0.198]	+2.3 [+0.8, +3.8]
16-candidate pools, $k = 4$	+0.0004 [−0.0017, +0.0026]	+0.046 [−0.031, +0.122]	+0.8 [−0.2, +2.0]

D.5 Density and max-sum in content modes

Table 15 gives the comparison behind Section 6.4. At matched ImageReward, density and max-sum tie on mean DreamSim on both banks, while density delivers 0.085 and 0.142 more distinct modes and a rare-mode share higher by 1.6 and 3.1 percentage points, all with intervals above zero. Matching mean ImageReward leaves a small PickScore difference: density's PickScore/26 is lower than max-sum's by 0.00055 [0.00023, 0.00088] pooled on the main benchmark, 24% of that judge's tolerance (Appendix B.4), and by 0.00013 [−0.00037, +0.00061] on the second bank. Appendix D.2 matches on PickScore instead.

Where the advantage comes from. Figure 6 splits the rare-mode advantage of Table 15 by the number j' of pool candidates that lie in any rare cluster, the count that Proposition 2 conditions on. The rare-mode share counts delivery into any rare cluster, so the rare content of the proposition is the union of rare clusters. On the three backbones with two survivors, requests with $j' \geq 2$ (36%

of requests) carry 1.29 [0.48, 2.11] of the 1.69-point advantage on these backbones, a conditional advantage of 3.6 points [1.4, 5.9]. On the second bank they carry 2.37 [1.30, 3.46] of 3.13 points, a conditional advantage of 6.5 points [3.5, 9.4]. Requests with a single rare candidate carry the rest, 0.40 [−0.22, 1.02] and 0.75 [−0.01, 1.53] points.

On $j' \geq 2$ requests density keeps two rare survivors 9.3 [7.2, 11.7] and 7.9 [5.0, 11.1] points more often than max-sum. It does not drop the champion more often than max-sum does (68.0% against 68.0% of these requests on the main benchmark, 62.9% against 58.5% on the second bank); the difference lies in which candidates fill the other slots. Max-sum recovers part of the gap on the main benchmark, where a rare survivor paired with a common one wins more often under max-sum (5.7 points [3.8, 7.7]).

Per backbone, the $j' \geq 2$ stratum carries a resolved advantage on Klein on both banks (+7.9 and +6.3 points) and on second-bank SD3.5 (+6.7). On main-benchmark SD3.5 the advantage sits in $j' = 1$ (+2.8 [0.2, 5.3]), and on FLUX.1-dev and on Infinity-2B, which keeps three survivors, no single stratum is resolved. In sixteen-candidate pools with two survivors, $j' \geq 2$ requests carry 1.47 [0.09, 2.82] of 2.34 points and single-candidate requests 0.87 [0.34, 1.42]. With three or four survivors no $j' \geq k$ stratum has a positive resolved advantage, and two small ones, holding 4.7% and 2.6% of requests, are negative; the rare-mode advantage is a two-survivor effect.

D.6 Robustness of the mode comparison

Mode definition. Table 15 uses one mode definition: eight average-linkage DreamSim clusters per prompt, with a cluster counted as rare if it holds at most one eighth of the 64 native final images. Table 16 varies it along four axes, with the same matched mixtures and bootstrap: the number of clusters $K \in \{4, 6, 8, 12, 16\}$, the rare threshold $f \in \{1/16, 1/8, 1/4\}$, whether single-image clusters count as modes, and the linkage. It also refits the clusters in OpenCLIP ViT-g/14 space, which the selector does not use.

Over the 60 DreamSim definitions per bank, the rare-mode advantage of density over max-sum is positive in 119 of 120 bank–definition pairs, the exception being −0.03 points, and no interval lies below zero. With the main definition in OpenCLIP space it is +1.5 points [+0.7, +2.4], against +1.6 in DreamSim space. Single images are not the main source: excluding single-image clusters lowers the main-definition advantage from 1.6 to 1.0 points [0.2, 1.8] on the main benchmark and from 3.1 to 3.0 [1.8, 4.1] on the second bank. How often the interval excludes zero depends on the definition on the main benchmark, where it does so in 41 of 60 definitions, against 48 of 60 on the second bank; with average linkage, single-image clusters excluded and $K \leq 6$, it is not resolved on the main benchmark. The distinct-mode difference is the more fragile measure. It stays positive almost everywhere, but it is not resolved under complete linkage on either bank, nor on the main benchmark once single-image clusters are excluded.

Rare modes without clusters. A cluster-free measure gives every native final image a rarity, its leave-one-out kernel surprisal among the prompt's 64 native final images, $s(y) = -\log\left(\frac{1}{63}\sum_{e \neq y} \exp(-d(y,e)/h)\right)$, with DreamSim distance d and bandwidth h equal to the prompt's median nearest-neighbour distance. The bandwidth depends on the native final images only. Tail mass is the share of delivered images among the prompt's rarest 1/4, 1/8 or 1/16 of native final images. Table 17 repeats the comparison of Table 15 with these measures.

At matched ImageReward density delivers the rarest native images more often than max-sum. On the main benchmark its share of the rarest eighth is 15.5% against 14.6% at the development strength, a difference of 0.9 points [0.2, 1.6]; native sampling delivers 12.5% and reward-only pruning 11.6%. Mean rarity is higher on both banks, by 0.024 [0.009, 0.045] and 0.031 [0.009, 0.061] nats. Across the full grid of four bandwidths, four measures, four reward positions and two banks, all 128 pooled differences are positive, 90 have intervals above zero, and none lies below zero.

Table 16: Sensitivity of Table 15 to the mode definition: density minus max-sum at matched mean ImageReward, pooled over backbones per prompt, with 95% stratified prompt-bootstrap intervals; bold intervals exclude zero. Each prompt's 64 native final images form K clusters (average linkage on DreamSim distance unless marked complete linkage or OpenCLIP ViT-g/14 features); a cluster is rare if it holds at most $64f$ images, and excluded singletons are neither rare nor modes. Table 15 is $K = 8$, $f = 1/8$, singletons counted. Across all K and f, rare-share intervals exclude zero in 13 and 15 of 15 definitions (main, second bank) with singletons counted and 7 and 14 without (none at $K \leq 6$ on the main benchmark); complete linkage gives 12 and 10 (counted), 9 and 9 (excluded), with no distinct-mode interval above zero; OpenCLIP clusters give 9 and 4 on the main benchmark. No interval in the grid lies below zero.

	Singleton clusters counted				Singleton clusters excluded			
K	ΔDistinct	ΔRare share (pp), $f =$			ΔDistinct	ΔRare share (pp), $f =$		
		$1/16$	$1/8$	$1/4$		$1/16$	$1/8$	$1/4$
Main benchmark (4 backbones)								
4	**+0.051** [+0.010, +0.092]	**+0.6** [+0.0, +1.1]	**+0.7** [+0.1, +1.4]	**+0.8** [+0.1, +1.6]	+0.019 [−0.017, +0.053]	+0.2 [−0.3, +0.6]	+0.3 [−0.3, +0.9]	+0.4 [−0.3, +1.2]
6	+0.036 [−0.017, +0.089]	+0.5 [−0.2, +1.2]	+0.8 [−0.0, +1.5]	**+1.0** [+0.1, +1.9]	−0.004 [−0.049, +0.039]	+0.0 [−0.5, +0.5]	+0.3 [−0.4, +0.9]	+0.5 [−0.3, +1.3]
8	**+0.085** [+0.023, +0.147]	**+1.0** [+0.2, +1.8]	**+1.6** [+0.7, +2.5]	**+1.4** [+0.4, +2.4]	+0.041 [−0.010, +0.092]	+0.4 [−0.2, +1.1]	**+1.0** [+0.2, +1.8]	+0.9 [−0.0, +1.8]
12	**+0.093** [+0.017, +0.168]	**+1.6** [+0.7, +2.5]	**+2.2** [+1.1, +3.3]	**+1.9** [+0.8, +2.9]	+0.058 [−0.003, +0.119]	**+1.2** [+0.5, +1.9]	**+1.7** [+0.8, +2.7]	**+1.4** [+0.5, +2.4]
16	**+0.102** [+0.018, +0.188]	**+2.3** [+1.3, +3.4]	**+2.8** [+1.7, +3.9]	**+2.5** [+1.5, +3.5]	+0.050 [−0.016, +0.116]	**+1.7** [+0.9, +2.5]	**+2.2** [+1.3, +3.1]	**+1.9** [+0.9, +2.8]
8 complete	+0.029 [−0.035, +0.091]	+0.5 [−0.1, +1.2]	**+1.5** [+0.5, +2.4]	**+1.2** [+0.2, +2.2]	+0.030 [−0.034, +0.093]	+0.6 [−0.1, +1.2]	**+1.5** [+0.5, +2.4]	**+1.2** [+0.2, +2.2]
8 OpenCLIP	**+0.081** [+0.021, +0.141]	**+0.8** [+0.0, +1.5]	**+1.5** [+0.7, +2.4]	**+1.4** [+0.4, +2.3]	+0.043 [−0.009, +0.096]	+0.3 [−0.3, +1.0]	**+1.1** [+0.3, +1.8]	**+0.9** [+0.0, +1.7]
Second bank (2 backbones)								
4	**+0.068** [+0.002, +0.136]	**+1.0** [+0.2, +1.9]	**+1.1** [+0.1, +2.1]	**+1.2** [+0.1, +2.3]	**+0.072** [+0.018, +0.127]	**+1.1** [+0.4, +1.8]	**+1.2** [+0.3, +2.0]	**+1.3** [+0.2, +2.3]
6	**+0.119** [+0.033, +0.207]	**+1.3** [+0.2, +2.3]	**+2.1** [+0.8, +3.4]	**+2.6** [+1.1, +4.1]	**+0.105** [+0.039, +0.175]	**+1.1** [+0.3, +2.0]	**+1.9** [+0.8, +3.1]	**+2.4** [+1.0, +3.8]
8	**+0.142** [+0.049, +0.239]	**+1.7** [+0.5, +2.9]	**+3.1** [+1.7, +4.5]	**+3.0** [+1.4, +4.6]	**+0.130** [+0.060, +0.203]	**+1.5** [+0.6, +2.5]	**+3.0** [+1.8, +4.1]	**+2.8** [+1.4, +4.2]
12	**+0.201** [+0.089, +0.314]	**+3.2** [+1.8, +4.6]	**+3.8** [+2.1, +5.5]	**+3.6** [+2.0, +5.2]	**+0.110** [+0.021, +0.201]	**+2.0** [+1.0, +3.1]	**+2.7** [+1.2, +4.1]	**+2.5** [+1.1, +3.8]
16	**+0.211** [+0.092, +0.329]	**+4.0** [+2.3, +5.6]	**+4.1** [+2.3, +5.9]	**+3.0** [+1.5, +4.6]	+0.081 [−0.016, +0.179]	**+2.4** [+1.0, +3.8]	**+2.5** [+0.9, +4.1]	+1.4 [−0.0, +2.8]
8 complete	+0.077 [−0.033, +0.186]	+0.7 [−0.3, +1.8]	**+3.2** [+1.6, +4.8]	**+2.1** [+0.5, +3.7]	+0.072 [−0.030, +0.173]	+0.7 [−0.3, +1.6]	**+3.2** [+1.6, +4.7]	**+2.0** [+0.5, +3.5]

Table 17: Rare-mode delivery without clusters. Each native final image's rarity is its leave-one-out kernel surprisal $s(y)$ among the prompt's 64 native final images (DreamSim distance); m_p is the prompt's median nearest-neighbour distance and $\bar{m}$ its median over main-benchmark prompts, one value per backbone. Tail mass is the share of delivered images among the prompt's rarest $1/4$, $1/8$ or $1/16$ of native final images (native sampling: 25, 12.5 and 6.25%); mean rarity is in nats. Differences are at matched mean ImageReward, at density's development strength (Dev., the setting of Table 15) and at 25, 50 and 75% of the reward range density and max-sum share; reward sampling is reward-only pruning with Plackett–Luce survivor choice, matched the same way. Pooled over backbones per prompt, 95% stratified prompt-bootstrap intervals; bold intervals exclude zero.

