GRID-BASED REPULSIVE SAMPLER FOR DISTRIBUTION REBALANCING IN GENERATIVE POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training a generative model begins by reducing a large candidate pool to a small fine-tuning set. Near-duplicates are removed above a similarity threshold and over-represented regions are capped, once, before training, and neither decision can be revisited afterwards. We propose the Grid-based Repulsive Sampler (GridReS), which rebalances the training distribution and keeps the pool intact. GridReS constructs a uniform grid over a low-dimensional projection of a pretrained embedding space. During training, occupancy-aware sampling redistributes probability mass across cells, while intra-cell repulsion scores candidates against a memory of recent draws from the surrounding neighbourhood. The partition supplies the locality that keeps repulsion at a bounded per-draw cost; repulsion controls the redundancy inside a region, which weighting over regions cannot reach. At coverage identical to random sampling, GridReS cuts cross-source near-duplicate co-exposure by 50% on the fine-tuning pool of Kandinsky-6 Lite, a text-to-audio-video model, and by 25% on OpenVid-100k. When Kandinsky-6 Lite is fine-tuned as a single job on the full pool, GridReS improves VBench imaging quality over random sampling by 1.84 points, beyond run-to-run variance, and trained per domain it matches the deployed deduplication pipeline at equal optimizer steps without discarding any data. Code is available at `https://github.com/94yvwj2hrj-eng/gridres`.

1 INTRODUCTION

Post-training a video generative model begins by reducing a candidate pool to a supervised fine-tuning (SFT) set roughly an order of magnitude smaller or more. Cosmos pre-trains on about 2×10^8 curated clips and fine-tunes on domain sets totalling about 1.7×10^7 videos (NVIDIA et al., 2025a), HunyuanVideo filters through cascaded stages that each remove a large part of their input before building roughly 1M SFT samples by human annotation (Kong et al., 2024), and the Kandinsky-6 pipeline studied here reduces its pool 8.3:1. Two operations recur across these pipelines, the removal of near-duplicates by embedding or hash similarity and the capping or down-sampling of over-represented regions (Polyak et al., 2024; Kong et al., 2024; Team Seaweed et al., 2025; Gao et al., 2025; Chen et al., 2025; Team Wan et al., 2025). Both are fixed before training and applied once, and a clip removed then is unavailable to every later stage.

The cost of those thresholds is not measured. Among these works only Stable Video Diffusion reports a controlled comparison, preferring a model trained on a filtered subset over one trained on unfiltered data of the same origin (Blattmann et al., 2023). That shows filtering helps at a fixed pipeline, not that the discarded clips had no value, and recent work finds that quality-imbalanced or uncurated training data, used with quality-aware training, can match or surpass a curated subset (Luo et al., 2026; Liang et al., 2026). Methods that reweight instead of deleting do so by cluster (Shao et al., 2024), by concept within each batch (Ghosh et al., 2026) or per sample (Zhao et al., 2026; Pais et al., 2025), and near-duplicates fall inside one cluster or concept by construction, where reweighting at that level cannot separate them.

We propose the Grid-based Repulsive Sampler (GridReS), an online sampler that replaces near-duplicate removal and per-region capping with a policy over the full pool (Figure 1). Each draw picks a cell and then a clip within it, and each step contributes one mechanism.

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.

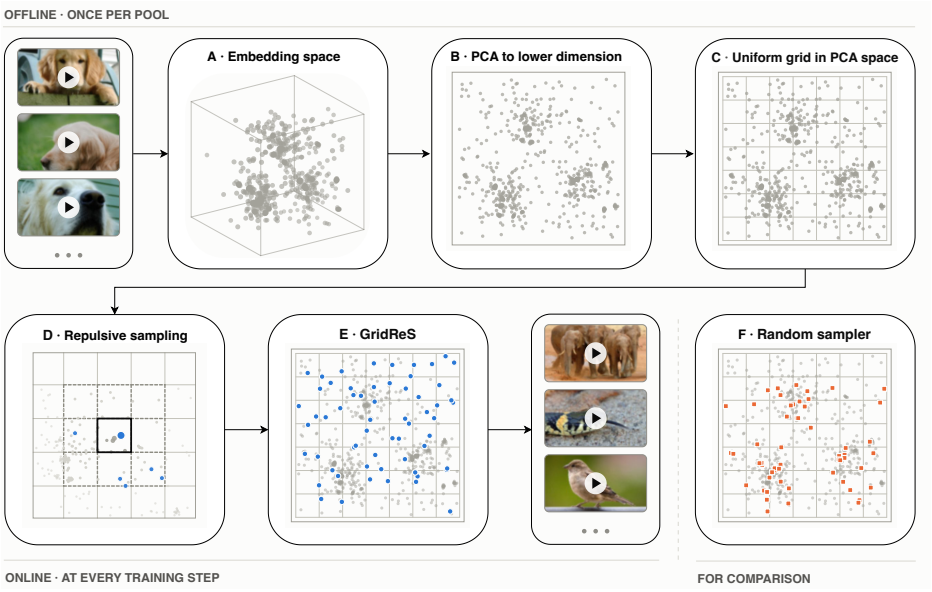


Figure 1: GridReS at a glance. Offline, a frozen encoder embeds every clip and a uniform grid partitions a principal component analysis (PCA) projection (Hotelling, 1933) of the embeddings. At every draw a cell is drawn with weight n_c^α, and a candidate within it is scored against a queue shared across the cell's 3^d neighbourhood. Panel F is uniform sampling; Figure 2 gives the details.

- **Grid partition in embedding space.** A PCA projection of the cached embedding is cut by a uniform grid, so a cell can be drawn with probability proportional to its occupancy raised to α, which sets the budget of a region independently of its size. The grid also supplies locality. Near-duplicate pairs fall almost entirely within a neighbourhood of adjacent cells (Appendix B), so a candidate can be scored against a bounded memory, not the whole pool.
- **Intra-cell repulsion.** Each cell keeps a memory of the most recent draws from its neighbourhood, and a candidate is scored by its distance to the nearest of them, so clips close to a recent draw are served later.

Nothing is removed, so a clip suppressed at one point in training remains available later. At coverage identical to random sampling, GridReS cuts cross-source near-duplicate co-exposure by 50% on the Kandinsky-6 SFT pool and by 25% on OpenVid-100k (Nan et al., 2025). On VBench (Huang et al., 2024), when Kandinsky-6 Lite, a text-to-audio-video model, is fine-tuned as a single job that runs 0.60 epochs over the whole Kandinsky-6 SFT pool, GridReS improves imaging quality over random sampling by 1.84 points, beyond run-to-run variance, with gains in aesthetic quality and subject consistency that exclude zero but not that variance. Trained as eleven per-domain jobs of 2.4 to 36.5 epochs each, the undeduplicated pool matches the curated pipeline (Figure 5). On OpenVid-100k, whose source is already filtered for quality, it matches random sampling on most dimensions, within that pool's seed variation, while drawing near-duplicates together less often (Section 4.3). Appendix A reviews related work.

2 Method

Setup and notation. The candidate pool $\mathcal{P}$, which we call the Kandinsky-6 SFT pool, holds the $N = 67{,}175$ clips in 11 domains that supervised fine-tuning of Kandinsky-6 Lite, a text-to-audio-video (T2AV) model, draws from before deduplication, and it is not public (Appendix D). OpenVid-100k, a 98,004-clip subset of OpenVid-1M (Nan et al., 2025) taken in whole shards, mimics these post-training conditions in public, since its source is already filtered for aesthetics, motion and clarity. A frozen encoder, Qwen3-VL-Embedding-8B (Li et al., 2026), chosen from six candidates for the most diverse space and the lowest near-duplicate rate (Appendix E), maps each clip x once and offline to a unit vector $\phi(x) \in \mathbb{R}^{4096}$. Redundancy is measured on the near-duplicate graph

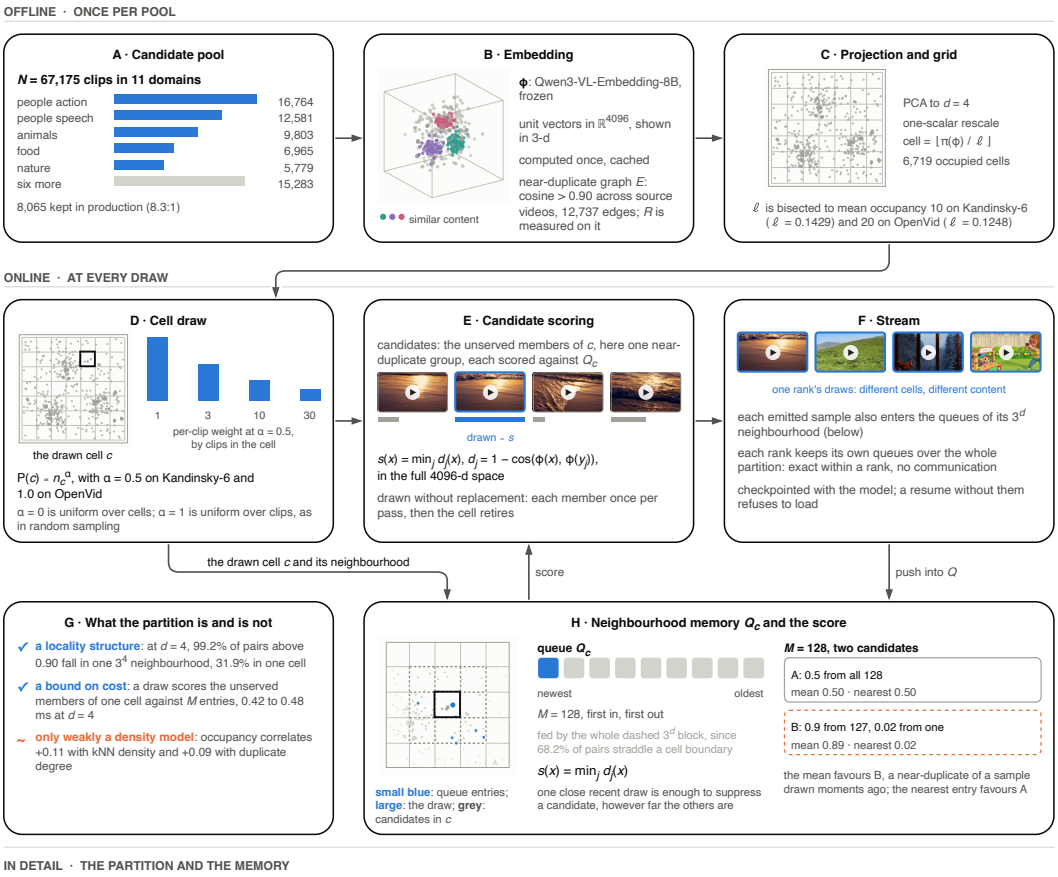


Figure 2: GridReS in detail, expanding Figure 1: the offline pass (A–C), the online draw (D–F), what the partition is and is not (G), and how the neighbourhood memory scores a candidate (H), with the worked case of Section 2.3. Panels B–D and H reuse the illustrative points and grid of Figure 1; the frames in E and F are OpenVid-100k clips.