		Density − max-sum					Density − reward sampling
Bank	Matched at	Rare-mode share (pp)	Mean rarity	Tail mass (pp), rarest			Tail mass, rarest $1/8$ (pp)
				$1/4$	$1/8$	$1/16$	
(a) Bandwidth $h = m_p$, at matched mean ImageReward							
Main	Dev.	**+1.6** [+0.7, +2.5]	**+0.024** [+0.009, +0.045]	**+1.2** [+0.2, +2.1]	**+0.9** [+0.2, +1.6]	**+0.8** [+0.3, +1.3]	**+4.0** [+3.4, +4.7]
	25%	**+2.7** [+1.7, +3.6]	**+0.032** [+0.016, +0.055]	**+1.8** [+0.8, +2.8]	**+1.4** [+0.5, +2.1]	**+1.2** [+0.6, +1.7]	**+5.9** [+5.2, +6.6]
	50%	**+1.8** [+0.9, +2.7]	**+0.025** [+0.009, +0.048]	**+1.3** [+0.3, +2.3]	**+0.9** [+0.2, +1.6]	**+0.8** [+0.3, +1.3]	**+4.7** [+4.0, +5.4]
	75%	**+0.9** [+0.1, +1.7]	**+0.017** [+0.004, +0.036]	**+1.0** [+0.2, +1.9]	+0.6 [−0.1, +1.2]	**+0.5** [+0.1, +1.0]	**+3.4** [+2.8, +4.0]
Second	Dev.	**+3.1** [+1.7, +4.5]	**+0.031** [+0.009, +0.061]	**+2.8** [+1.1, +4.5]	+0.8 [−0.4, +2.0]	+0.6 [−0.3, +1.5]	**+6.0** [+4.9, +7.1]
	25%	**+3.1** [+1.7, +4.5]	**+0.033** [+0.011, +0.062]	**+2.4** [+0.8, +4.1]	+0.4 [−0.8, +1.6]	+0.6 [−0.3, +1.4]	**+6.5** [+5.4, +7.6]
	50%	**+2.6** [+1.2, +3.9]	**+0.028** [+0.006, +0.060]	**+2.1** [+0.5, +3.7]	+0.5 [−0.6, +1.6]	+0.4 [−0.4, +1.2]	**+5.3** [+4.4, +6.3]
	75%	**+1.4** [+0.3, +2.5]	**+0.023** [+0.003, +0.052]	**+1.7** [+0.4, +3.0]	+0.3 [−0.7, +1.2]	+0.1 [−0.7, +0.9]	**+3.6** [+2.7, +4.4]
(b) Other bandwidths, at the development strength							
Main	$h = \frac{1}{2}m_p$	—	**+0.049** [+0.021, +0.090]	**+1.1** [+0.2, +2.0]	**+0.8** [+0.1, +1.5]	**+0.9** [+0.4, +1.4]	—
	$h = 2m_p$	—	**+0.011** [+0.004, +0.023]	**+1.0** [+0.1, +2.0]	**+0.9** [+0.2, +1.6]	**+0.7** [+0.2, +1.3]	—
	$h = \bar{m}$	—	**+0.008** [+0.001, +0.015]	**+1.1** [+0.2, +2.1]	**+0.8** [+0.1, +1.5]	**+0.8** [+0.3, +1.3]	—
Second	$h = \frac{1}{2}m_p$	—	**+0.061** [+0.020, +0.118]	**+3.0** [+1.3, +4.6]	+0.8 [−0.3, +2.0]	+0.5 [−0.3, +1.4]	—
	$h = 2m_p$	—	**+0.016** [+0.004, +0.033]	**+2.6** [+1.0, +4.3]	+0.8 [−0.4, +2.0]	+0.4 [−0.5, +1.3]	—
	$h = \bar{m}$	—	**+0.012** [+0.001, +0.023]	**+2.7** [+0.9, +4.4]	+0.9 [−0.3, +2.0]	+0.6 [−0.3, +1.4]	—

E ADDITIONAL BASELINES

This appendix gives the setup and full results for the baselines of Section 6.2, Feynman–Kac resampling with the best-of-N reference, greedy MMR and k-DPP selection and CADS, and for two further alternatives: independently tuned schedules and final-rule temperatures, and density applied to the final images instead of the preview. It supports two claims. At a budget of four trajectories, no selection or resampling alternative delivers more output diversity than champion-free selection at comparable reward. CADS changes the candidates rather than which of them finish, and it combines with density.

E.1 FEYNMAN–KAC RESAMPLING

We run Feynman–Kac steering (Singhal et al., 2025) as a single-output policy on the main benchmark for all four backbones. A request carries N particles started from the first N cached native candidates, so resampling, best-of-N and pruning start from the same pool. At each resampling point every particle is decoded to its preview, scored with the same ImageReward call, and resampled multinomially with weights proportional to $\exp(r/(\sigma_r\, T))$, where σ_r is the reward scale of Table 5. All but the first copy of a duplicated particle are partially renoised with strength s, so lineages can separate again; on Infinity-2B, whose sampler has no noise to renoise, they redraw their remaining scales with fresh randomness instead. The request returns the particle with the highest final ImageReward. By default, resampling happens at the checkpoints the pruning schedule uses. $N = 4$ matches the budget of the pruning schedule (4 against 3.93 trajectories on Infinity-2B), and $N = 8$, run on Klein and SD3.5, doubles it. The temperature $T = 0.5$ and renoising strength $s = 0.3$ were chosen on SD3.5 over 13 of the benchmark prompts, as the pair with the highest delivered reward (every pair stayed below pruning's), and are used unchanged on all four backbones.

Table 18 reports the results. At equal budget, resampling does not change diversity measurably, with all four DreamSim intervals including zero, while it loses 0.024 to 0.054 ImageReward. On Klein and SD3.5, doubling the budget to $N = 8$ raises reward and lowers diversity. The lineage statistic explains why: 51% to 64% of $N = 4$ requests on the three diffusion backbones end with all particles descended from one initial noise, and 23% on Infinity-2B. Resampling every four steps is the one variant that gains diversity ($+0.0088$), but it spends 0.091 ImageReward and stays below the best-of-N reference curve. Against density directly, resampling with four particles delivers 0.0022 to 0.0171 less DreamSim on all four backbones, every interval below zero, and 0.010 to 0.042 less ImageReward on SD3.5, FLUX.1-dev and Infinity-2B; on Klein the reward difference, -0.007, is unresolved. The comparison covers resampling weights that depend on reward alone; weights with an explicit diversity term are outside its scope.

Best-of-N reference. Best-of-N spends the same kind of budget without pruning. The last column of Table 18 reads each method against the best-of-N curve through native sampling, best-of-4 and best-of-8 at the method's own ImageReward (Appendix C.3). Reward-only pruning lies on or below this curve on every backbone (-0.0073 to -0.0007). Density lies above it on all four, by 0.0027 to 0.0097 with every interval above zero, and max-sum on three, with SD3.5 on the curve ($+0.0007$, unresolved). Appendix D.2 repeats the comparison with PickScore as the quality axis.

E.2 GREEDY MMR AND k-DPP

Greedy maximal marginal relevance (MMR) keeps the preview champion and then adds the candidate that maximises scaled reward plus η times its minimum distance to the kept set, with distances divided by $\bar{d}$ (Carbonell & Goldstein, 1998). We sweep $\eta \in \{0.05, \ldots, 6.4\}$ in factors of two, plus its diversity limit, which keeps the champion and the farthest candidate. The k-DPP samples survivor sets with weight $\exp(b \sum_{i \in S} v_i) \det(F_S F_S^\top)$ for $b \in \{0.5, 1, \ldots, 16\}$ (Kulesza & Taskar, 2012). Both run on the main benchmark with the same previews, schedule and argmax delivery, and are evaluated exactly.

Table 1 places both in the main comparison, and Table 19 gives the paired differences against density. Greedy MMR never excludes the champion, at any strength, so on Klein, SD3.5 and Infinity its whole curve stays above density's reward (Appendix A.1); we compare density with MMR's diver-

Table 18: Resampling instead of pruning, on the main benchmark of 200 PartiPrompts and the same eight independent requests per prompt. Feynman–Kac steering is run as a single-output policy: N particles initialised from the same cached initial noises as the pruning candidates, multinomial resampling on the same previews reward-only pruning cuts at, duplicated lineages partially renoised (on Infinity-2B, whose sampler has no noise to renoise, duplicates redraw their remaining scales), final ImageReward argmax. Differences are paired over prompts against reward-only pruning at the same budget, with 95% stratified prompt-bootstrap intervals. "Frontier excess" is the method's delivered DreamSim minus the best-of-N reference curve (native $\rightarrow$ best-of-4 $\rightarrow$ best-of-8) interpolated at the method's own delivered ImageReward: positive means the method keeps diversity that spending the same reward by enlarging N does not buy. "One ancestor" is the share of requests whose final particles all descend from a single initial noise. The $N = 8$ and variant rows were run on Klein and SD3.5; Infinity-2B's pruning costs 3.93 trajectories.

Method	Budget	ΔImageReward	ΔDreamSim	Δdistinct/8	Frontier excess
FLUX.2 Klein 4B					
Reward-only pruning	4×	—	—	—	−0.0013 [−0.0039, +0.0010]
FK steering, $N = 4$ (62% one ancestor)	4×	−0.0361 [−0.0459, −0.0267]	+0.0022 [−0.0014, +0.0059]	+0.01 [−0.13, +0.16]	−0.0011 [−0.0039, +0.0015]
FK steering, $N = 8$ (33% one ancestor)	8×	+0.0235 [+0.0129, +0.0349]	−0.0021 [−0.0048, +0.0007]	−0.04 [−0.18, +0.11]	−0.0028 [−0.0052, −0.0006]
Density	4×	−0.0290 [−0.0358, −0.0225]	+0.0125 [+0.0102, +0.0150]	+0.49 [+0.35, +0.62]	+0.0097 [+0.0067, +0.0128]
Max-sum	4×	−0.0253 [−0.0310, −0.0197]	+0.0113 [+0.0089, +0.0140]	+0.29 [+0.17, +0.41]	+0.0087 [+0.0059, +0.0116]
SD3.5 Medium					
Reward-only pruning	4×	—	—	—	−0.0073 [−0.0102, −0.0044]
FK steering, $N = 4$ (64% one ancestor)	4×	−0.0494 [−0.0626, −0.0370]	+0.0013 [−0.0033, +0.0060]	−0.10 [−0.24, +0.05]	−0.0099 [−0.0140, −0.0059]
FK steering, $N = 4$, no rejuvenation (64% one ancestor)	4×	−0.1153 [−0.1348, −0.0973]	+0.0030 [−0.0017, +0.0079]	−0.08 [−0.22, +0.07]	−0.0128 [−0.0169, −0.0086]
FK steering, $N = 4$, resample every 4 steps (97% one ancestor)	4×	−0.0909 [−0.1109, −0.0732]	+0.0088 [+0.0030, +0.0153]	−0.03 [−0.18, +0.13]	−0.0053 [−0.0098, −0.0006]
FK steering, $N = 8$ (33% one ancestor)	8×	+0.0323 [+0.0199, +0.0453]	−0.0046 [−0.0086, −0.0005]	−0.20 [−0.33, −0.07]	−0.0088 [−0.0130, −0.0043]
Density	4×	−0.0274 [−0.0343, −0.0207]	+0.0184 [+0.0139, +0.0234]	+0.36 [+0.24, +0.49]	+0.0087 [+0.0046, +0.0131]
Max-sum	4×	−0.0119 [−0.0162, −0.0078]	+0.0092 [+0.0065, +0.0120]	+0.17 [+0.06, +0.28]	+0.0007 [−0.0026, +0.0041]
FLUX.1-dev					
Reward-only pruning	4×	—	—	—	−0.0030 [−0.0053, −0.0007]
FK steering, $N = 4$ (51% one ancestor)	4×	−0.0543 [−0.0670, −0.0426]	+0.0037 [−0.0005, +0.0079]	+0.04 [−0.10, +0.17]	−0.0028 [−0.0068, +0.0009]
Density	4×	−0.0121 [−0.0155, −0.0084]	+0.0101 [+0.0072, +0.0131]	+0.30 [+0.20, +0.41]	+0.0063 [+0.0032, +0.0095]
Max-sum	4×	−0.0126 [−0.0157, −0.0094]	+0.0114 [+0.0087, +0.0142]	+0.23 [+0.12, +0.34]	+0.0076 [+0.0046, +0.0105]
Infinity-2B					
Reward-only pruning	3.93×	—	—	—	−0.0007 [−0.0021, +0.0006]
FK steering, $N = 4$ (23% one ancestor)	4×	−0.0241 [−0.0320, −0.0172]	+0.0014 [−0.0003, +0.0032]	+0.03 [−0.09, +0.15]	+0.0003 [−0.0008, +0.0015]
Density	3.93×	−0.0138 [−0.0167, −0.0111]	+0.0036 [+0.0025, +0.0048]	+0.21 [+0.11, +0.30]	+0.0027 [+0.0011, +0.0043]
Max-sum	3.93×	−0.0276 [−0.0345, −0.0219]	+0.0049 [+0.0032, +0.0066]	+0.26 [+0.15, +0.37]	+0.0038 [+0.0019, +0.0055]

Table 19: Density delivers more output diversity and more distinct modes than both classical set-diversity selectors on every backbone. Density against greedy MMR and the k-DPP on the main benchmark, with 95% prompt-bootstrap intervals. Greedy MMR: density minus MMR at density's ImageReward on FLUX.1-dev, and minus MMR's most diverse setting on the other backbones, where even that setting keeps more reward than density. k-DPP: density minus the k-DPP curve at density's ImageReward, using exact mixtures. Modes: distinct clusters among eight delivered images.

	Density − greedy MMR		Density − k-DPP	
Backbone	ΔDreamSim	ΔModes	ΔDreamSim	ΔModes
FLUX.2 Klein 4B	+0.0076 [+0.0050, +0.0103]	+0.40 [+0.27, +0.53]	+0.0072 [+0.0053, +0.0093]	+0.36 [+0.25, +0.47]
SD3.5 Medium	+0.0117 [+0.0072, +0.0170]	+0.21 [+0.08, +0.34]	+0.0115 [+0.0079, +0.0155]	+0.28 [+0.19, +0.38]
FLUX.1-dev	+0.0045 [+0.0014, +0.0082]	+0.25 [+0.14, +0.35]	+0.0057 [+0.0035, +0.0080]	+0.21 [+0.12, +0.29]
Infinity-2B	+0.0033 [+0.0021, +0.0045]	+0.23 [+0.13, +0.33]	+0.0020 [+0.0011, +0.0030]	+0.15 [+0.08, +0.23]

sity limit there, which still keeps 0.005 (unresolved) to 0.010 more ImageReward than density. The k-DPP can drop the champion and excludes it on 3% to 54% of requests, depending on b.

E.3 Independently tuned schedules and final-rule temperatures

The comparisons above fix the $6 \rightarrow 2$ schedule and argmax delivery for every method. Two studies let each method tune these as well. In the first, each family chose one of nine equal-cost schedules and a final softmax inverse temperature θ on development data that included the main benchmark, under a common quality budget, and was evaluated on the second bank. On the second bank density delivers the highest DreamSim of the three tuned families on both backbones, 0.1728 on Klein and 0.2435 on SD3.5, against 0.1702 and 0.2346 for tuned max-sum and 0.1655 and 0.2274 for tuned reward-only pruning, at the lowest ImageReward of the three (1.4148 and 1.3865). Compared with the tuned reward-only family at density's own quality cost, its DreamSim is higher by

Table 20: Strength and final-rule temperature tuned jointly. Klein, SD3.5 and FLUX.1-dev. Each family receives the same development banks and the same quality budget and chooses its selection strength and final-rule inverse temperature θ at the fixed $6 \to 2$ schedule; the chosen configuration is then evaluated on the main benchmark against reward-only pruning with argmax delivery (paired 95% prompt bootstrap; quality budget is the worst judge's share of its tolerance at the point estimate and at the bound). Reward-only pruning has no configuration whose diversity bound clears zero on every development bank, so its row is the configuration the quality-budget constraints alone select. The budget constrains the choice on the development banks; on the main benchmark a chosen configuration can exceed it, as density does on SD3.5. Right: for every softened configuration (finite θ) of max-sum and density, its excess is its main-benchmark gain minus the same selector's argmax curve interpolated at the same quality budget; the columns count configurations whose excess interval lies above or below zero.

Backbone	Family	Strength	θ	ΔDreamSim [95%]	Quality budget	Softened	> 0	< 0
Klein	Reward-only	–	1	$+0.0019$ [$+0.0007$, $+0.0031$]	54% / 57%			
	Max-sum	$\lambda = 1.6$	argmax	$+0.0113$ [$+0.0089$, $+0.0140$]	51% / 71%	40	0	19
	Density	$\beta = 1/32$	argmax	$+0.0125$ [$+0.0102$, $+0.0150$]	61% / 88%	27	0	6
SD3.5	Reward-only	–	2	$+0.0024$ [$+0.0010$, $+0.0039$]	44% / 58%			
	Max-sum	$\lambda = 0.8$	4	$+0.0119$ [$+0.0093$, $+0.0147$]	72% / 94%	45	0	26
	Density	$\beta = 1/16$	8	$+0.0191$ [$+0.0148$, $+0.0240$]	107% / 136%	30	0	24
FLUX.1-dev	Reward-only	–	2	$+0.0020$ [$+0.0006$, $+0.0035$]	45% / 60%			
	Max-sum	$\lambda = 0.8$	4	$+0.0128$ [$+0.0104$, $+0.0154$]	85% / 112%	45	0	37
	Density	$\beta = 1/4$	8	$+0.0120$ [$+0.0092$, $+0.0150$]	82% / 110%	31	0	12

$[0.0036, 0.0086]$ on Klein and $[0.0078, 0.0155]$ on SD3.5. At the lower quality cost of the tuned reward-only policy, the corresponding intervals include zero.

In the second study, each family chose its strength and θ jointly at the $6 \to 2$ schedule and was evaluated on the main benchmark (Table 20). Its quality budget is the worst judge's share of its tolerance (Appendix B.4). A softer final rule never improves on a champion-free selector's own argmax curve. Across 218 softened configurations, none lies above the argmax curve of the same selector at the same quality budget, and 124 lie below it. Softened settings enter the tuned choices only where the strength grid is coarser than the quality budget. Softening helps only reward-only pruning, whose best tuned setting adds 0.0019 to 0.0024 DreamSim, far less than the 0.011 to 0.019 that density and max-sum add. Softening substitutes for a missing diversity mechanism rather than adding to one.

E.4 Density on the final images

The diversity term can also be spent after generation: run best-of-4 and deliver the finished image with the largest $\beta R_i/\sigma_R - \log n_i$, with the crowding n_i computed over the four final DINOv2 features. This rule has no preview and no champion constraint, and it costs the same four trajectories. Its reward scale σ_R and kernel width $\tau_F = \bar{d}_F/4$ come from the development banks by the recipe of Appendix B.3, and its strength sweeps β from 8 to 2^{-12}. Figure 8 and Table 21 compare it with density at the midpoint preview.

Selecting at the preview reaches higher reward and delivers rare modes more often. The final-image rule cannot exceed the reward of best-of-4, its own reward-only end, and density's development strength lies above that reward on FLUX.1-dev and Infinity-2B while delivering 0.0050 and 0.0024 more DreamSim. Where the two curves share a reward range, at its middle on the main benchmark, density delivers images from rare modes 3.9 percentage points [2.6, 5.2] more often, 0.12 [0.04, 0.21] more distinct modes and 0.0021 [0.0001, 0.0040] more DreamSim, at equal PickScore. On the second bank the pooled differences keep their sign but are not resolved: DreamSim $+0.0020$ [-0.0007, $+0.0047$] and rare-mode share $+1.0$ [-0.5, $+2.5$] points. The DreamSim lead grows to 0.0047 at half the final kernel width and vanishes at twice it, where the rare-mode lead remains ($+3.0$ points [1.7, 4.4]). With six finished images, at six trajectories, the final-image rule overtakes the progressive one: the preview is what lets progressive selection choose among six candidates for the price of four.

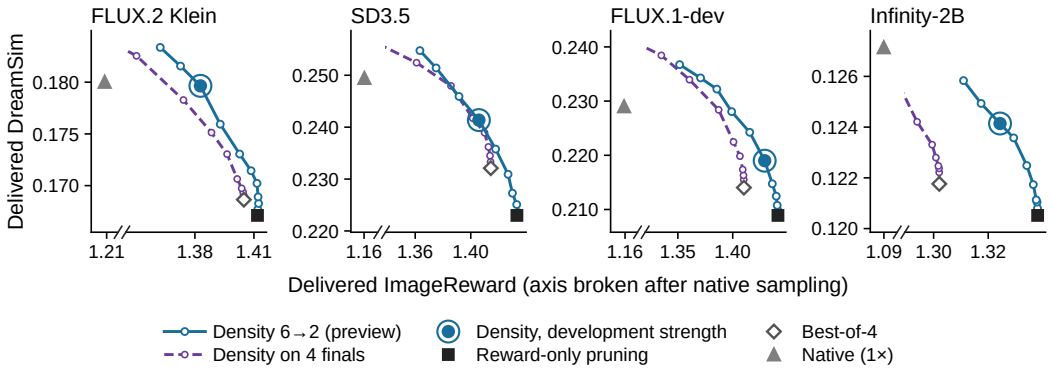

Figure 8: Density at the midpoint preview ($6 \to 2$, solid) and density on the four final images of best-of-4 (dashed), both at four full trajectories, on the main benchmark. The final-image rule ends at best-of-4 and cannot reach a higher reward. The preview rule lies above it on Klein and FLUX.1-dev, matches it on SD3.5, and reaches rewards it cannot on FLUX.1-dev and Infinity-2B.

Table 21: Density on the midpoint preview ($6 \to 2$) minus density on the four final images of best-of-4 ($\arg\max_i \beta R_i/\sigma_R - \log n_i$ over the four finals, no preview), both at four full trajectories, at matched mean ImageReward at the midpoint of the two curves' shared ImageReward range. Paired differences with 95% stratified prompt-bootstrap intervals (10,000 replicates). The final-image rule uses the development-calibrated final-image scale σ_R and kernel width $\tau_F = \bar{d}_F/4$ and sweeps β from 8 to 2^{-12} and 0; both curves are matched with exact per-request policy mixtures. Modes counts distinct content clusters among the eight delivered images; Rare is the share of delivered images in clusters holding at most one eighth of the prompt's 64 native final images. On Infinity-2B the two curves share no reward range: density's weakest strength ($\beta = 1/128$) delivers 1.311 ImageReward, above best-of-4's 1.302. Pooled rows average the backbones where the rows above are matched.