$E = \{(i,j) : i < j,\ v(i) \neq v(j),\ \langle \phi(x_i), \phi(x_j) \rangle > \tau\}$ with $\tau = 0.90$ and $v(i)$ the source video of clip i, computed exactly over all pairs ($|E| = 12{,}737$). Four in five of the 64,312 pairs above τ are two scenes of one source video and are excluded. For a sampler with per-draw marginal p,

$$R(p) \;=\; 2 \sum_{(i,j)\in E} p_i\, p_j \tag{1}$$

is the probability that two independent draws form a near-duplicate pair, reported as the ratio $\rho(p) = R(p)/R_{\text{unif}}$ to uniform sampling. Appendix B derives what a sampler can and cannot do to R.

2.1 Partition

A PCA projection π to $\mathbb{R}^d$, centred at the median and rescaled by a single scalar, is followed by a uniform grid of side ℓ that assigns clip x to the cell $c(x) = \lfloor \pi(\phi(x))/\ell \rfloor$. We use $d = 4$. Appendix B.5 gives the case for a small d, and Section 5 compares values. Because d, ℓ and N fix the occupancy distribution jointly, ℓ is not portable, so it is bisected per pool to a target mean occupancy, 10 on the Kandinsky-6 SFT pool and 20 on OpenVid-100k ($\ell = 0.1429$ and 0.1248). The grid ignores domain labels. It is fitted once on the whole pool and reused by every per-domain job, with the occupancy n_c of cell c counting only that job's rows. Only occupied cells are stored, and a draw costs one lookup, M distance evaluations for each unserved member of the drawn cell (Section 2.3), so at most Mn_c, and 3^d queue writes.

2.2 CELL WEIGHTING

A cell is drawn with probability $P(c) \propto n_c^\alpha$ over the occupied cells. At $\alpha = 0$ every occupied cell is equally likely. At $\alpha = 1$ every clip is, since the marginal equation 4 reduces to $1/N$, and $\rho = 1$ exactly. Unlike a threshold, α is continuous, applied at sampling time, and can be changed without discarding data. We use $\alpha = 0.5$ on the Kandinsky-6 SFT pool and 1.0 on OpenVid-100k. The weighting lowers R only where occupancy tracks a cell's share of E (Appendix B.4). At $d = 4$ it does so weakly on the Kandinsky-6 SFT pool, where the grid alone lowers R by 25% at $\alpha = 0.5$, and not on OpenVid-100k, where the grid alone raises R at every $\alpha < 1$ (Section 6). The weighting also concentrates draws, which makes a bounded queue more likely to hold a candidate's near-duplicate. On the Kandinsky-6 SFT pool it raises effective diversity, with the Vendi score (Friedman & Dieng, 2023) of the draws at 1.146 relative to random at $\alpha = 0$ against 1.018 at $\alpha = 1$, and on OpenVid-100k it lowers it (Appendix H.2).

2.3 INTRA-CELL REPULSION

Cell weighting decides how often a region is visited, but redundancy depends on which of its members comes back. Two clips that a semantic deduplicator would merge land in the same cell or in neighbouring ones by construction, so drawing uniformly inside a cell keeps returning the same dense pocket. GridReS makes that choice depend on what was drawn recently. Each cell c reads a queue Q_c of the last $M = 128$ draws from its 3^d neighbourhood, and every draw is pushed into the queues of all 3^d cells around it, since at the geometry of Appendix B.5 most near-duplicate pairs straddle a cell boundary. A candidate x is scored by its distance, in the full embedding space, to the nearest entry of the queue,

$$s(x) \;=\; \min_{y_j \in Q_c} d_j(x), \qquad d_j(x) \;=\; 1 - \langle \phi(x), \phi(y_j) \rangle \tag{2}$$

where an empty queue scores 1 and the oldest entry is evicted first. A small bonus of $0.1/(1 + n_x)$, with n_x the number of times x has already been drawn, keeps every member drawable.

We take the nearest entry because an average lets many distant entries hide a close one. With $M = 128$, a candidate 0.9 away from 127 entries and 0.02 from the last averages 0.89 and is preferred to one 0.5 away from all of them, although it nearly repeats a clip drawn moments earlier (Figure 2). A harmonic mean rescues this case only for a short queue: it reads 0.17 at $M = 10$ but 0.67 at $M = 128$. Pooled over the other settings, the nearest entry lowers R relative to the harmonic mean by 8.8% on the Kandinsky-6 SFT pool and 1.5% on OpenVid-100k (Appendix H).

The draw inside a cell must also cover the cell, because a clip that is never served lowers R for the wrong reason. GridReS picks among the cell's unserved members with probability proportional to their scores, retires the cell once every member has been served, and refills the cells only when all of them are spent. Repulsion therefore changes only the order in which members appear, and each stream passes over the pool exactly once per epoch of its own draws (Section 4.2). On the Kandinsky-6 SFT pool this reordering alone lowers R by 24% at $\alpha = 0$ and by 40% at $\alpha = 1$. Drawing from the full membership at every visit would let repulsion postpone a clip indefinitely; even uniform draws with replacement reach only $1 - (1 - 1/n_c)^{n_c}$ of a cell in n_c visits, which tends to 63.2% as n_c grows. Each data-parallel rank runs its own stream over the whole pool, with its own queues, which are checkpointed with the model (Appendix C).

3 EXPERIMENTS

Table 1 summarises the setup. Both pools are used for training and for parameter selection, and each reports the draw distribution, its Vendi diversity and VBench of trained models. Target occupancy and α are set on each pool separately, and the dimension is $d = 4$ on the Kandinsky-6 SFT pool (Section 5). The trained OpenVid-100k models use an earlier configuration, $M = 32$ with a harmonic-mean score and $d = 3$, the best dimension on that pool at this queue length and score (Appendix H.2), while every draw-distribution result on both pools uses the configuration of Section 2. Every run fine-tunes Kandinsky-6 Lite for 5,000 optimizer steps on 8 GPUs with one clip per GPU per step, so it draws 40,000 clips (the OpenVid-100k runs continue to 10,000 steps), and two schedules spread those draws differently. A *single job* trains one model on the whole Kandinsky-6

Table 1: Experimental setup. Model-level comparisons use the Kandinsky-6 SFT pool unless marked, and Appendix I gives the full configuration.

Pools	Kandinsky-6 SFT pool, 67,175 candidate clips with audio in 11 domains, before deduplication, not public. OpenVid-100k, 98,004 clips of OpenVid-1M without audio, public, a stand-in for SFT data.
Embedding	Qwen3-VL-Embedding-8B, 4096 dimensions, computed once offline, chosen from six candidate encoders (Appendix E). Near-duplicate graph at cosine > 0.90.
Model	Kandinsky-6 Lite, 3B parameters, T2AV. 5,000 steps on 8 GPUs, one clip per GPU per step, learning rate 10^{-5} after 500 warm-up steps, EMA decay 0.999. On OpenVid-100k the audio loss is off and the audio branches are frozen.
Schedules	*Single job*: one model on the whole pool, 0.60 epochs. *Per domain*: eleven jobs, one per domain, souped, 2.4 to 36.5 epochs of a domain. The OpenVid-100k runs are single jobs, 0.41 epochs at 5,000 steps and 0.82 at 10,000.
GridReS	$d = 4$, $M = 128$, nearest-entry score, drawn in proportion to score without replacement within cells; target occupancy 10 and $\alpha = 0.5$ on the Kandinsky-6 SFT pool, 20 and $\alpha = 1$ on OpenVid-100k. Trained OpenVid-100k models and the per-domain model marked ($d = 3$, $M = 32$) use $d = 3$, $\ell = 0.10$, $M = 32$ and a harmonic-mean score.
Baselines	Random permutation of the full pool. Production curation, deduplication and per-domain capping to 8,065 clips, then random. ClusterClip (Shao et al., 2024). For the draw distribution only, CABS-DM (Ghosh et al., 2026), SemDeDup (Abbas et al., 2023) and a facility-location coreset.
Evaluation	VBench imaging quality, aesthetic quality, subject consistency, background consistency and dynamic degree on 500 prompts with shared seeds, paired bootstrap (Efron & Tibshirani, 1993) over prompts with 4,000 resamples, read against the seed floor of Table 3. R from simulated draws at budgets $B \in \{0.10, 0.25, 0.50\}N$ and from training draw logs.

SFT pool for 0.60 epochs, counting the draws of all 8 ranks; since each rank runs its own stream, a stream makes 0.074 passes (Section 4.2). The *per-domain* schedule, which the production pipeline uses, trains eleven jobs, one per domain label, and merges them by souping, that is, by averaging their weights (Wortsman et al., 2022), as Kandinsky 5.0 does with its SFT soups (Kandinsky Lab, 2025), so each job runs 2.4 to 36.5 epochs of its domain, or 0.30 to 4.6 passes per stream, and a clip is drawn 6.6 times on average. A curator deletes clips before drawing uniformly, so its coverage is bounded by its keep fraction, and every value of R is reported beside its coverage. Quality is measured on generations and not on training loss. Each configuration generates 500 clips from one fixed prompt set with shared seeds, so every comparison is paired, and the video stream is scored with VBench (Huang et al., 2024) on five dimensions that do not read the prompt. Its prompt-conditioned dimensions expect VBench's own short prompts or at most 77 text tokens, and our prompts are long captions in the style of the training data that could not be shortened without changing the comparison. A difference counts only if its interval lies beyond the seed floor, the change a retrain with a new seed produces, which alone passes the prompt-level bootstrap (Appendix F).

4 RESULTS

Distributional results come from simulated runs on both pools, each paired against a random permutation at the same budget, with run counts in the captions and Appendix H. R is aggregated as the geometric mean of the paired ratio with a bootstrap interval, and below R denotes this ratio; runs where it is exactly zero are dropped and counted (Appendix H).

4.1 THE INDUCED DRAW DISTRIBUTION

GridReS halves co-exposure at unchanged coverage. At the pool coverage of random sampling, GridReS cuts cross-source near-duplicate co-exposure by 50.3% on the Kandinsky-6 SFT pool and by 24.6% on OpenVid-100k (Table 2), the largest reduction of any sampler at that coverage. CABS-DM leaves R at random sampling's level on both pools, and inside a per-domain job its concepts, the domain labels, reduce to one, so there it draws as random sampling does (Appendix F.5); the grid without repulsion reaches 0.749 on the Kandinsky-6 SFT pool. The projection dimension, the

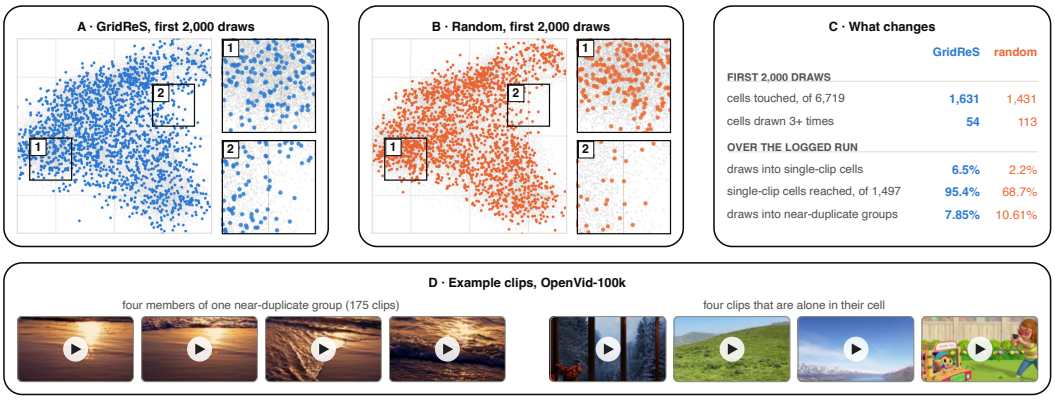


Figure 3: Where the draws land. **(A, B)** The first 2,000 draws of GridReS and of random sampling in the single-job runs on the Kandinsky-6 SFT pool, over the first two PCA components, with the dots enlarged for legibility. Beside each map the same two regions are enlarged: in the dense region 1 random sampling returns to the same clusters, and the sparse edge 2 is reached more often by GridReS. **(C)** Counts over the first 2,000 draws and shares over the whole logged run; a near-duplicate group is a connected component of the cosine > 0.94 graph. **(D)** Examples of the two kinds of content that panel C counts, members of one near-duplicate group and clips alone in their cell, from OpenVid-100k; Appendix J shows more.

Table 2: Induced draw distribution, pooled over $B \in \{0.10, 0.25, 0.50\}N$, as the geometric mean of the paired ratio to a random permutation with zero-R runs dropped. Both grid rows use the configuration of Section 2, with $\alpha = 0.5$ on the Kandinsky-6 SFT pool and $\alpha = 1$ on OpenVid-100k; grid only keeps the cell weighting and draws uniformly inside the cell. Bold marks the lowest R/R_{rand} in the first block and the highest source coverage. Run counts are in Appendix H.

	Kandinsky-6 SFT pool (67,175)			OpenVid-100k (98,004)	
Method	$R/R_{\text{rand}} \downarrow$	pool cov.	source cov.	$R/R_{\text{rand}} \downarrow$	pool cov.
Samplers at random sampling's coverage					
random (permutation)	1.000	0.283	0.393	1.000	0.283
CABS-DM (Ghosh et al., 2026)	1.003	0.283	0.392	1.004	0.283
grid only, $\alpha = 0.5$ / 1	0.749	0.283	0.402	1.024	0.283
GridReS, $\alpha = 0.5$ / 1	**0.497**	0.283	**0.409**	**0.754**	0.283
ClusterClip draws clusters uniformly; curators reduce the pool first					
ClusterClip (Shao et al., 2024)	0.450	0.191	0.312	1.144	0.241
SemDeDup (Abbas et al., 2023)	0.104	0.104	0.204	0.174	0.283
facility location (Wei et al., 2015)	0.101	0.113	0.204	—	—

cell side, the nearest-entry score and the queue length each contribute (Section 5 and Appendices G and H).

It reaches more source videos and more of the sparse regions. On the Kandinsky-6 SFT pool GridReS draws from 4% more source videos than random sampling at the same budget, 0.409 of them against 0.393, the highest in Table 2. In the single job's first 2,000 draws it reaches 1,631 cells against 1,431 for random sampling, and draws 54 cells three or more times against 113 (Figure 3). Over its draw log it sends three times as many draws into single-clip cells, reaches 95.4% of them against 68.7%, and lands 7.85% of draws in near-duplicate groups against 10.61%.

Effective diversity rises. The Vendi score of the draws exceeds random sampling's by 10.2% on the Kandinsky-6 SFT pool, on an interval of [9.5, 10.9], and by 1.1% on OpenVid-100k, where $\alpha = 1$ leaves every clip equally likely (Appendix H.2).

Repulsion adds to the grid at every setting. Paired on everything else, repulsion lowers R by 24 to 40% on the Kandinsky-6 SFT pool and by 14 to 26% on OpenVid-100k across $\alpha \in [0, 1]$, most where the weighting does least (Figure 4a and Appendix H.2).

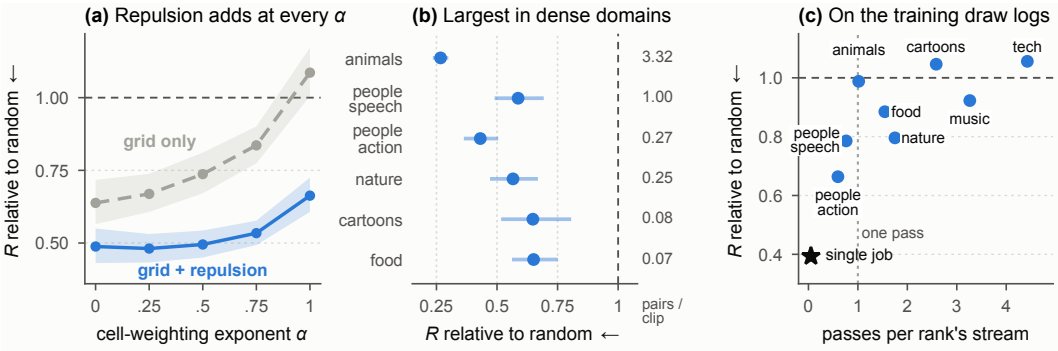


Figure 4: **(a)** R over α on the Kandinsky-6 SFT pool, for the grid alone and with repulsion, 1,440 runs with 95% bands. **(b)** Per-domain R of GridReS, ten seeds, with 95% intervals, the domains ordered by near-duplicate pairs per clip (right column), same-source pairs included. **(c)** R on the training draw logs of the eight logged per-domain jobs, about 80,000 draws each, against the passes per rank's stream, and the single job over its first 27,200 draws; the dotted line marks one pass, where R equals 1 for every covering sampler by construction. The dashed line is random sampling.

Lower R comes only with less of the pool. On the Kandinsky-6 SFT pool the three lowest R belong to ClusterClip, SemDeDup and facility location, which draw 10 to 19% of the pool against 28% for random sampling and reach 20 to 31% of its source videos against 41% for GridReS.

4.2 Where the effect is largest

The effect is largest where near-duplicates are densest. Across six domains R runs from 0.267 on *animals* to 0.651 on *food*, with Spearman $r_s = -0.83$ against near-duplicate pairs per clip counting same-source pairs (Figure 4b) and -0.71 against the cross-source pairs R counts. *Animals*, the densest domain on both counts, gains the most.

R reads the spacing of near-duplicates within a pass. Each rank's stream draws every clip once per pass, so at a whole number of passes every covering sampler, random sampling included, has drawn every clip equally often and R equals 1 by construction. R therefore measures how far apart near-duplicates are drawn within a pass. There GridReS reads 0.448, 0.482 and 0.560 offline at budgets of 0.10, 0.25 and 0.50 of a pass, and 0.66 and 0.79 on the training draw logs of the two logged per-domain jobs below one pass (Figure 4c). Repulsion orders every pass in the same way, so the spacing it produces is not confined to the first pass, although R, which counts co-draws over all draws so far, cannot show it once a pass completes. The per-domain models, trained for up to 4.6 passes per stream, match the curated pipeline on VBench (Section 4.3).

4.3 Trained models

Trained as a single job, GridReS beats random sampling. On the same raw pool, with only the sampler changed, it gains 1.84 on imaging quality, on an interval of [1.24, 2.44] beyond the seed floor, gains 0.0060 on aesthetic quality and 0.38 on subject consistency, and ties on the other two dimensions. In the per-domain schedule, where each job runs 2.4 to 36.5 epochs of its domain, it gains 0.64 on imaging quality, on an interval that excludes zero.