Bank	Backbone / rule	ΔDreamSim ($\times 10^{-3}$)	ΔModes	ΔRare (pp)	ΔPickScore/26 ($\times 10^{-3}$)
Main	FLUX.2 Klein 4B	+3.1 [+0.3, +6.1]	+0.149 [−0.001, +0.295]	+3.2 [+1.2, +5.2]	+0.4 [−0.2, +1.1]
	SD3.5 Medium	+0.2 [−3.9, +4.0]	+0.043 [−0.106, +0.190]	+3.8 [+1.6, +6.0]	+0.2 [−0.6, +1.2]
	FLUX.1-dev	+3.0 [−0.4, +6.6]	+0.178 [+0.037, +0.315]	+4.7 [+2.7, +6.7]	+0.1 [−0.7, +0.9]
	Infinity-2B	no shared reward range			
	Pooled (Klein+SD3.5+FLUX.1-dev)	+2.1 [+0.1, +4.0]	+0.123 [+0.035, +0.207]	+3.9 [+2.6, +5.2]	+0.2 [−0.2, +0.8]
Second	FLUX.2 Klein 4B	+1.1 [−1.7, +3.9]	−0.018 [−0.162, +0.123]	+0.8 [−1.2, +2.8]	+0.2 [−0.6, +1.0]
	SD3.5 Medium	+2.9 [−1.6, +7.2]	+0.060 [−0.086, +0.205]	+1.2 [−1.0, +3.4]	−0.1 [−0.9, +0.6]
	Pooled	+2.0 [−0.7, +4.7]	+0.021 [−0.081, +0.119]	+1.0 [−0.5, +2.5]	+0.0 [−0.6, +0.6]

E.5 Diversity-aware sampling: CADS

CADS (Sadat et al., 2024) anneals the prompt condition early in sampling, $\hat{y} = \sqrt{g(t)}\, y + s\sqrt{1 - g(t)}\, n$, with g rising linearly from 0 at $t \geq t_2$ to 1 at $t \leq t_1$. We use $t_1 = 0.8$, $t_2 = 1.0$, $s = 0.001$ and no rescaling ($\psi = 0$), the setting of an earlier SD3.5 study, fixed before this experiment and not tuned on these prompts. The annealing covers the first 14 of 32 steps, so the preview at the midpoint cut is scored under the clean prompt. Following Algorithm 1 of Sadat et al. (2024), both the prompt condition and the null condition of classifier-free guidance are annealed, with independent noise. CADS candidates use the prompts, seeds and initial latents of the SD3.5 main benchmark, and with CADS switched off the pipeline reproduces the main-benchmark images bit for bit. Density keeps its native calibration and strength grid.

Table 22 and Figure 9 give the results. At $1\times$, CADS raises delivered DreamSim by 0.045 [0.038, 0.052] and lowers ImageReward by 0.098 [0.066, 0.130], with PickScore and CLIP also lower. Under reward-only pruning, CADS candidates reach ImageReward 1.406 against 1.433 for native candidates. At that reward they deliver more diversity than density on native candidates, 0.264 against 0.241 DreamSim, with PickScore and CLIP tied. Selection adds to this: on CADS candidates, density raises delivered DreamSim over reward-only pruning by 0.018 [0.014, 0.022] at the middle of their common reward range, against 0.015 on native candidates, and CADS with density exceeds native candidates with density by 0.034 [0.026, 0.041]. CADS changes how candidates

Table 22: CADS candidates versus native candidates on SD3.5 (200 prompts, eight requests per prompt). Best-of-4 keeps the best of four full trajectories; the pruning rules cut six candidates to two at the midpoint and deliver the survivor with the higher final ImageReward, so every rule except $1\times$ sampling costs four trajectories. Top: absolute scores of the delivered images. DreamSim is the mean distance between images delivered to different requests, Modes the expected number of distinct content clusters among the eight delivered images, and Rare the share of requests that receive an image from a rare cluster (at most 16 of the prompt's 128 pooled final images). Clusters are fitted once per prompt on both candidate sets together, so these mode counts differ from Table 15; most rare clusters hold only CADS images (4.0 CADS-only versus 0.7 native-only clusters per prompt). Bottom: paired per-prompt differences with 95% prompt-bootstrap intervals. *Matched* rows compare policies at equal mean ImageReward; rows (ii) and (iv) are shown at the midpoint of the two curves' shared reward range and have the same sign at its 25% and 75% points. In (ii), reward-only pruning reaches lower rewards by drawing its survivors from a softmax over preview reward. At equal reward, CADS with reward-only pruning delivers more diversity than density on native candidates (i); density still adds diversity on CADS candidates (ii), and the two combine (iv). CADS lowers ImageReward by 0.098 at $1\times$ (iii).

	Native candidates				CADS candidates			
Delivery rule	IR↑	DreamSim↑	Modes↑	Rare (%)↑	IR↑	DreamSim↑	Modes↑	Rare (%)↑
$1\times$ sampling	1.1607	0.2495	1.88	6.9	1.0630	0.2948	2.68	19.5
Best-of-4	1.4144	0.2321	1.72	5.8	1.3925	0.2680	2.28	14.9
Reward-only pruning	1.4332	0.2230	1.65	5.3	1.4065	0.2644	2.31	15.4
Density, $\beta = 1/16$	1.4059	0.2414	1.81	7.9	1.3799	0.2839	2.69	20.4

	Paired difference	ΔIR	ΔDreamSim	ΔModes	ΔRare (pp)
(i)	Native + density	*matched*	-0.0233	-0.497	-7.5
	− CADS + reward-only		$[-0.0312, -0.0154]$	$[-0.653, -0.343]$	$[-9.9, -5.1]$
(ii)	CADS + density	*matched*	$+0.0182$	$+0.411$	$+6.1$
	− CADS + reward-only		$[+0.0143, +0.0221]$	$[+0.312, +0.514]$	$[+4.7, +7.6]$
	CADS + density ($\beta = 1/16$)	-0.027	$+0.0195$	$+0.385$	$+5.1$
	− CADS + reward-only	$[-0.034, -0.020]$	$[+0.0154, +0.0239]$	$[+0.270, +0.500]$	$[+3.4, +6.7]$
(iii)	CADS $1\times$ − native $1\times$	-0.098	$+0.0453$	$+0.795$	$+12.6$
		$[-0.130, -0.066]$	$[+0.0385, +0.0525]$	$[+0.640, +0.950]$	$[+10.1, +15.0]$
(iv)	CADS + density	*matched*	$+0.0335$	$+0.751$	$+11.0$
	− native + density		$[+0.0263, +0.0406]$	$[+0.598, +0.901]$	$[+8.6, +13.4]$

are generated and density changes which of them finish, so the two combine. The mode columns use clusters fitted jointly on both candidate sets. Most rare clusters there hold only CADS images, so mode differences between the candidate sets partly count content that native sampling does not produce.

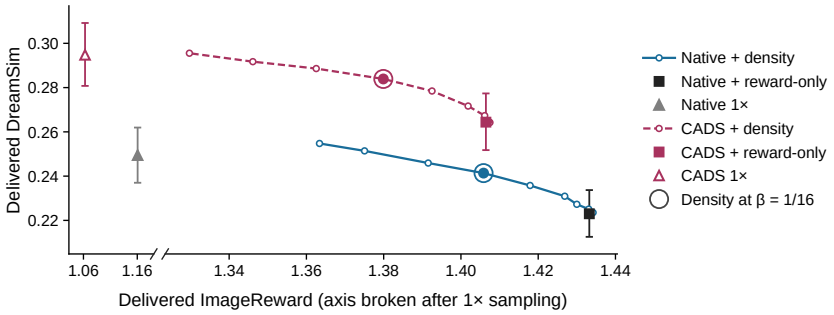

Figure 9: Output diversity against delivered ImageReward on SD3.5 for native candidates (solid) and CADS candidates (dashed). Each curve is density-aware selection over its strength grid, ending at reward-only pruning (square); rings mark the development strength $\beta = 1/16$, triangles $1\times$ sampling, and bars 95% prompt-bootstrap intervals. CADS moves the curve up and lowers the highest reward it reaches; on either candidate set, density raises output diversity above reward-only pruning.

F Champion interventions

This appendix gives the setup and complete results of the interventions in Section 6.3, which hold everything fixed except the requirement to keep the preview champion. It supports three claims. Requiring the champion lowers output diversity for both selectors, and the loss does not depend on preview errors. At matched reward the lost diversity is lost on locked requests, where Lemma 1 fixes the output. The same machinery shows that density's choice of replacement matters beyond the decision to exclude the champion.

F.1 Setup and full contrasts

Each selector has a free and a champion-required version (step 4c in Table 6). The required density rule keeps the preview champion and then the $k-1$ highest density scores among the remaining candidates. The required max-sum rule maximises J over the subsets that contain the champion; at $k = 2$ it coincides with champion-first greedy max-sum on every request and strength (Appendix A.1). Candidates, previews, native continuations, calibration, strengths and argmax delivery are identical across the two versions. The strengths are the development strengths of Table 5, and the curves sweep β from 4 down to $1/128$ in factors of two and $\lambda \in \{0.05, 0.1, \ldots, 6.4\}$ in factors of two.

Table 23 gives the second-bank comparison at the development strengths, and Table 24 lists all twelve contrasts. Figure 10 shows each selector's free and required curves on the second bank. Freeing the champion raises delivered DreamSim in every one, by 0.0029 to 0.0127 with every interval above zero, at a reward cost of 0.005 to 0.025 ImageReward. In the DINO-free space of Appendix D.4, all eight main-benchmark contrasts keep their direction with intervals above zero. On the second bank about half of all requests are locked (48.9% on Klein, 49.1% on SD3.5). Free density excludes the champion on 41.6% and 32.6% of requests, and free max-sum on 38.0% and 19.9%. Exclusion is not the same event as a new winner or a new mode: on Klein, free density changes reward-only pruning's winner on 46.9% of requests but its content cluster on only 22.4%.

F.2 Set and output diversity

A required version could convert less simply because its set-diversity gain is smaller. Figure 11 rules this out: it plots both gains behind Table 2, with 95% stratified prompt-bootstrap intervals, at density's development ImageReward; required versions that cannot give up that much reward are shown at their most diverse strength, which keeps more reward (the reward floor of Appendix A.1). The required versions' set-diversity gains are at least 92% of the free versions' and larger in six of the eight selector–backbone pairs. Their output-diversity gains have intervals above zero but stay below the free ones on every backbone. The second bank shows the same on Klein: free density's

Table 23: Requiring the preview champion on the second bank (200 prompts per backbone). Only this requirement changes; strength, candidates, previews and final-reward delivery are identical. Modes: distinct content clusters among the eight delivered images; Dropped: share of requests on which the champion does not survive. Freeing the champion raises DreamSim and the mode count in all four comparisons at a small reward cost (paired intervals: Table 24).

Backbone	Selector	Champion	ImageReward↑	DreamSim↑	Modes↑	Dropped
FLUX.2 Klein	Density	Required	1.4395	0.1635	2.770	0.0%
FLUX.2 Klein	Density	Free	1.4148	0.1728	3.110	41.6%
FLUX.2 Klein	Max-sum	Required	1.4399	0.1632	2.715	0.0%
FLUX.2 Klein	Max-sum	Free	1.4256	0.1702	2.935	38.0%
SD3.5	Density	Required	1.4096	0.2298	2.940	0.0%
SD3.5	Density	Free	1.3899	0.2422	3.285	32.6%
SD3.5	Max-sum	Required	1.4134	0.2291	2.870	0.0%
SD3.5	Max-sum	Free	1.4085	0.2342	2.990	19.9%

Table 24: Paired free-minus-required contrasts. Second-bank intervals are two-sided 98.75% prompt-bootstrap intervals (Appendix C.3); main-benchmark intervals are 95%. Reward differences are paired point estimates.

Bank	Backbone	Selector	Δ DreamSim	Interval	Δ ImageReward
Main	FLUX.2 Klein	density	0.00938	[0.00733, 0.01154]	-0.02185
Main	FLUX.2 Klein	max-sum	0.00593	[0.00387, 0.00803]	-0.01613
Main	SD3.5	density	0.01269	[0.00903, 0.01693]	-0.02082
Main	SD3.5	max-sum	0.00551	[0.00334, 0.00795]	-0.00549
Main	FLUX.1-dev	density	0.00607	[0.00360, 0.00861]	-0.00883
Main	FLUX.1-dev	max-sum	0.00699	[0.00480, 0.00918]	-0.00853
Main	Infinity	density	0.00285	[0.00193, 0.00380]	-0.01042
Main	Infinity	max-sum	0.00381	[0.00221, 0.00541]	-0.02289
Second	FLUX.2 Klein	density	0.00921	[0.00647, 0.01204]	-0.02475
Second	FLUX.2 Klein	max-sum	0.00701	[0.00459, 0.00948]	-0.01429
Second	SD3.5	density	0.01242	[0.00849, 0.01666]	-0.01964
Second	SD3.5	max-sum	0.00511	[0.00238, 0.00795]	-0.00489

two survivors lie barely further apart in DINOv2 preview distance than the required rule's (0.1884 against 0.1854), while delivered DreamSim differs clearly (0.1728 against 0.1635). Appendix A.4 shows that set diversity need not even order selectors by output diversity.