Trained per domain, the full pool matches the curated pipeline. Against production deduplication, GridReS on the undeduplicated pool differs on no dimension beyond the seed floor, with imaging quality at -0.26 on an interval of [-0.67, +0.18]. Against ClusterClip it differs on no dimension beyond the seed floor either. In V-JEPA 2 features (Assran et al., 2025), a space the sampler never uses (Appendix E.2), the per-domain model improves on production curation in precision (Kynkäänniemi et al., 2019) and density (Naeem et al., 2020) by more than the seed gap, 0.016 and 0.028, and in recall and coverage by less (Appendix G.4).

Table 3: Trained models on the Kandinsky-6 SFT pool. Each row is GridReS ($d = 4$) minus the configuration it is compared with, as a paired VBench difference over 500 prompts. Imaging quality and subject consistency are on a 0–100 scale, the rest on 0–1. A star marks a 95% interval that excludes zero, and bold one that lies entirely beyond the seed floor. Production background, at a lower bound of 0.00001 (t-test $p = 0.053$), is left unstarred. Appendix G lists further configurations.

Compared with	imaging ↑	aesthetic ↑	subject ↑	background ↑	dynamic ↑
Single job: one model on the whole pool, 0.60 epochs					
random sampling	**+1.84***	+0.0060*	+0.38*	+0.0015	−0.024
in units of the seed floor	2.3	2.9	1.0	1.0	−1.2
Per domain: eleven jobs souped into one model, 2.4 to 36.5 epochs of a domain					
production curation	−0.26	−0.0029	+0.09	+0.0012	−0.022
random sampling	+0.64*	−0.0007	−0.36*	−0.0013*	+0.008
ClusterClip	+0.11	−0.0033*	−0.08	+0.0004	−0.006
GridReS at $d = 3$, $M = 32$	**+1.70***	+0.0023	+0.32*	+0.0006	−0.020
GridReS at $d = 8$	+0.97*	+0.0023	+0.05	+0.0003	+0.006
seed floor	0.80	0.0021	0.36	0.0015	0.020

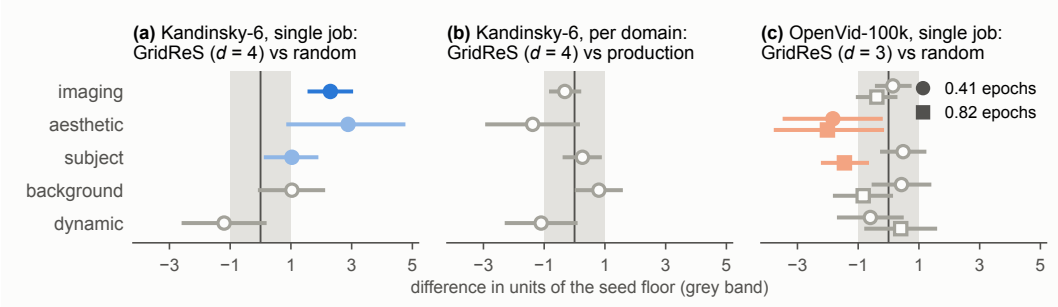


Figure 5: VBench differences in units of each dimension's seed floor, with 95% intervals over 500 prompts. **(a)** As a single job at 0.60 epochs, GridReS beats random sampling on imaging quality beyond the floor, and on aesthetic quality and subject consistency with intervals that exclude zero. **(b)** Per domain, at 2.4 to 36.5 epochs, it matches the production pipeline. **(c)** On OpenVid-100k it matches random sampling on four of five dimensions at 0.41 epochs, and trails on two at 0.82 with neither interval beyond the floor. As in Table 3, a dark marker has its whole interval beyond the floor (bold) and a light one an interval that excludes zero (star); blue favours GridReS, orange the baseline, grey a tie. Panel (c) uses the Kandinsky-6 floor, though OpenVid-100k seeds vary more.

The configuration of Section 2 matters. The configuration of Section 2 improves on ($d = 3$, $M = 32$) by 1.70 on imaging quality, on an interval of [1.33, 2.07], and on $d = 8$ by 0.97, which also trails it in V-JEPA 2 precision and density by more than the seed gap.

On OpenVid-100k GridReS matches random sampling's quality. At $\alpha = 1$, where cell weighting is inert, the OpenVid-100k runs of Section 3 tie random sampling at 5,000 steps on four of five dimensions (Figure 5c), and the aesthetic differences at 5,000 and 10,000 steps, 0.0038 and 0.0042, lie within the spread of four random-sampling seeds, which differ by up to 0.0052. Matching random sampling here shows that the rebalancing costs no measurable quality. OpenVid-1M is already filtered for aesthetics, motion and clarity, so random draws from it are a strong baseline, and GridReS matches them while changing which clips the model sees together: in simulation, the configuration of these runs draws near-duplicates together 17% less often at the same coverage (Appendix H.2).

5 Ablations

Each ablation varies one setting, is paired on everything else, and is measured on the induced draw distribution. Appendices G and H give the full sweeps.

The partition bounds the cost of repulsion. Scoring a candidate against its neighbourhood queue costs M distance evaluations instead of N, and a draw scores only the unserved members of one cell. Sharing the queue across the neighbourhood instead of keeping it per cell is worth 7.3% on the Kandinsky-6 SFT pool and 3.0% on OpenVid-100k, pooled over the other settings, as the boundary-straddling pairs of Appendix B.5 predict. The neighbourhood also carries part of the reduction. At matched coverage, one global queue of the same length over the whole domain, with no grid, taking the best of 64 unserved candidates per draw, reads 0.675 where GridReS reads 0.495, a paired ratio of 0.750 [0.660, 0.850] over 133 runs.

Projection dimension. At coverage matched to 0.283, $d = 4$ beats $d = 3$ at $\ell = 0.10$ ($p = 2.6 \times 10^{-6}$, paired Wilcoxon test (Wilcoxon, 1945)) and $d = 5$ ($p = 2.2 \times 10^{-3}$) on the Kandinsky-6 SFT pool (Table 6). Duplicate capture, the fraction of near-duplicate pairs inside one cell, does not predict this ordering: at the cell sides of Table 6 it rises from 32.9% at $d = 3$ to 71.5% at $d = 8$ while R worsens, and $d = 4$ wins at 44.9%. The one controlled model-level test agrees with the offline ordering, with $d = 8$ trailing $d = 4$ by 0.97 on imaging quality, on an interval of [0.63, 1.32], at matched occupancy (Table 3), and by 0.034 in V-JEPA 2 density (Appendix G.4). A high d also costs more, 2.54 ms per draw at $d = 8$ against 0.33 ms at $d = 3$, and past $d \approx 11$ bisection cannot reach the target occupancy.

6 Limitations and Negative Results

Cell weighting depends on the pool. On its own it lowers R on the Kandinsky-6 SFT pool, to 0.749 at $\alpha = 0.5$, and raises it on OpenVid-100k at every $\alpha < 1$, to 1.17 at 0.5 and 1.40 at 0, returning to 1.02 only at $\alpha = 1$, where the weighting is inert. Its best setting differs too: on the Kandinsky-6 SFT pool $\alpha = 0.5$ improves on $\alpha = 1$ by 25% but cannot be told from $\alpha = 0.25$, while OpenVid-100k prefers $\alpha = 1$ (Appendix H.1). Its premise holds only weakly: occupancy correlates $+0.11$ with k-nearest-neighbour density and $+0.09$ with cross-source duplicate degree (Appendix C.6). Nor is the PCA basis shown to be the better one: at matched occupancy a random projection does better on one of the two domains tested and worse on the other (Appendix A.7).

Some measurements favour the baselines. On OpenVid-100k SemDeDup keeps 75% of the pool, which budgets of at most half the pool leave at random sampling's coverage, and reaches a lower R than GridReS, 0.174 against 0.754. Trained per domain, GridReS trails random sampling on subject and background consistency by 0.36 and 0.0013 and ClusterClip on aesthetic quality by 0.0033, with intervals that exclude zero but do not clear the seed floor. On OpenVid-100k it trails random sampling on aesthetic quality by 0.0038 and 0.0042 at 5,000 and 10,000 steps, within the seed spread of 0.0052, and at 10,000 steps on subject consistency by 0.53, against a difference of 0.47 between two random-sampling seeds. As a single job, GridReS trails its random control in V-JEPA 2 precision, density and coverage by more than the seed gap, although it leads on VBench imaging quality (Appendix G.4). In the single job's own draw log GridReS reaches fewer distinct clips over its first 27,200 draws, 19.6% of the pool against 33.7%, because the eight rank streams concentrate on the same sparse cells (Appendix H.7).

R does not reach past one pass. Because it counts co-draws over all draws so far, R equals 1 at every whole number of passes and cannot show whether later passes space near-duplicates as the first does; six of the eleven per-domain jobs exceed one pass per stream. A variant over a window of recent draws would show it, and we have not computed one.

Remaining limitations. Each trained configuration is a single run apart from the retrains behind the seed floor and three further random-sampling seeds on OpenVid-100k, and every result is for one model family at one scale.

7 Conclusion

The partition in GridReS is first a locality structure and only weakly a density model. It makes repulsion affordable, and repulsion cuts near-duplicate co-exposure at unchanged coverage and, in a single job, improves imaging quality by 1.84 points over random sampling. The reduction a threshold and a per-region cap perform once before training can be performed at every draw, with nothing discarded.

AI USE STATEMENT

Generative AI was used in this work and we disclose its use by task. It was used for **writing**: polishing the manuscript and grammar, checking it for logical inconsistencies, checking that claims needing a citation carry one, checking that its structure and cross-references. It was **not** used for translation and synthetic data generation, and it was not used to generate, impute or modify any experimental result. Every table and figure is computed from the recorded per-run results. The authors take full responsibility for all content.

ETHICS STATEMENT

Data provenance. The primary pool, the Kandinsky-6 SFT pool, is a licensed video collection used for post-training in a production pipeline. We cannot release it or disclose its acquisition terms and the configuration of the pipeline stages upstream of the sampler. To keep the claims checkable without it, the draw-distribution results and a set of trained models are also reported on OpenVid-100k, whole shards of the public OpenVid-1M (Nan et al., 2025); the sampler is released at `https://github.com/94yvwj2hrj-eng/gridres`. Every table and figure names the pool its numbers come from.

Memorization. This work keeps near-duplicates in the pool and controls how often they are drawn together, rather than deleting them. Duplication is a known driver of verbatim reproduction (Carlini et al., 2023; Somepalli et al., 2023), so keeping them carries a privacy and copyright risk that deletion avoids. We have not measured memorization, and it should be evaluated alongside quality whenever a pool keeps its near-duplicates. Exact and perceptually identical copies are rejected upstream of the pool (Appendix B.1); we recommend keeping that step even when near-duplicates are kept, and a deployment of GridReS should keep measuring extraction.

REPRODUCIBILITY STATEMENT

The sampling rule is specified in Section 2, with Algorithm 1 giving one draw and Appendix C the scaling and the distributed semantics; every hyperparameter, including the ones we did not tune, is in Appendix I, the evaluation protocol is in Appendix F, and the implementation is available at `https://github.com/94yvwj2hrj-eng/gridres`. Because the primary pool cannot be released, three parts of the paper can be checked without it. The draw-distribution comparison of Table 2 and the α, target-occupancy and queue-length sweeps are also run on OpenVid-100k, which is public, and Appendix D.2 lists the shards that form it. Every distributional quantity is computed from the draw log the sampler emits (Appendix I.1), so it can be measured on any candidate pool without a trained model. The exhausted-fraction rule of Appendix H.1 is a closed form over the occupancy histogram that can be checked before any sampling is run.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023.

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory (ICDT)*, pp. 420–434, 2001.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International Conference on Database Theory (ICDT)*, pp. 217–235, 1999.

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 5622–5633, 2018.

Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, Ting-Che Lin, Shilong Zhang, Fu Li, Chuan Li, Xing Wang, Yanghua Peng, Peize Sun, Ping Luo, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Goku: Flow based video generative foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23516–23527, 2025.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *ACM Symposium on Computational Geometry (SoCG)*, pp. 253–262, 2004.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, Xunsong Li, Yifu Li, Shanchuan Lin, Zhijie Lin, Jiawei Liu, Shu Liu, Xiaonan Nie, Zhiwu Qing, Yuxi Ren, Li Sun, Zhi Tian, Rui Wang, Sen Wang, Guoqiang Wei, Guohong Wu, Jie Wu, Ruiqi Xia, Fei Xiao, Xuefeng Xiao, Jiangqiao Yan, Ceyuan Yang, Jianchao Yang, Runkai Yang, Tao Yang, Yihang Yang, Zilyu Ye, Xuejiao Zeng, Yan Zeng, Heng Zhang, Yang Zhao, Xiaozheng Zheng, Peihao Zhu, Jiaxin Zou, and Feilong Zuo. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Adhiraj Ghosh, Vishaal Udandarao, Thao Nguyen, Matteo Farina, Mehdi Cherti, Jenia Jitsev, Sewoong Oh, Elisa Ricci, Ludwig Schmidt, and Matthias Bethge. Concept-aware batch sampling improves language-image pretraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3056–3068, 2026.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing (STOC)*, pp. 604–613, 1998.

Kandinsky Lab. Kandinsky 5.0: A family of foundation models for image and video generation. *arXiv preprint arXiv:2511.14993*, 2025.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5148–5157, 2021.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 3519–3529, 2019.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

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

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 30233–30249, 2022.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 3927–3936, 2019.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Yize Li, Yihua Zhang, Sijia Liu, and Xue Lin. Pruning then reweighting: Towards data-efficient training of diffusion models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025.

Zhiyang Liang, Ziyu Wan, Hongyu Liu, Dong Chen, Qiu Shen, Hao Zhu, and Dongdong Chen. LACON: Training text-to-image model from uncurated data. In *European Conference on Computer Vision (ECCV)*, 2026.

Xiangyang Luo, Qingyu Li, Yuming Li, Guanbo Huang, Yongjie Zhu, Wenyu Qin, Meng Wang, Pengfei Wan, and Shao-Lun Huang. Beyond the golden data: Resolving the motion-vision quality dilemma via timestep selective training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43440–43449, 2026.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning (ICML)*, volume 119, pp. 7176–7185, 2020.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical AI. *arXiv preprint arXiv:2511.00062*, 2025a.

NVIDIA, Francesco Ferroni, Prithvijit Chattopadhyay, Greg Heinrich, Mike Ranzinger, Roberto Amoroso, Alice Luo, Andrew Wang, and Ming-Yu Liu. Cosmos-Embed1: A joint video-text embedder for physical AI. `https://huggingface.co/nvidia/Cosmos-Embed1-448p`, 2025b.

Valeria Pais, Daniele Faccio, Luis Oala, and Marco Aversa. Autoguided online data curation for diffusion model training. *arXiv preprint arXiv:2509.15267*, 2025.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

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 *International Conference on Machine Learning (ICML)*, volume 139, pp. 8748–8763, 2021.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10064–10083, 2024.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In

Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, volume 35, pp. 25278–25294, 2022.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

Yunfan Shao, Linyang Li, Zhaoye Fei, Hang Yan, Dahua Lin, and Xipeng Qiu. Balanced data sampling for language model training with clustering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14012–14023, 2024.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058, 2023.

Team Seaweed, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, Feng Cheng, Feilong Zuo, Xuejiao Zeng, Ziyan Yang, Fangyuan Kong, Meng Wei, Zhiwu Qing, Fei Xiao, Tuyen Hoang, Siyu Zhang, Peihao Zhu, Qi Zhao, Jiangqiao Yan, Liangke Gui, Sheng Bi, Jiashi Li, Yuxi Ren, Rui Wang, Huixia Li, Xuefeng Xiao, Shu Liu, Feng Ling, Heng Zhang, Houmin Wei, Huafeng Kuang, Jerry Duncan, Junda Zhang, Junru Zheng, Li Sun, Manlin Zhang, Renfei Sun, Xiaobin Zhuang, Xiaojie Li, Xin Xia, Xuyan Chi, Yanghua Peng, Yuping Wang, Yuxuan Wang, Zhongkai Zhao, Zhuo Chen, Zuquan Song, Zhenheng Yang, Jiashi Feng, Jianchao Yang, and Lu Jiang. Seaweed-7B: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pp. 402–419, 2020.

Vishaal Udandarao, Ameya Prabhu, Adhiraj Ghosh, Yash Sharma, Philip H. S. Torr, Adel Bibi, Samuel Albanie, and Matthias Bethge. No "zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 61735–61792, 2024.

Jarkko Venna and Samuel Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *Artificial Neural Networks – ICANN 2001*, volume 2130 of *Lecture Notes in Computer Science*, pp. 485–491. Springer, 2001.

Apoorv Vyas, Heng-Jui Chang, Cheng-Fu Yang, Po-Yao Huang, Luya Gao, Julius Richter, Sanyuan Chen, Matt Le, Piotr Dollár, Christoph Feichtenhofer, Ann Lee, and Wei-Ning Hsu. Pushing the frontier of audiovisual perception with large-scale multimodal correspondence learning. *arXiv preprint arXiv:2512.19687*, 2025.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning (ICML)*, volume 37, pp. 1954–1963, 2015.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, volume 162, pp. 23965–23998, 2022.

Cheng Zhang, Hedvig Kjellström, and Stephan Mandt. Determinantal point processes for mini-batch diversification. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Wanru Zhao, Yihong Chen, Yuzhi Tang, Wentao Ma, Shengchao Hu, Shell Xu Hu, Alex Iacob, Abhinav Mehrotra, and Nicholas D. Lane. Rethinking data curation in LLM training: Online reweighting offers better generalization than offline methods. In *International Conference on Learning Representations (ICLR)*, 2026.

A Related work

A.1 Curation in video generation pipelines

Recent video generation systems all reduce a candidate pool before training, and the reduction is reported but not evaluated. Seedance defines several hundred categories by visual style, motion type and other attributes and collects fine-tuning data within each (Gao et al., 2025). Seaweed-7B curates 700k fine-tuning clips by human annotation with a balanced distribution (Team Seaweed et al., 2025). Wan deduplicates its candidate pool and then selects a set amount of data from each of 100 clusters, so that small regions of a long-tailed distribution survive (Team Wan et al., 2025). Goku removes near-duplicate clips by perceptual hashing of keyframes, keeping the copy with the higher aesthetic score, and rebalances 86 subcategories by down- and oversampling (Chen et al., 2025), and Kandinsky 5.0 removes visually similar scenes by perceptual hashing and samples across 10,000 embedding clusters (Kandinsky Lab, 2025). Cosmos trains a domain classifier, fine-tunes one model per domain and merges them, which removes the need to choose a mixture weight (NVIDIA et al., 2025a). Public corpora ship with their curation already applied, so a user of OpenVid inherits selection thresholds they can read about but cannot revisit, since the discarded clips are not released (Nan et al., 2025). In each case balance is a stage justified by construction. We found no work in this set that measures it against the alternative of omitting it.

Offline deduplication. The methods these pipelines use share a form. SemDeDup (Abbas et al., 2023) clusters an embedding space with k-means and, within each cluster, keeps one item from every group above a cosine threshold, and we run it at matched budget as a curator baseline (Table 2). Such a method cannot revisit the decision once training has begun, which is the property GridReS changes.

A.2 What curation costs

Two lines of evidence bear on whether the discarded material has value, and they disagree. For filtering: Stable Video Diffusion trains on a filtered subset and on an unfiltered set of the same origin and finds the filtered model preferred, which is the one controlled comparison of this kind we located (Blattmann et al., 2023). Against it: LACON trains a text-to-image model on uncurated data by conditioning on its quality signals instead of filtering, and reports better generation quality than baselines trained only on filtered data at equal compute (Liang et al., 2026), and Beyond the Golden Data reports the same direction for quality-imbalanced video data under timestep-selective training (Luo et al., 2026). Two further results constrain the threshold from both sides. Per-concept performance, including the accuracy of generated images, rises roughly log-linearly with the concept's frequency in the training data (Udandarao et al., 2024), which argues against thinning a region, while duplication drives verbatim reproduction (Carlini et al., 2023), which argues for it. Our position is narrower than either side of this argument. We do not claim the discarded clips are valuable. We claim the decision to discard them is taken once, before any training signal exists, and that a sampler can defer it.