F.3 Locked share and preview quality

Lemma 1 makes the locked share $\Pr\{a = a^\star\}$ the fraction of requests on which a champion-preserving rule cannot change the output. It depends only on how well the preview ranks the final rewards. For N exchangeable candidates whose preview and final rewards are bivariate Gaussian with correlation ρ, $\Pr\{a = a^\star\}$ is a fixed function of ρ and N with no free parameter (Proposition 1). Figure 12 compares it with the measured locked share at every checkpoint we cached: the quarter and midpoint previews on Klein and SD3.5, steps 8, 16 and 24 of 32 on FLUX.1-dev, and scale 11 of 13 on Infinity-2B, on both banks where available.

The locked event depends only on ranks, so the natural correlation is a rank correlation. Panel (b) converts Kendall's rank correlation τ_{K} to $\rho = \sin(\pi\tau_{\mathrm{K}}/2)$, the Gaussian relation between the two; there, eleven of twelve points lie within 0.02 of the curve and the twelfth, the second-bank SD3.5 quarter preview, within 0.04. Panel (a) uses the request-demeaned Pearson correlation of Appendix B.5. The five midpoint points agree as well (residuals -0.033 to $+0.011$, every interval covering zero), the two late points lie below the curve, and the quarter points lie above it by up to 0.045, by more than 0.02 at four of five. Either way, the constraint binds more often as previews improve and would bind on every request with a perfect preview.

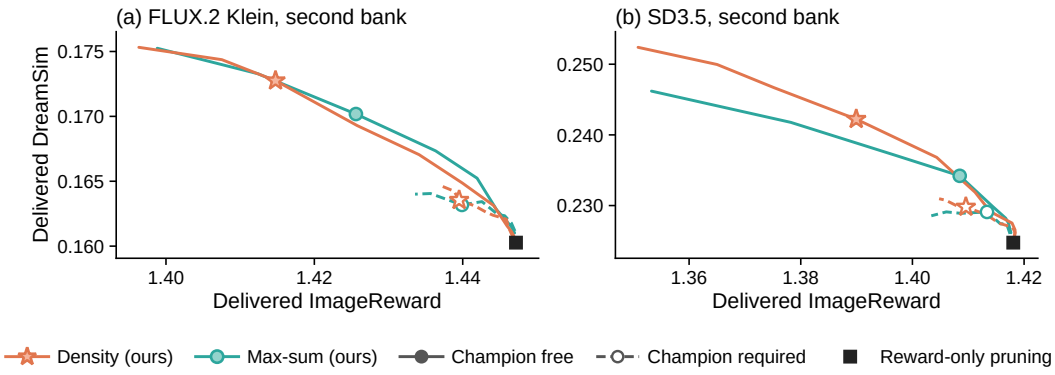

Figure 10: Free and champion-required curves on the second bank, each selector over its strengths. Solid curves may drop the champion; dashed curves must keep it and stay close to reward-only pruning. Markers: development strengths.

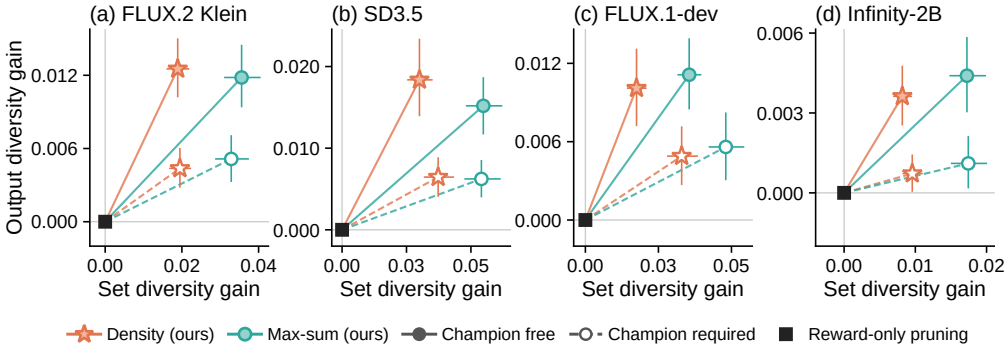

Figure 11: Keeping the champion blocks most of the conversion of set diversity into output diversity. Main benchmark, at density's development ImageReward: each selector's gain over reward-only pruning in set diversity (mean DreamSim among a request's survivors) and in output diversity (Eq. 1), with 95% intervals. Solid, filled: champion free; dashed, open: champion required. On Klein, SD3.5 and Infinity-2B even the most diverse required version keeps more reward than density.

Pool size. The same law predicts how the locked share falls as the pool grows. On the sixteen-candidate pools of Appendix G, the request-demeaned Pearson correlation of preview and final ImageReward is 0.661 on Klein and 0.588 on SD3.5. With these values and no fitted parameter, the law predicts the measured locked share within 0.02 at every pool size on Klein; on SD3.5 the prediction lies 0.03 to 0.04 below the measured share and falls at nearly the same rate (Table 25).

Table 25: Measured locked share on the sixteen-candidate pools of Appendix G against Proposition 1 at the measured preview–final Pearson correlation ($\rho = 0.661$ on Klein, 0.588 on SD3.5), with no fitted parameter.

	Klein		SD3.5	
N	measured	predicted	measured	predicted
6	0.472	0.473	0.457	0.427
8	0.422	0.428	0.421	0.378
12	0.359	0.372	0.356	0.321
16	0.321	0.339	0.321	0.289

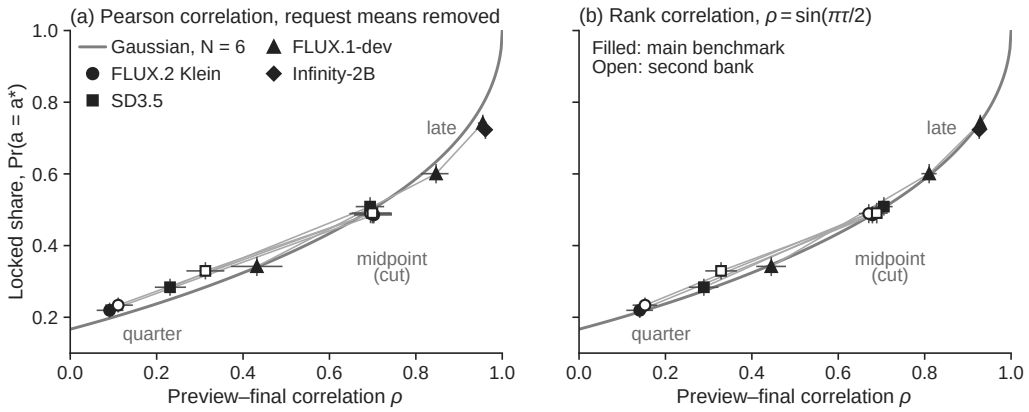

Figure 12: The share of locked requests follows preview quality as a Gaussian model with no free parameters predicts ($N = 6$). Each point is one backbone, bank and checkpoint, with 95% prompt-bootstrap intervals; grey lines join the checkpoints of one backbone. (a) Request-demeaned Pearson correlation between preview and final reward. (b) Kendall's rank correlation, converted as $\rho = \sin(\pi\tau_{\mathrm{K}}/2)$.

F.4 Matched reward and the per-request ledger

Because the fixed-strength contrasts also change reward, we compare free and required curves at equal mean ImageReward, using the exact mixtures of Appendix C.3. At 25%, 50% and 75% of the reward range that free and required curves share, ordered from low to high reward, the free rule has an interval above zero in $8/8$, $7/8$ and $2/8$ backbone–selector comparisons on the main benchmark. On the second bank the counts are $4/4$, $3/4$ and $1/4$. The gap narrows toward reward-only pruning: on the second bank it shrinks from 0.0014–0.0056 DreamSim at 25% to 0.0005–0.0015 at 75%. The ledger below shows where the gap comes from and why it narrows.

Where the cost of the constraint falls. A per-request ledger expresses Lemma 1 in DreamSim units. Fix a prompt with requests $q = 1, \ldots, 8$, and let w_q be the image that reward-only pruning delivers on request q. For a policy, let c_q be the probability that it delivers a different image on request q. Let g_q be the expected gain when it does, measured as the new image's extra distance to the other requests' reward-only images. Then, exactly for every prompt,

$$D(\text{policy}) - D(\text{reward-only}) = 2 \cdot \frac{1}{8} \sum_{q=1}^{8} c_q \, g_q + \bar{I}, \tag{9}$$

where $\bar{I}$ collects pairs of requests on which both images change; it is small and usually negative. The identity reproduces the evaluator to within 3×10^{-16}. Splitting requests into locked ($a = a^\star$) and unlocked ones splits the difference between two policies accordingly. For a free rule against its required version, the locked part equals $2Lc_Lg_L$, where L is the share of locked requests, because the required rule never changes a locked request. Table 26a applies the ledger to both selectors on both banks at the middle of the shared reward range; the findings below also use the 25% and 75% positions.

At matched reward the gap sits on locked requests. Across both banks, both selectors and the three matched positions (36 rows), the locked part is positive with an interval above zero in every row. In the 25 rows where the total gap is resolved, the locked part accounts for 66% to 166% of it, and 23 of the 25 intervals include 100%. The unlocked part has an interval that includes zero in 33 of 36 rows. On unlocked requests the free rule matches the required rule's diversity at higher reward, and it spends the saved reward on locked requests.

The constraint decides which requests can change, not how often. At matched reward the two rules change the winner about equally often overall; the average change rates differ by at most

Table 26: Where the champion constraint costs diversity, in DreamSim units ($\times 10^{-3}$) with 95% prompt-bootstrap intervals. A request is *locked* when its preview champion is also the final-reward winner, so any rule that keeps the champion must deliver that image. (a) At matched mean ImageReward, the free-minus-required gap splits exactly into a locked-request term, an unlocked term and an interaction between requests; the locked-request term carries the gap. (b) Against a control that drops the champion on exactly the same requests and is matched to density's reward, density gains through which candidate replaces the winner (gain per change), not through changing it more often (frequency).

(a) Free minus champion required, at the midpoint of the common reward range

Bank	Backbone	Rule	Gap	Locked-request term	Unlocked term	Interaction
Main	FLUX.2 Klein	Density	$+2.09$ [$+0.76$, $+3.45$]	$+1.97$ [$+1.10$, $+2.89$]	$+0.50$	-0.39
		Max-sum	$+2.66$ [$+1.13$, $+4.26$]	$+2.27$ [$+1.41$, $+3.21$]	$+0.94$	-0.55
	SD3.5	Density	$+1.75$ [-0.54, $+4.45$]	$+2.98$ [$+1.07$, $+5.60$]	-1.02	-0.22
		Max-sum	$+3.43$ [$+1.46$, $+5.63$]	$+3.47$ [$+1.89$, $+5.36$]	$+0.48$	-0.52
	FLUX.1-dev	Density	$+2.20$ [$+0.25$, $+4.14$]	$+3.65$ [$+1.91$, $+5.44$]	-0.80	-0.65
		Max-sum	$+3.41$ [$+1.28$, $+5.54$]	$+5.27$ [$+3.79$, $+6.94$]	-0.94	-0.92
	Infinity-2B	Density	$+0.80$ [$+0.19$, $+1.41$]	$+0.78$ [$+0.39$, $+1.21$]	$+0.01$	$+0.01$
		Max-sum	$+0.92$ [$+0.19$, $+1.72$]	$+0.76$ [$+0.27$, $+1.34$]	$+0.18$	-0.03
Second	FLUX.2 Klein	Density	$+1.08$ [-0.02, $+2.18$]	$+1.17$ [$+0.60$, $+1.80$]	-0.07	-0.02
		Max-sum	$+2.60$ [$+1.02$, $+4.22$]	$+2.11$ [$+1.24$, $+3.07$]	$+0.91$	-0.42
	SD3.5	Density	$+2.35$ [$+0.67$, $+4.06$]	$+3.34$ [$+2.07$, $+4.73$]	$+0.14$	-1.13
		Max-sum	$+3.50$ [$+1.71$, $+5.31$]	$+2.75$ [$+1.55$, $+4.05$]	$+1.14$	-0.38

(b) Density minus the same-event control, at density's development strength

Bank	Backbone	Total	Frequency part	Gain-per-change part	Interaction
Main	FLUX.2 Klein	$+5.62$ [$+3.71$, $+7.67$]	-0.63 [-1.06, -0.27]	$+7.31$ [$+5.13$, $+9.75$]	-1.05
	SD3.5	$+10.06$ [$+6.61$, $+14.00$]	-1.20 [-1.97, -0.53]	$+14.37$ [$+9.93$, $+19.79$]	-3.10
	FLUX.1-dev	$+5.34$ [$+3.32$, $+7.42$]	-0.53 [-0.93, -0.13]	$+6.76$ [$+4.40$, $+9.24$]	-0.89
	Infinity-2B	$+1.57$ [$+0.83$, $+2.35$]	-0.25 [-0.41, -0.11]	$+2.15$ [$+1.34$, $+3.00$]	-0.33
Second	FLUX.2 Klein	$+4.87$ [$+3.12$, $+6.68$]	-1.26 [-1.75, -0.82]	$+7.35$ [$+5.04$, $+9.77$]	-1.21
	SD3.5	$+9.66$ [$+7.32$, $+12.07$]	-2.28 [-3.08, -1.55]	$+14.34$ [$+11.18$, $+17.61$]	-2.40

0.036. The required rule compensates on unlocked requests, where it changes the winner 6.5 to 16.8 percentage points more often, but each of those changes adds less distance. The share of locked requests alone does not predict the gap. The locked part also depends on how often the free rule changes a locked request (8% to 19%) and on the gain per change, which varies sixfold across backbones. Infinity has the most locked requests (72%) but the smallest locked part.