A.3 Subset selection and coresets

Facility location (Wei et al., 2015) and k-centre greedy (Sener & Savarese, 2018) optimise coverage of an embedding space explicitly, and they do so more directly than a similarity threshold does. We run a greedy facility-location coreset as a baseline for that reason, and it reaches the lowest R on the Kandinsky-6 SFT pool, over its 15 surviving runs. What this family shares is timing, not objective. The output is a fixed subset, chosen before the first optimizer step and not revisited.

Pruning then Reweighting is the closest to our position, since it follows a pruning stage with class-level reweighting derived from a reference diffusion model (Li et al., 2025), but the pruning decision is still made once and the reweighting acts across classes rather than within them.

A.4 Cluster-based and online sampling

ClusterClip samples uniformly over k-means clusters and caps how often a sample can be drawn, because uniform sampling over clusters repeats the documents of small clusters until the model overfits them (Shao et al., 2024), which is evidence from inside the prior work that repetition, not only coverage, limits a balanced sampler. Among online methods, CABS constructs each batch against a target distribution, but defines that target over a concept vocabulary annotated across 128M pairs (Ghosh et al., 2026). ADAPT casts selection, mixing and balancing as data reweighting and reweights each sample's loss online by its embedding similarity to a validation set (Zhao et al., 2026). Autoguided curation scores candidates online with a second model (Pais et al., 2025), which is the closest setting to ours. The cluster- and concept-level methods act on regions, and near-duplicates fall inside one region by construction, so reweighting regions cannot separate them. The per-sample methods require a signal the data does not carry, whether a validation set or a reference model. GridReS reweights the draw rather than the loss, and uses only the geometry of a pretrained embedding.

A.5 Repulsive and determinantal sampling

Diversifying mini-batches with a determinantal point process (DPP) was proposed by Zhang et al. (2017), who draw batches with probability decreasing in their internal redundancy. That is the same operation as our second mechanism and the closest prior art to it. The difference is cost and scope: a DPP over a batch needs a kernel over the candidate set, while we score one candidate against a bounded queue, and the DPP acts inside a batch drawn at random, not inside a region of the embedding space. One result argues against us. CommonIT reports that homogeneous mini-batches drawn from a single data group, by task, embedding cluster or length, help instruction tuning (Rao et al., 2024), which is the opposite of batch-level repulsion, so we leave cell ordering as an open ablation (Appendix G.2) rather than assume our side of the question.

A.6 Position of this work

The contribution is that the two operations a production pipeline performs once, near-duplicate removal and per-region capping, can instead be performed at every draw, at bounded cost, with nothing removed from the pool. The nearest prior operation is mini-batch DPP diversification, which acts on redundancy inside a batch at the cost of a kernel over the candidate set. We are not aware of prior work that acts on redundancy inside a partition of the embedding space during training, which is where the two operations this paper replaces do their work.

A.7 Locality-sensitive hashing and the price of a data-dependent basis

The partition of Appendix B.5 is a projection followed by quantisation into fixed-width buckets, which follows the p-stable locality-sensitive hashing (LSH) construction (Datar et al., 2004) built on the framework of Indyk & Motwani (1998). Two differences separate it from that guarantee. LSH projects on random directions with a random offset and inherits a collision guarantee from the stability of the projection distribution, while a PCA basis carries no such guarantee and depends on the data it is fitted on. On this pool the dependence is small: refitting the projection of Section 2.1 with the same seed on two disjoint halves of $\mathcal{P}$ moves its subspace by at most 5.1° (Appendix E.4). LSH also handles items falling on the wrong side of a bucket boundary by querying several independent tables, where GridReS probes the 3^d neighbouring buckets of a single table. At $d = 4$ that covers $\beta = 0.992$ of pairs above τ against 0.319 for one cell, which is enough for the mechanism but is not a bound. A random-projection variant recovers the guarantee and gives up the principal axes, and at matched occupancy it does not settle the question. On *tech* the PCA basis reads 0.552 against the random projection's 0.575, and on *nature* the random projection reads 0.490 against the PCA basis's 0.515, so the sign reverses between two domains of one pool over 15 to 17 paired runs each. We therefore do not claim the PCA basis is the better choice.

B NOTATION AND PRELIMINARIES

Table 4 collects the symbols used in the paper. The rest of this appendix sets up the measure of redundancy and works out what a sampler can and cannot do to it: a sampler that only chooses among regions trades redundancy against coverage, occupancy weighting helps only where occupancy tracks duplicate density, which holds weakly on the Kandinsky-6 SFT pool and not on OpenVid-100k, and the partition is built in a low dimension because locality is cheap there and not in the full space.

Table 4: Notation used in the paper.

Symbol	Meaning
$\mathcal{P}$, N	candidate pool and its size; for a per-domain job, the domain's clips
$\phi(x)$	unit-norm embedding of clip x in $\mathbb{R}^{4096}$
$v(i)$	source video of clip i
τ	near-duplicate threshold on cosine similarity, 0.90
E	cross-source near-duplicate pairs (i, j) with $i < j$
p, u	per-draw marginal over $\mathcal{P}$, and the uniform marginal
$R(p)$	probability that two independent draws form a pair in E, equation 1
R_{unif}, ρ	R under uniform sampling, $2\lvert E\rvert/N^2$, and the ratio R/R_{unif}
R/R_{rand}	paired ratio to a random permutation at the same budget, the reported quantity
π, d	PCA projection and its dimension
ℓ, $c(x)$	cell side, and the cell of clip x
n_c	occupancy of cell c, counted over the job's rows
α, Z	cell-weighting exponent, and the normaliser $\sum_c n_c^\alpha$
Q_c, M	queue read by cell c, and its length
$s(x)$, n_x	score of candidate x, and the number of times x has been drawn
B	budget, the number of draws
q_c, E_c, m_c	cell probability of a region-level sampler, the pairs of E inside c, and their number
ε_i, κ	relative deviation of p_i from uniform, and how the weights at the two ends of a pair move together, equation 8
β	fraction of pairs above τ whose two cells are neighbours
r_s	Spearman rank correlation

B.1 SETUP

Write $\mathcal{P}$ for the candidate pool, with $N = |\mathcal{P}| = 67{,}175$ clips. It holds the clips from which supervised fine-tuning of Kandinsky-6 Lite, a text-to-audio-video (T2AV) model, is drawn before deduplication; it is not public, and every clip carries one of 11 domain labels (Appendix D.1). A frozen multimodal encoder ϕ, Qwen3-VL-Embedding-8B (Li et al., 2026), maps each clip x to a vector $\phi(x) \in \mathbb{R}^{4096}$ over a 5.04-second, 121-frame window. The model emits L2-normalised vectors, verified over the pool at norms between 0.9998 and 1.0002, so $\phi(x)$ lies on the unit sphere. Embeddings are computed once offline and cached, and no encoder runs during training.

One property of that space matters later. Mean pairwise cosine similarity between unrelated clips is $+0.151$ rather than 0, so the embeddings occupy a cone and the usable cosine-distance range is compressed to roughly $[0.22, 1.06]$ instead of $[0, 2]$. The near-duplicate threshold of Appendix B.2 sits far above that background, and the compression is one more reason to score a candidate by its nearest queue entry, since an average over a compressed range dilutes one close neighbour further.

The public pool, OpenVid-100k (Appendix D.2), goes through the same encoder and graph construction, and the two pools are named wherever a number is reported for both. The data curation pipeline for Kandinsky-6 reduces $\mathcal{P}$ to 8,065 clips, a ratio of 8.3:1 (Appendix D.1), and two operations account for all of it: removal of near-duplicates above a similarity threshold, and capping of over-represented regions. These two are what GridReS replaces. Rejection of exact and perceptually identical copies happens upstream and removes no clip from $\mathcal{P}$.

B.2 A NEAR-DUPLICATE GRAPH AND A CO-EXPOSURE MEASURE

Semantic deduplication declares two items duplicates above a similarity threshold in embedding space (Abbas et al., 2023); we make that relation explicit instead of applying it. Let $G = (\mathcal{P}, E)$

with $E = \{(i, j) : i < j,\ v(i) \neq v(j),\ \langle \phi(x_i), \phi(x_j) \rangle > \tau\}$, where $v(i)$ is the source video of clip i and $\tau = 0.90$, computed exactly over all pairs, giving $|E| = 12{,}737$ edges, or 0.190 per clip. Of the 64,312 pairs above τ, 80.2% are two scenes of one source video; they are similar by construction and are not the redundancy a sampler should suppress, so they are excluded. A sampler induces a per-draw marginal p over $\mathcal{P}$, and $R(p)$ of equation 1 is the probability that two independent draws form a near-duplicate pair. Under uniform sampling $p_i = 1/N$ and $R_{\text{unif}} = 2|E|/N^2$, so all results are reported as the ratio $\rho(p) = R(p)/R_{\text{unif}}$.

B.3 A REGION-LEVEL SAMPLER TRADES REDUNDANCY AGAINST COVERAGE

Let the cells c partition $\mathcal{P}$ with $n_c = |c|$, write $c(i)$ for the cell of clip i, and let a sampler pick a cell with probability q_c and then draw uniformly inside it, so $p_i = q_{c(i)}/n_{c(i)}$. With E_c the edges of E internal to c and $m_c = |E_c|$, the intra-cell part of equation 1 is

$$R_{\text{in}} \;=\; 2 \sum_{c} \frac{m_c\, q_c^2}{n_c^2} \tag{3}$$

Distinct clips taken from c over B draws number $n_c\big(1-(1-q_c/n_c)^B\big)$ in expectation, or $q_c B$ to first order. The two differ in their order in q_c, so to first order scaling a cell's contribution to equation 3 by a factor λ scales that cell's coverage by $\sqrt{\lambda}$. Every sampler that only chooses among regions is bound by this trade, and a deleting curator is its limiting case at $q_c = 0$.

A policy acting inside a cell is not bound by it. At fixed q_c the intra-cell term is $2q_c^2 \sum_{(i,j)\in E_c} p_{i|c} p_{j|c}$ for a conditional distribution $p_{\cdot|c}$ over c, and shaping $p_{\cdot|c}$ reduces it at unchanged coverage. This is the argument for the second mechanism.

B.4 THE OCCUPANCY EXPONENT, AND WHEN IT COULD WORK

GridReS draws a cell with probability proportional to n_c^α. Composing that with a uniform within-cell draw gives a closed form for the marginal,

$$p_i \;=\; \frac{n_{c(i)}^{\alpha-1}}{Z}, \qquad Z \;=\; \sum_{c} n_c^\alpha \tag{4}$$

At $\alpha = 1$ the marginal is exactly $1/N$, so the weighting is the identity map and $\rho \equiv 1$ for every partition and cell side. At $\alpha < 1$ mass moves toward small cells, and substituting equation 4 into equation 1 gives

$$\rho \;=\; \frac{N^2}{|E|\, Z^2} \sum_{(i,j)\in E} \big(n_{c(i)}\, n_{c(j)}\big)^{\alpha-1} \tag{5}$$

The sum averages $(n_{c(i)} n_{c(j)})^{\alpha-1}$ over edges against the same average over all pairs, so for $\alpha < 1$ it falls below 1 only if near-duplicate edges are over-represented in high-occupancy cells. Occupancy must carry information about local duplicate density for the exponent to do anything.

Whether it does depends on the geometry and the pool. At the $d = 4$ of Section 2.1, occupancy correlates $+0.111$ with k-nearest-neighbour density in the full embedding space and $+0.087$ with cross-source duplicate degree (Appendix C.6), so edges sit slightly more often in high-occupancy cells. On the Kandinsky-6 SFT pool the grid alone then reads 0.638 at $\alpha = 0$ and rises with α (Table 13), the ordering Equation 5 predicts; its 1.086 at $\alpha = 1$, where $\rho = 1$ exactly, comes from drawing cells independently (Appendix H.2). On OpenVid-100k the grid alone lies above random at every $\alpha < 1$ (Table 14). What decides the outcome is the condition, a property of the encoder, the projection and the pool, and not occupancy weighting as a family.

A perturbation argument says the same thing about any region-level reweighting. Write A for the symmetric adjacency matrix of E and $\deg(i)$ for the number of edges at clip i, so that $R(p) = p^{\mathsf{T}} A p$, and let $p = u + \delta$ with u uniform and $\sum_i \delta_i = 0$. Then

$$R(u+\delta) \;=\; R_{\text{unif}} \;+\; \frac{2}{N}\sum_i \deg(i)\,\delta_i \;+\; \delta^{\mathsf{T}} A\,\delta \tag{6}$$

so a reweighting orthogonal to duplicate degree leaves R unchanged to first order, and any reduction has to come from the second-order term. A is not positive semi-definite, so that term can be negative, but making it so requires the reweighting to align with the structure of E and not merely to vary with occupancy. Equation 5 says which alignment: what enters the condition is a cell's share of E rather than its share of $\mathcal{P}$, and that share is computable offline from the graph already built. We do not test that variant here.

B.5 Why the partition is built in a low-dimensional projection

A grid of side ℓ over a projection of extent Λ into d dimensions has $(\Lambda/\ell)^d$ cells, so holding mean occupancy at a target t requires $\ell \approx \Lambda(t/N)^{1/d}$. The side grows toward Λ with d until the partition separates nothing, and past $d \approx 11$ the bisection of Section 2.1 cannot reach the target at all. Dimension defeats the distances as well: the farthest-to-nearest neighbour ratio of a query tends to 1 under broad conditions (Beyer et al., 1999), the faster under L_k norms with larger k (Aggarwal et al., 2001), so in $\mathbb{R}^{4096}$ an equal-volume cell is not informative. The partition is therefore built in a projection to $d \le 5$, while every repulsion score is computed in the full space, where E is defined.

Projecting and then quantising into fixed-width buckets follows the construction of locality-sensitive hashing (Indyk & Motwani, 1998; Datar et al., 2004), and it serves the same purpose: a nearest-neighbour question becomes a bucket lookup. Appendix A.7 sets out where our construction departs from that guarantee. Writing β for the fraction of pairs (i, j) above τ with $c(j)$ in the 3^d neighbourhood of $c(i)$, a memory of the M most recent draws from that neighbourhood can hold a candidate's partner at a cost of M distance evaluations per candidate rather than N. At the geometry of Section 2.1, $\beta = 0.992$ over all pairs above τ and 0.982 over the cross-source pairs of E, while only 0.319 and 0.228 share a single cell (Table 5), which is why the memory is shared across the neighbourhood.

B.6 Relation to determinantal point processes

The behaviour the second mechanism needs is already characterised by a determinantal point process (DPP) (Kulesza & Taskar, 2012): given a positive semi-definite kernel L over $\mathcal{P}$, a DPP draws S with $P(S) \propto \det L_S$. For a pair,

$$\det L_{\{i,j\}} \;=\; L_{ii}L_{jj} - L_{ij}^2 \tag{7}$$

so the probability that the drawn set is exactly $\{i, j\}$ falls with the square of their similarity, and in the marginal kernel $K = L(L + I)^{-1}$ of the same process the probability of drawing both, $K_{ii}K_{jj} - K_{ij}^2$, behaves the same way. This is exactly the suppression of near-duplicate co-exposure that equation 1 measures and the second-order term of equation 6 permits. This is the origin of the mechanism, and mini-batch diversification by a DPP (Zhang et al., 2017) is its closest existing form. Exact sampling requires an eigendecomposition of L over the ground set (Kulesza & Taskar, 2012), and that ground set changes as clips retire during an epoch, so the kernel would be rebuilt during training. GridReS keeps the pairwise suppression of equation 7 and replaces the global kernel with a bounded per-neighbourhood memory, inheriting none of the DPP's guarantees.

Nor is GridReS an approximation to a DPP, for three reasons. First, the score is not determinantal: greedy DPP MAP adds the item maximising $\det(L_{S\cup\{i\}})/\det(L_S)$, which for L the Gram matrix of ϕ is the squared distance from ϕ_i to the *span* of the selected set S, whereas equation 2 is a minimum of pairwise distances, and a candidate can be far from every queue member individually and still lie close to their span. Second, masking the kernel outside a neighbourhood is not structure-preserving: zeroing off-diagonal entries of a positive semi-definite kernel need not preserve positive semi-definiteness, and without that $\det(L_S)$ is not a probability. Third, a recency queue is not conditioning: DPP conditioning has an exact Schur-complement kernel, whereas a first-in-first-out memory is a different process, and we draw a stream rather than a fixed-size subset. Windowed greedy log-determinant selection (Chen et al., 2018) is the closest design precedent for a bounded

repulsion memory, and what remains of the correspondence is that inspiration and a cost comparison (Chen et al., 2018).

C THE SAMPLER IN DETAIL

Algorithm 1 gives one draw as it runs on a rank. The subsections after it cover the choices that Section 2 states without argument: how the projection is scaled, what to report about the geometry, how the state is distributed across ranks, why reweighting cells carries a cost, and how closely occupancy tracks density.

C.1 ONE DRAW

Algorithm 1 One draw of GridReS on one rank. U_c is the set of members of cell c not yet served in the current pass.

pick a cell c with $U_c \neq \emptyset$, with probability proportional to n_c^α
for each $x \in U_c$ **do**
    $s(x) \leftarrow \min_{y \in Q_c} \big(1 - \langle \phi(x), \phi(y) \rangle\big)$, or 1 if Q_c is empty
    $s(x) \leftarrow s(x) + 0.1/(1 + n_x)$
end for
draw $x^\star \in U_c$ with probability $s(x^\star) / \sum_{x \in U_c} s(x)$
$U_c \leftarrow U_c \setminus \{x^\star\}$ and $n_{x^\star} \leftarrow n_{x^\star} + 1$
for each occupied cell c' in the 3^d neighbourhood of c **do**
    append $x^\star$ to $Q_{c'}$, and drop the oldest entry if $|Q_{c'}| > M$
end for
if $U_{c'} = \emptyset$ for every cell c' **then**
    refill every $U_{c'}$ with all members of c'
end if
return $x^\star$

Two details of the algorithm matter for the measurements. The choice inside a cell is random, with probability proportional to the score, so a candidate close to a recent draw is postponed rather than excluded, and the bonus $0.1/(1 + fn_x)$ keeps a candidate at zero distance drawable. Retiring a cell once its members are served, and refilling the cells only when all are spent, makes each rank's stream a permutation of the pool at every pass. Scoring one candidate costs M inner products in the full space and a draw scores the unserved members of one cell, so a draw costs at most Mn_c distance evaluations however large the pool is.

C.2 SCALING

The projected coordinates are centred at their median and divided by a *single* scalar, the 99.9th percentile of $|z|$ over all axes, placing 99.9% of coordinates in $[-1, 1]$. Because the divisor is one scalar this is a similarity transform: it rescales every distance by the same factor and leaves the partition's geometry intact. PCA components are orthonormal, so the projection is orthogonal: it never lengthens a distance, it treats every direction of the subspace alike, and a uniform grid in unrescaled PCA units is already uniform in the embedding's own Euclidean metric restricted to the subspace; a per-axis rescaling would destroy that. Measured anisotropy of the four projected coordinates is only $1.61\times$, with standard deviations from 0.289 to 0.465 in the scaled units (measured on the coordinates; the variance shares of Appendix E.4 are the range finder's estimates), so a per-axis correction buys little and costs that uniformity. What the single scalar does buy is independence from the encoder's output magnitude: ℓ becomes a fraction of the pool's radius, although it must still be set per pool (Section 2.1).