The cost shrinks near reward-only pruning but does not vanish. Pooled over backbones, the free-minus-required gap is 0.18 to 0.28 of the free rule's own gain over reward-only pruning in the top quarter of the shared reward range, against 0.38 to 0.55 at its low end, with every pooled interval excluding zero; there the required rule still buys diversity cheaply on unlocked requests. At the other end, the required curves stop 0.005 to 0.019 ImageReward below reward-only pruning (Appendix A.1), while the development strengths of both selectors give up 0.010 to 0.032.

F.5 Replacement controls

Density decides both when to drop the champion and which candidate replaces it. The same-event control separates the two. Let $e(C) = \mathbf{1}\{a \notin S_{\text{density}}(C)\}$ record whether density drops the champion from pool C. The control keeps this decision on every request and draws a reward-weighted subset among those that make the same decision,

$$P_b(S \mid C) \;\propto\; \exp\Big(b \sum_{i \in S} v_i\Big)\, \mathbf{1}\big\{|S| = k,\; \mathbf{1}\{a \notin S\} = e(C)\big\}. \tag{10}$$

It therefore drops the champion on exactly the requests where density does and differs only in the candidate that replaces it. Four further controls replace density's crowding term: reward sampling without replacement (Plackett–Luce on scaled preview reward), density penalties shuffled among the pool's candidates (all 720 permutations of a six-candidate pool), k-DPP sampling on the preview

Table 27: Replacement controls on the second bank, matched to density's mean ImageReward without extrapolation. Density and Control: delivered DreamSim (higher is more diverse); the paired interval is for density minus control and conditions on the fitted mixture weights.

Backbone	Control	Reward	Density	Control	Paired interval
FLUX.2 Klein	Same-event	1.4148	0.1728	0.1679	[0.00312, 0.00668]
FLUX.2 Klein	Shuffled density	1.4148	0.1728	0.1634	[0.00729, 0.01143]
FLUX.2 Klein	Reward sampling	1.4148	0.1728	0.1640	[0.00642, 0.01113]
FLUX.2 Klein	k-DPP	1.4148	0.1728	0.1659	[0.00470, 0.00912]
FLUX.2 Klein	DPP MAP	1.4148	0.1728	0.1696	[0.00078, 0.00547]
FLUX.2 Klein	Exact max-sum	1.4148	0.1728	0.1727	[-0.00306, 0.00307]
SD3.5	Same-event	1.3899	0.2422	0.2326	[0.00732, 0.01207]
SD3.5	Shuffled density	1.3899	0.2422	0.2277	[0.01171, 0.01758]
SD3.5	Reward sampling	1.3899	0.2422	0.2286	[0.01049, 0.01697]
SD3.5	k-DPP	1.3899	0.2422	0.2306	[0.00860, 0.01486]
SD3.5	DPP MAP	1.3899	0.2422	0.2366	[0.00255, 0.00896]
SD3.5	Exact max-sum	1.3899	0.2422	0.2390	[-0.00051, 0.00719]

features (Appendix E.2), and the DPP's most probable subset. Each control sweeps its inverse temperature b, and all are evaluated exactly.

At matched mean ImageReward on the main benchmark, density delivers more DreamSim than the same-event control on all four backbones, with every interval above zero. The differences are 0.0056, 0.0101, 0.0053 and 0.0016 on Klein, SD3.5, FLUX.1-dev and Infinity-2B, and all four keep their direction in the DINO-free space of Appendix D.4. On the second bank, with every control matched to density's mean ImageReward (Table 27), density is above the same-event control, shuffled penalties, reward sampling, k-DPP sampling and the DPP's most probable subset on both backbones, and it ties with exact max-sum. Matching mean ImageReward does not equalise every judge: against the same-event control, density's PickScore/26 is lower by 0.0006 on Klein and 0.0005 on SD3.5.

Density against the replacement controls. The ledger of Appendix F.4 shows how density beats the same-event control (Table 26b). On locked requests the two change the winner with exactly the same probability, because both drop the champion on the same requests. On unlocked requests the control changes the winner more often, which raises its average change rate over all requests by 0.018 to 0.063, yet each of its changes adds less distance. The frequency component of the difference is therefore negative, 10% to 26% of the total, and the gain per change accounts for more than the whole advantage on all six backbone–bank combinations. The same holds against shuffled penalties, reward sampling and k-DPP sampling (18 of 18 comparisons). Against the DPP's most probable subset, density also changes winners more often, with a positive frequency component on four of the six. Against exact max-sum the total difference is unresolved on all six: max-sum changes winners more often, density gains more per change, and the two effects cancel.

F.6 Oracle rewards

The oracle replaces each preview reward by the candidate's final reward divided by the final-reward scale σ_R, and keeps the preview features and strengths. It uses future outcomes, so it is a diagnostic and not a policy. Every request is then locked, and every champion-preserving rule returns the final winner of the whole pool, exactly as reward-only pruning does (Lemma 1). The free rules still drop that winner on 22% to 44% of requests, gaining 0.012 to 0.029 DreamSim per changed request. Free minus required ranges from $+0.0069$ to $+0.0174$ DreamSim over the five combinations of a diffusion backbone and a bank and both selectors, with all ten intervals above zero, at a cost of 0.010 to 0.038 ImageReward. Preview errors are therefore not needed for the gains: freeing the champion trades reward for diversity even when previews are perfect. On Infinity-2B, where we ran the oracle for density alone, its DreamSim gain over reward-only pruning changes by -0.00005 $[-0.00056, 0.00044]$ when final rewards replace the previews.

G LARGER CANDIDATE POOLS

This appendix asks how the comparisons of Sections 6.3 and 6.4 change as the pool grows from 6 to 16 candidates. It supports four statements. Density and exact max-sum tie on mean DreamSim at every pool size, and density's rare-mode advantage with two survivors grows with the pool. Champion-first greedy selection falls further behind as the pool grows. Requiring the champion lowers the share of set diversity that reaches the output at every pool size and survivor count. Selection time stays negligible throughout.

Setup. Sixteen candidates per request come from the two independent seed runs of the earlier 200-prompt Klein and SD3.5 bank (Appendix C.1). Its first run is part of the development data on which both backbones' strengths were chosen, so this appendix compares complete strength curves at matched reward, not those strengths. Request q of the first seed run is paired with request q of the second, giving eight requests of sixteen candidates for each of 200 prompts per backbone. A pool of $N \in \{6, 8, 12, 16\}$ uses the first N candidates, and one midpoint cut keeps $k \in \{2, 3, 4\}$, at a budget of $N/2 + k/2$ native trajectories. Density, exact max-sum and champion-first greedy max-sum each sweep their strength grids. We compare them at the middle of the reward range they share, with the exact mixtures of Appendix C.3. Content clusters are refitted on the 128 final images of each prompt, with the same one-eighth threshold for rare modes. Results pool the two backbones; this is a replay of cached images, with no generation and no rescoring.

Findings. Table 28 reports the differences. First, density and exact max-sum tie on mean DreamSim in all twelve (N, k) settings, with differences of at most 0.001 and every interval including zero. Second, with $k = 2$ density's rare-mode share exceeds max-sum's by 2.3 percentage points $[0.8, 3.8]$ at $N = 16$ and by 1.3 $[0.0, 2.6]$ at $N = 12$. The increase from $N = 6$ to $N = 16$ is 1.9 points $[0.2, 3.6]$, carried mainly by SD3.5. At $N = 6$ the difference is 0.4 points $[-0.7, 1.5]$ and is not resolved on these banks. The shared reward range moves down as N grows, and its middle lies 0.03 to 0.08 ImageReward below reward-only pruning, so this growth combines a larger pool with a lower reward level. With three or four survivors the advantage shrinks and is not resolved. Third, champion-first greedy max-sum falls further behind as the pool grows. Density minus greedy DreamSim rises from 0.0007 $[-0.0009, 0.0022]$ at $N = 6$ to 0.0046 $[0.0023, 0.0069]$ at $N = 16$ with $k = 2$. Exact max-sum minus greedy (not tabulated) has a positive point estimate in all twelve settings, and its growth from $N = 6$ to $N = 16$ is resolved at every k. Greedy never excludes the champion, whereas density and exact max-sum exclude it on 32% to 35% of requests at $N = 6$ and on 50% to 58% at $N = 16$ ($k = 2$).

Conversion across pool sizes and survivor counts. Section 6.3 measures, at six candidates and two survivors, how much of a selector's gain in set diversity reaches the output. Figure 13 repeats the measurement for every N and k, on each backbone separately. Each family is matched to reward-only pruning minus 0.025 ImageReward, half the tolerance of Appendix B.4, and its conversion is its gain in output diversity divided by its gain in set diversity, both over reward-only pruning. The free version converts more than the required one in every cell for both selectors, at this loss and also at losses of 0.015 and 0.04. At a loss of 0.025, density converts 0.41 to 0.82 against 0.16 to 0.33, and max-sum 0.26 to 0.44 against 0.09 to 0.26. With sixteen candidates and four survivors, free density converts 0.72 and 0.50 on Klein and SD3.5, against 0.28 and 0.24 for the required version; free max-sum converts 0.36 and 0.31, against 0.17 and 0.18. The gap is largest with two survivors, where the champion takes half the slots, and narrows as more candidates survive. For density it is wider at sixteen candidates than at six for every k: with two survivors, free density rises from 0.63 to 0.82 on Klein and from 0.49 to 0.64 on SD3.5, while the required version stays between 0.17 and 0.24. The locked share falls from 0.47 (Klein) and 0.46 (SD3.5) at six candidates to 0.32 at sixteen, as Proposition 1 predicts (Table 25). The cell with six candidates and two survivors repeats the setting of Table 2 on a different bank and at a fixed loss rather than at density's development reward, and agrees with it in direction. Champion-first greedy max-sum converts within 0.03 of required exact max-sum in every cell at this loss.

How far the required rules reach. Required density cannot lose 0.025 ImageReward in 23 of the 24 cells and required max-sum in 10, against 2 and 1 cells for the free versions; there they are taken at their most diverse strength (dots in Figure 13), so keeping the champion also caps how much

Table 28: Larger candidate pools: differences at matched mean ImageReward (midpoint of the common reward range), pooled over FLUX.2 Klein and SD3.5 per prompt, with 95% prompt-bootstrap intervals. Each request pairs two independent eight-candidate seed runs of an earlier 200-prompt bank, so pools of $N \leq 16$ candidates keep k survivors at a budget of $(N + k)/2$ full trajectories. Distinct modes counts content clusters among the eight delivered images; the rare-mode share is the share of delivered images in clusters holding at most one eighth of the prompt's final images. Champion-first greedy builds the max-sum subset starting from the preview champion.

(a) Density minus max-sum

N	k	Budget	ΔDreamSim ($\times 10^{-3}$)	ΔDistinct modes	ΔRare-mode share (pp)
6	2	4×	-0.4 [-2.2, $+1.3$]	$+0.04$ [-0.04, $+0.12$]	$+0.4$ [-0.7, $+1.5$]
	3	4.5×	-0.8 [-2.5, $+0.7$]	$+0.00$ [-0.06, $+0.07$]	-0.4 [-1.3, $+0.5$]
	4	5×	-0.9 [-2.3, $+0.4$]	-0.02 [-0.06, $+0.03$]	-0.3 [-0.9, $+0.3$]
8	2	5×	-1.0 [-3.3, $+1.3$]	$+0.00$ [-0.08, $+0.09$]	-0.1 [-1.4, $+1.3$]
	3	5.5×	$+0.2$ [-1.7, $+2.0$]	$+0.00$ [-0.06, $+0.07$]	-0.4 [-1.3, $+0.5$]
	4	6×	-0.3 [-1.8, $+1.2$]	-0.01 [-0.06, $+0.05$]	-0.3 [-1.1, $+0.5$]
12	2	7×	$+0.9$ [-1.7, $+3.5$]	$+0.08$ [-0.01, $+0.17$]	$+1.3$ [$+0.0$, $+2.6$]
	3	7.5×	$+0.6$ [-1.6, $+2.7$]	$+0.02$ [-0.06, $+0.10$]	$+0.6$ [-0.5, $+1.7$]
	4	8×	-0.1 [-2.1, $+1.8$]	-0.00 [-0.08, $+0.07$]	-0.3 [-1.3, $+0.6$]
16	2	9×	$+0.5$ [-2.0, $+3.0$]	$+0.10$ [-0.00, $+0.20$]	$+2.3$ [$+0.8$, $+3.8$]
	3	9.5×	$+0.5$ [-2.0, $+3.0$]	$+0.03$ [-0.06, $+0.12$]	$+1.1$ [-0.2, $+2.4$]
	4	10×	$+0.4$ [-1.7, $+2.6$]	$+0.05$ [-0.03, $+0.12$]	$+0.8$ [-0.2, $+2.0$]

(b) Density minus champion-first greedy max-sum

N	k	Budget	ΔDreamSim ($\times 10^{-3}$)	ΔDistinct modes	ΔRare-mode share (pp)
6	2	4×	$+0.7$ [-0.9, $+2.2$]	$+0.09$ [$+0.03$, $+0.15$]	$+0.8$ [-0.1, $+1.6$]
	3	4.5×	$+0.6$ [-0.9, $+2.1$]	$+0.04$ [-0.01, $+0.09$]	$+0.4$ [-0.3, $+1.2$]
	4	5×	-0.1 [-1.3, $+1.1$]	-0.01 [-0.06, $+0.03$]	-0.1 [-0.7, $+0.5$]
8	2	5×	$+1.3$ [-0.5, $+3.1$]	$+0.04$ [-0.03, $+0.11$]	$+0.8$ [-0.3, $+1.8$]
	3	5.5×	$+2.0$ [$+0.4$, $+3.6$]	$+0.05$ [-0.01, $+0.11$]	$+0.3$ [-0.6, $+1.2$]
	4	6×	$+1.1$ [-0.3, $+2.4$]	$+0.01$ [-0.04, $+0.07$]	-0.0 [-0.9, $+0.8$]
12	2	7×	$+2.7$ [$+0.6$, $+4.7$]	$+0.08$ [$+0.00$, $+0.16$]	$+1.1$ [$+0.1$, $+2.0$]
	3	7.5×	$+1.1$ [-0.8, $+3.1$]	$+0.04$ [-0.03, $+0.11$]	$+0.5$ [-0.4, $+1.5$]
	4	8×	$+0.8$ [-1.0, $+2.6$]	$+0.03$ [-0.04, $+0.09$]	$+0.1$ [-0.9, $+1.1$]
16	2	9×	$+4.6$ [$+2.3$, $+6.9$]	$+0.18$ [$+0.10$, $+0.26$]	$+2.4$ [$+1.4$, $+3.5$]
	3	9.5×	$+3.6$ [$+1.5$, $+5.7$]	$+0.15$ [$+0.08$, $+0.22$]	$+2.0$ [$+1.0$, $+3.0$]
	4	10×	$+3.1$ [$+1.1$, $+5.0$]	$+0.05$ [-0.02, $+0.12$]	$+1.2$ [$+0.2$, $+2.2$]

diversity a rule can buy (Appendix A.1). Free minus required output diversity has a 95% interval above zero in all 48 comparisons at this loss, 33 of them against a required rule at its cap. At a loss of 0.015, where 38 comparisons are at equal reward, 32 intervals exclude zero and 47 of 48 point estimates are positive. The replay reproduces every arm it shares with Table 28 to within 10^{-14}, the required rules keep the champion on every request, and with two survivors greedy and required exact max-sum select the same survivors for all 128,000 request–strength pairs.

Selection time. Density's selection step takes 10.6 to 12.0 μs per request, almost independent of N. Exact max-sum grows with the number of subsets, from 19.9 μs (15 subsets) to 128.6 μs (1,820 subsets), and greedy takes 45 to 127 μs. Every selection step stays below 0.13 ms per request, negligible next to generation.

Scope. With sixteen candidates and two survivors, reward-only pruning itself delivers less reward than best-of-9 at the same budget. The largest cluster holds 76% to 80% of each prompt's images, so rare clusters are often single outliers. Mixture weights are fixed at their full-sample values, and no multiplicity correction is applied.

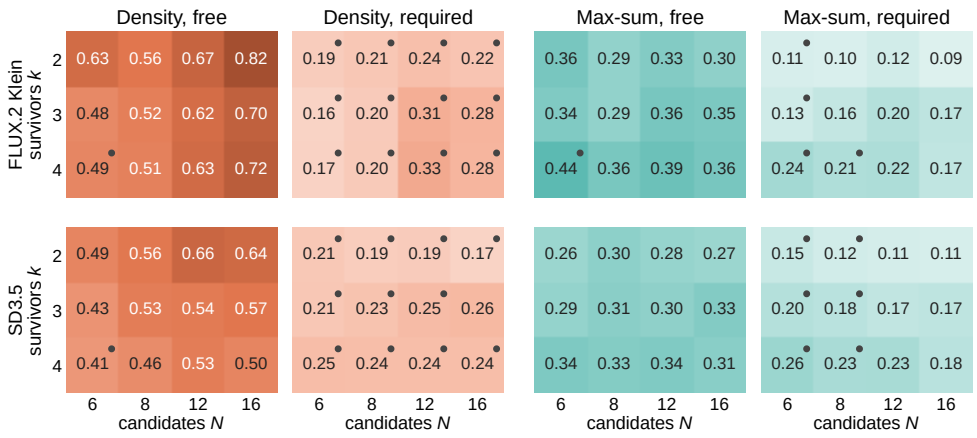

Figure 13: Conversion of set-diversity gain into output diversity for every pool size N and survivor count k on the sixteen-candidate pools, at 0.025 ImageReward below reward-only pruning. The free version converts more than the required one in every cell. A dot marks a rule that cannot lose 0.025 in that cell, shown at its most diverse strength.

H THE CHAMPION CONSTRAINT IN TEXT GENERATION

Lemma 1 uses only the preview ranking and final-reward delivery, so it holds for any generator whose candidates are pruned on previews. We test this with a language model by repeating the champion interventions of Section 6.3; every other step matches the image experiments.

Setup. The prompts are 200 AlpacaEval instructions (Li et al., 2023) from its five source datasets. Qwen2.5-1.5B-Instruct (Yang et al., 2024) writes each response at temperature 1.0 and top-p 0.95, with at most 160 new tokens, a limit that 73% of responses reach. OpenAssistant's DeBERTa-v3-large reward model (Köpf et al., 2023) gives the *final reward* on the complete response and the *preview reward* on its first 40% of tokens. Each request samples $N = 6$ responses, keeps $k = 2$ survivors and delivers the survivor with the higher final reward. The selectors use unit-norm all-mpnet-base-v2 embeddings (Reimers & Gurevych, 2019) of the same prefix that the preview reward scores, with cosine distances d_{ij}, so they see only what is available at the cut. Output diversity, Eq. (1), is measured on the embeddings of the complete responses, over eight independent requests per instruction. Every fifth instruction (40 in all) serves only to set σ_r, σ_R, $\bar{d}$ and $\tau = \bar{d}/4$; the other 160 are evaluated. Selectors, strength grids and intervals follow Appendices F and C.3, with the source datasets as bootstrap strata, and every selector is replayed exactly on the 9,600 cached responses. These distances are not DreamSim distances, so only their signs and orderings compare with the image results.

Lemma 1 holds exactly. With the 40% preview, 39.5% of requests are locked, against 49% to 72% on the main image benchmark. A selector's *winner-change rate* is the share of requests on which it delivers a different response from reward-only pruning. Lemma 1 bounds this rate by the unlocked share, 60.5%, for every champion-preserving selector. None of our 26 champion-preserving settings exceeds the bound: champion-required density at ten strengths, champion-required exact max-sum at eight and champion-first greedy max-sum at eight, the last two identical at $k = 2$ (Appendix A.1). The highest reaches 38.7%, while free settings reach 70.3% because they can also change locked requests. The oracle preview, the final reward divided by σ_R, locks every request; it keeps the 40%-prefix embeddings, unlike the 100% row of Table 29, which embeds complete responses. All 26 champion-preserving settings then deliver reward-only pruning's response on every request, with no difference in reward or diversity, whereas free settings still change the delivered response on up to 66.4% of requests and raise output diversity from 0.200 to 0.278.

Table 29: The champion constraint in text under nine previews of the same cached responses. Each prefix preview selects on the embedding of the prefix it scores; the 100% row and the two other-model previews score and embed the complete response. Locked: share of requests whose preview champion is also the final winner. Winner change: highest share of requests on which a setting delivers a different response from reward-only pruning. Pres. covers reward-only pruning and the 26 champion-preserving settings; Free covers the 18 free settings. Diversity range: spread of output diversity over the same settings. Matched gap: free minus champion-required output diversity at equal mean final reward, pooled over density and exact max-sum, with 95% prompt-bootstrap intervals. When the preview is the final reward, every required setting coincides with reward-only pruning. Diversity is mean cosine distance between complete-response embeddings, not DreamSim.

	Locked	Winner change (%)		Diversity range		Matched gap
Preview	(%)	Pres.	Free	Pres.	Free	[95% interval]
Prefix of the response, scored by the final reward model						
10%	22.3	48.8	72.0	0.017	0.042	+0.0022 [−0.0007, +0.0041]
20%	29.1	46.2	73.5	0.023	0.066	+0.0039 [+0.0012, +0.0062]
30%	33.3	40.9	71.2	0.023	0.068	+0.0029 [+0.0012, +0.0052]
40% (main)	39.5	38.7	70.3	0.025	0.076	+0.0015 [−0.0002, +0.0032]
60%	51.3	30.1	68.0	0.019	0.079	+0.0005 [−0.0011, +0.0021]
80%	57.7	28.0	67.6	0.011	0.085	+0.0032 [+0.0017, +0.0046]
100% (final reward)	100.0	0.0	66.7	0.000	0.090	—
Complete response, scored by another reward model						
DeBERTa-v3-base	40.6	37.5	72.7	0.029	0.100	+0.0021 [−0.0001, +0.0046]
ELECTRA-large	41.6	36.2	71.2	0.033	0.101	−0.0002 [−0.0034, +0.0027]

Free versus champion-required at equal final reward. For each champion-required strength, we evaluate the free curve at the same mean final reward by linear interpolation between neighbouring free strengths, and average the diversity differences over the selector's strengths; exact per-request mixtures (Appendix C.3) change the result by less than 0.0001. At the main preview, the free versions deliver +0.0015 [−0.0002, +0.0032] more diversity, pooled over both selectors, an interval that covers zero; density (+0.0016) and exact max-sum (+0.0014) contribute about equally. Across the previews of Table 29, the gap is positive at seven of the eight imperfect previews, and its interval lies above zero at the 20%, 30% and 80% prefixes.

Preview quality sets the locked share. We re-scored the cached responses under the nine previews of Table 29. Seven score a prefix of 10% to 100% of the tokens, where the 100% prefix is the final reward itself. Two score the complete response with another OpenAssistant reward model, while the large model still picks the delivered response. The locked share rises from 22.3% to 57.7% over the partial prefixes. For all eight previews other than the final reward, the prediction of Proposition 1, computed from Kendall's rank correlation as in Appendix F.3, lies within 0.015 of the measured locked share. As more requests are locked, champion-preserving selectors can change less: their highest winner-change rate stays between 0.61 and 0.66 times the unlocked share, and the range of output diversity they reach peaks at 0.025 at the 40% prefix and falls to 0.011 at 80% and to zero at the final reward. Free selectors change the delivered response on 67% to 74% of requests on every preview, and the diversity range they reach grows from 0.042 at the 10% prefix to 0.090 at the final reward, as longer prefixes show them more of each response.

The locked share alone does not set the gap. The matched gap does not follow the locked share. It is +0.0032 [+0.0017, +0.0046] at the 80% prefix, where 57.7% of requests are locked, and +0.0005 [−0.0011, +0.0021] at the 60% prefix. The image ledger of Appendix F.4 shows the same pattern. The locked share fixes which requests a champion-preserving selector cannot change; the gap also depends on how often the free selector changes a locked request, and on what it delivers there.

Density and exact max-sum. At equal mean final reward, the two free selectors deliver the same mean diversity, density minus exact max-sum being +0.0020 [−0.0014, +0.0052], and density covers 0.09 [0.02, 0.15] more content clusters among the eight delivered responses, as on images (Sec-

tion 6.4). The clusters are eight average-linkage clusters of each instruction's 48 cached complete responses, fixed for all selectors.

Scope. The study uses one generator, one final reward model, one instruction bank and one embedding model, with one delivered response per request. With more survivors the gap in text vanishes: it is $+0.0003$ $[-0.0006, +0.0013]$ with three of six kept, -0.0002 $[-0.0010, +0.0005]$ with four and -0.0003 $[-0.0007, -0.0001]$ with five, where the required version delivers slightly more. The oracle collapse of every champion-preserving setting holds exactly at every survivor count. The other-model previews change the preview model, not the final reward model.

I HUMAN STUDY

This appendix describes the blinded studies of Section 6.5 on SD3.5 and FLUX.2 Klein: their material, questions, raters and analysis. It also shows how the human variety judgements relate to the change in DreamSim. It supports the claim that raters see density's added variety at each backbone's development strength and judge quality equal on almost all prompts.

Material. Each rater received 126 pairs of eight-image strips built from cached SD3.5 images of the main benchmark; no image was generated for the study. The 100 main pairs compare density ($\beta = 1/16$) with same-schedule reward-only pruning on 100 prompts, sampled by PartiPrompts category with a fixed seed. Both strips show the eight independent requests of the same prompt, selected from the same candidates. Twenty control pairs compare one uniformly drawn native candidate per request with reward-only pruning, and six null pairs show the same strip twice. Images are shown at 200 pixels in a 2×8 grid with the prompt above. Row order and pair order are randomised per rater with a fixed seed, and the keys were kept outside the rater folders. The Klein pack repeats this design on the same 100 prompts and the same control and null prompts, with Klein's cached images and density at $\beta = 1/32$; its selections reproduce the Klein column of Table 1.

Questions and raters. Raters answered three questions in a fixed order without going back. Variety: "Which row shows more variety across the eight images?" (A, B or same). Quality: "Which row has better-looking images overall?" (A, B or same). Validity: "Does either row contain an image that does not match the prompt?" (A, B, both or neither). The same two people rated both packs, SD3.5 first, without knowing which method produced which row. A vision–language model (GPT-6 Astra) answered the same questions on the SD3.5 pack; we report it separately and never pool it with the human raters.

Analysis. Each judgement scores 1 for density, $\frac{1}{2}$ for "same" and 0 for pruning. Scores are averaged over the two raters within a prompt, and intervals resample prompts. On SD3.5, density's variety score is 72.5% $[65.5\%, 79.0\%]$, and its quality score is 49.25% $[47.75\%, 50.5\%]$. Both raters call quality equal on 94 of 100 prompts; on one prompt both prefer pruning, and on none do both prefer density. On the 70 prompts where both raters give a decisive variety answer, 54 favour density, 13 favour pruning and 3 conflict, a density share of 0.77 $[0.66, 0.85]$. Agreement on variety is Cohen's $\kappa = 0.77$. Twelve main pairs deliver identical strips under both methods, and both raters answered "same" on all of them and on all six null pairs. Both raters flag only the density row as containing an off-prompt image on 2 prompts and only the pruning row on none; on 4 they flag both rows. In the control pairs, native sampling is judged more varied than pruning on 10 prompts against 3, with 3 conflicts. The model judge agrees with the human consensus on 69% of variety calls and 89% of quality calls; it answers "same" more often but points the same way.

Klein. On Klein, density's variety score is 70.3% $[62.5\%, 77.5\%]$, and its quality score is 50.0% $[49.25\%, 50.75\%]$. Both raters call quality equal on 98 of 100 prompts; on each of the other two, one rater calls it equal. On the 84 prompts where both raters give a decisive variety answer, 56 favour density, 18 favour pruning and 10 conflict, a density share of 0.67 $[0.56, 0.76]$. Agreement on variety is Cohen's $\kappa = 0.71$. Nine main pairs deliver identical strips under both methods, and both raters answered "same" on all of them and on all six null pairs. No prompt has an off-prompt image flagged by both raters in only one row; on 1 they flag both rows. In the control pairs, native sampling is judged more varied than pruning on 13 prompts against 5, with 2 conflicts. Neither rater favours the top or the bottom row (binomial $p = 0.18$ and 0.77).

Table 30: Human variety judgements against the output-diversity metric. Density against same-schedule reward-only pruning on the same 100 main-benchmark prompts, SD3.5 at $\beta = 1/16$ and FLUX.2 Klein at $\beta = 1/32$, rated blind by the same two people. Prompts are split at the quintiles of ΔDreamSim, the change in mean pairwise DreamSim distance over the eight delivered images the raters actually saw. n counts judgements (two raters $\times$ the prompts in the bin, 20 prompts per bin except the first two, which split at $\Delta\text{DreamSim} = 0$); each judgement scores 1 for density, 0 for pruning and $\frac{1}{2}$ for "same". Intervals average the two raters within each prompt and resample prompts (20,000 bootstraps). Section 6.5 gives the pooled scores.

SD3.5			FLUX.2 Klein		
ΔDreamSim range	n	Density share [95%]	ΔDreamSim range	n	Density share [95%]
-0.0290 to 0.0000	38	0.382 [0.211, 0.566]	-0.0211 to 0.0000	38	0.408 [0.211, 0.618]
0.0000 to 0.0046	42	0.548 [0.440, 0.655]	0.0000 to 0.0043	42	0.643 [0.500, 0.786]
0.0046 to 0.0192	40	0.800 [0.650, 0.925]	0.0043 to 0.0119	40	0.613 [0.463, 0.750]
0.0192 to 0.0311	40	0.950 [0.875, 1.000]	0.0119 to 0.0261	40	0.875 [0.725, 1.000]
0.0311 to 0.2855	40	0.938 [0.875, 0.988]	0.0261 to 0.1204	40	0.963 [0.900, 1.000]

Relation to DreamSim. Table 30 splits the prompts by the change in DreamSim between the two strips the raters saw. The variety preference rises with that change: raters favour pruning's strip where the metric does (0.38), are near even around zero (0.55), and prefer density's strip in 80% to 95% of judgements once it adds more than 0.005. Klein shows the same pattern: 0.41 where the metric favours pruning, 0.61 to 0.64 for gains up to 0.012, and 0.88 to 0.96 above. In a logistic model of the SD3.5 votes, the DreamSim change predicts preferences, and adding the change in distinct modes does not improve the fit (likelihood-ratio $\chi^2 = 0.47$, $p = 0.49$). Adding the DreamSim change to a model of distinct modes improves it strongly ($\chi^2 = 22.2$). DreamSim is therefore the better predictor of what raters see. The link is measured on each backbone separately, and its bins should not be compared across backbones, because DreamSim differences are not comparable across them.

J QUALITATIVE EXAMPLES

This appendix shows complete sets of delivered images, so that the reader can see what a change in output diversity looks like. The examples illustrate the quantitative results of Section 6; they are not evidence of average effect size.

How the examples were selected. Every gallery shows all eight independent requests of a prompt in their original order, every image is an unedited cached output, and prompts were chosen after the outcomes were known. Figure 14 comes from the SD3.5 human-study prompts at the development strength. A prompt qualified if both raters called density's strip more varied and the quality the same, the model judge also called it more varied, its DreamSim change fell in the third or fourth quintile of the 100 prompts, and its distinct-mode gain was one or two. Twelve prompts qualified; we show one that is also a control pair, so that native sampling can be shown as well. Figure 15 uses density with $\beta = 1/8$ and max-sum with $\lambda = 1.6$ (development strengths on SD3.5: $1/16$ and 0.8). We screened the cached DIMCIM banks of Klein and SD3.5 for prompts on which both selectors change the winner on at least three of eight requests and lose at most 0.15 mean ImageReward, shortlisted the six prompts per backbone with the largest smaller DreamSim gain of the two selectors, and picked two by visual inspection. On them density gains 0.078 and 0.069 DreamSim and max-sum 0.052 and 0.043, at ImageReward losses of 0.017 to 0.068, well above the average: over all 100 SD3.5 DIMCIM prompts the same settings give (ImageReward, DreamSim) of $(1.2415, 0.2453)$ for reward-only pruning, $(1.2167, 0.2598)$ for max-sum and $(1.2269, 0.2597)$ for density.



Figure 14: Delivered images across eight independent requests for "an F1 race car" on SD3.5. Rows show native sampling, reward-only pruning and density ($\beta = 1/16$), with columns in request order. Pruning and density select from the same candidates at the same budget. DreamSim is 0.1746 for pruning and 0.2012 for density, with two and three distinct modes, respectively. Both raters judge density more varied and the rows equal in quality. This illustrative prompt was selected post hoc from twelve qualifying cases in the middle diversity-change quintiles; all images are unedited cached outputs.

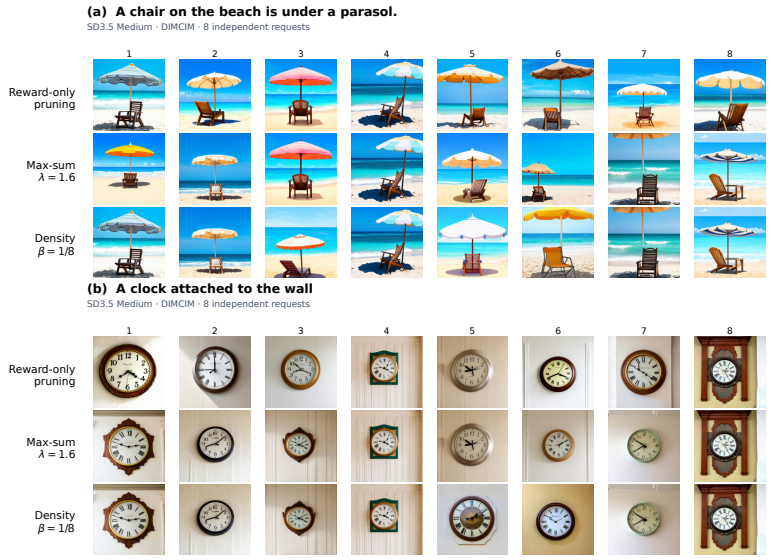

Figure 15: Eight independent SD3.5 requests for two DIMCIM prompts, in order: reward-only pruning, max-sum ($\lambda = 1.6$) and density ($\beta = 1/8$), two settings with the same DreamSim over the suite. All rows use the same candidates, the $6 \to 2$ schedule and final-reward argmax; prompts were selected after the outcomes were known, by the rule stated above, and images are unedited.