C.3 REPORTING THE GEOMETRY

Reporting (d, ℓ) is not enough: what the sampler sees is the occupancy distribution $\{n_c\}$, which d, ℓ and the pool size N fix jointly, so comparing dimensions at a fixed cell side really compares

occupancies. The configuration of Section 2 therefore tunes ℓ to a target mean occupancy, and where dimensions are compared we report occupied cells alongside d and ℓ. Table 5 gives the locality of near-duplicate pairs at $d = 4$ for four target occupancies, among them the $\ell = 0.1429$ of Section 2.1, and Table 6 the comparison of projection dimensions behind Section 5.

Table 5: Where near-duplicate pairs of the Kandinsky-6 SFT pool fall at $d = 4$, for the 64,312 pairs above τ, same-source pairs included, and for the 12,737 cross-source pairs of E. The cell side is bisected to each target mean occupancy over the pool, and the second row is the geometry of Section 2.1. A pair straddles a boundary when it is not inside one cell. The simulations renormalise the stored embeddings, whose norms lie within 2.5×10^{-4} of 1; this moves one clip across a boundary, giving 6,720 occupied cells against the 6,719 of the stored rows. Occupancy figures are pool-level; the sampler runs on per-domain restrictions of the same grid, where mean occupancy at $\ell = 0.1429$ ranges from 1.44 on *interiors* to 6.43 on *food*.

			all pairs above τ		cross-source pairs	
ℓ	Occupied cells	Mean occ.	Same cell	3^4 nbhd.	Same cell	3^4 nbhd.
0.1114	13,432	5.0	23.6%	96.7%	19.4%	96.0%
0.1429	6,720	10.0	31.9%	99.2%	22.8%	98.2%
0.1811	3,362	20.0	40.0%	99.9%	28.7%	99.8%
0.2190	1,920	35.0	47.8%	$> 99.9\%$	44.4%	$> 99.9\%$

Table 6: Kandinsky-6 SFT pool, 6 domains $\times$ 10 seeds $\times$ 3 sub-epoch budgets, paired, with coverage matched to 0.283 in every row so the comparison is not confounded by how much of the pool each configuration touches. Zero-R runs are dropped, 3 to 6 of 180 per row. Cells is the mean number of occupied cells per domain; capture is the fraction of near-duplicate pairs inside one cell over the whole pool, from a geometry sweep that has no entry at two of the cell sides. The cell sides are those of the dimension sweep, coarser than the $\ell = 0.1429$ of Section 2.1, so the $d = 4$ row is not the configuration of Table 2. The configuration with the highest capture has the worst outcome.

d	ℓ	cells	$R/R_{\text{rand}} \downarrow$ (95% CI)		capture
4	0.2197	694	**0.664**	[0.607, 0.728]	44.9%
5	0.2197	1064	0.725	[0.658, 0.796]	43.9%
3	0.0919	901	0.773	[0.699, 0.854]	32.9%
3	0.1000	756	0.777	[0.701, 0.863]	—
8	0.5000	635	0.827	[0.746, 0.916]	—
8	0.5251	548	0.905	[0.807, 1.013]	**71.5%**

C.4 Distributed state

Queues and counters are mutable state read by the sampling policy, so data-parallel training must decide who owns them. GridReS gives each rank its own queue over the whole partition: every rank may draw from any cell, no communication is required, repulsion is exact within a rank, and the global batch is the union of separate repulsive streams, one per rank. Sharding cells across ranks so that each queue lives in one place is an approximation rather than an improvement, because neighbourhoods overlap and no assignment of cells to ranks keeps every neighbourhood local; the links it drops are exactly the boundary near-duplicates the shared queue exists to catch. Queues and counters are part of the training state and are checkpointed with the model. A resume that restored the model but reinitialised them would empty every queue and restart the pass, and nothing in the loss curve would show it, so the checkpoint refuses to load state that is missing a queue for any cell.

Because every rank samples the whole pool with its own state, a rank's stream completes a pass after N of its own draws. A job of T steps at one clip per rank per step therefore makes T/N passes per stream while drawing $8T/N$ epochs of the pool, and it is the per-stream count that decides when R equals 1 by construction (Section 4.2). The random controls confirm that the streams are independent: over the first 27,200 draws of the single-job run, random sampling covers 0.337 of the pool, against 0.340 for eight independent permutations of 3,400 draws each and 0.405 for one shared permutation.

C.5 THE COST OF REWEIGHTING

The reason $\alpha < 1$ has a cost at all is that R is a quadratic form in the draw weights. Writing $p_i = (1 + \varepsilon_i)/N$ with $\sum_i \varepsilon_i = 0$, the ratio to uniform sampling is exactly

$$\frac{R(p)}{R_{\text{unif}}} \;=\; 1 \;+\; \frac{1}{|E|} \sum_{(i,j)\in E} (\varepsilon_i + \varepsilon_j) \;+\; \kappa\, \mathrm{CV}^2(p), \tag{8}$$

with $\mathrm{CV}^2(p) = N^{-1} \sum_i \varepsilon_i^2$ the squared coefficient of variation of the weights and $\kappa = |E|^{-1} \sum_{(i,j)\in E} \varepsilon_i \varepsilon_j / \mathrm{CV}^2(p)$ the extent to which the weights at the two ends of an edge move together. The middle term is the first-order term of equation 6. If cell weights are assigned independently of E, it vanishes in expectation, and the last term is non-negative in expectation, since the two ends of an edge share a weight when they share a cell and have independent weights otherwise, so such a reweighting raises R in expectation. Since $\mathrm{CV}^2(p) \to 0$ as $\alpha \to 1$ this cost vanishes at $\alpha = 1$, but it is only one of α's two effects. The other, exhaustion, has the opposite sign and is what Appendix H.1 measures.

C.6 CELL OCCUPANCY IS A WEAK PROXY FOR DENSITY

Weighting a cell by n_c^α with $\alpha < 1$ is usually motivated by occupancy standing in for local density. On the Kandinsky-6 SFT pool it does so weakly. At the $d = 4$ of Section 2.1, with $\ell = 0.1429$, cell occupancy correlates $+0.111$ with k-nearest-neighbour density ($k = 10$) in the full embedding space, $+0.087$ with each clip's degree in E and $+0.128$ with its degree over all pairs above τ. Occupancy therefore explains 0.8 to 1.6% of the variance of each quantity, so most of what the weighting injects is uncorrelated with the structure it is meant to balance. Section 6 and Appendix H.2 give what it does on each pool.

D DATA

D.1 THE KANDINSKY-6 SFT POOL

The pool holds 67,175 clips cut from 20,496 source videos, each labelled with one of eleven domains (Table 7). Domains differ in size by a factor of 15, from 16,764 clips of *people action* to 1,097 of *interiors*, and in redundancy by far more: counting same-source pairs, *animals* has 3.32 near-duplicate pairs per clip and *interiors* 0.03. The production pipeline keeps 8,065 clips, 12.0% of the pool, and tends to keep a larger share of the small domains, from 8.6% of *people speech* to 33.6% of *interiors*. The per-domain schedule inherits the size differences directly: at 40,000 draws a job makes 2.4 epochs of *people action* and 36.5 of *interiors*, or 0.30 and 4.56 passes per stream. All 59,110 clips the pipeline removes, 88.0% of the pool, go to redundancy removal and per-domain capping, the two operations GridReS replaces.

D.2 OPENVID-100K

OpenVid-1M is a public text-to-video corpus selected for aesthetics, temporal consistency, motion and clarity (Nan et al., 2025), which is the kind of filtering an SFT candidate pool has already passed through, and a 98,004-clip subset of it is of the same order of size as the Kandinsky-6 SFT pool. It carries no audio, so every OpenVid-100k run trains with the audio loss off (Table 16). The subset is taken in whole shards rather than as a random sample of clips, because near-duplicates cluster within a shard and a pair survives random subsampling only if both of its clips do: a random 10% of clips keeps about 1% of the pairs. Three shards are dense in within-shard near-duplicates, parts 15, 18 and 23, with 6.9 to 9.7 pairs per clip and 38,532 clips between them, and fifteen are sparse, parts 31, 39, 52, 65, 107, 112, 115, 116, 120, 122, 123, 124, 131, 137 and 142, with at most 0.02 pairs per clip and 59,472 clips. Clips whose frame size is not a multiple of 32 pixels, 38,578 of them, were re-encoded to meet the alignment the training loader requires, so none is lost.

Table 7: Composition of the Kandinsky-6 SFT pool. *Kept* is the production subset. Epochs and passes are those of one 5,000-step per-domain job at one clip per GPU on 8 GPUs. Pairs per clip count near-duplicate pairs above $\tau = 0.90$ inside the domain, with same-source pairs and on the cross-source graph; the cross-source count was recorded for eight domains. In the total row the first density sums the within-domain pairs, 60,694, whereas the pool-wide count of 64,312 gives 0.957 (Appendix D.3); the cross-source total is pool-wide.

					pairs per clip	
Domain	clips	kept	epochs	passes	all	cross-source
people action	16,764	1,611	2.4	0.30	0.265	0.050
people speech	12,581	1,076	3.2	0.40	0.998	0.044
animals	9,803	862	4.1	0.51	3.321	0.740
food	6,965	683	5.7	0.72	0.072	0.030
nature	5,779	759	6.9	0.87	0.246	0.065
cartoons	3,988	1,027	10.0	1.25	0.080	0.052
architecture	3,904	503	10.2	1.28	0.081	—
music	2,958	485	13.5	1.69	1.830	0.655
tech	2,116	379	18.9	2.36	0.700	0.280
art	1,220	311	32.8	4.10	1.350	—
interiors	1,097	369	36.5	4.56	0.033	—
total	67,175	8,065			0.904	0.190

D.3 The near-duplicate graph

Pairs above $\tau = 0.90$ number 64,312 on the Kandinsky-6 SFT pool, 0.957 per clip. Four in five of them, 80.2%, join two scenes cut from the same source video. Such pairs are similar by construction and say nothing about redundancy in the collection, so E keeps only the 12,737 pairs across source videos, 0.190 per clip. The two graphs agree on where most redundancy sits (Table 7): *animals* holds 53.6% of the 60,694 within-domain pairs and 57.0% of all 12,737 cross-source pairs, and *animals* and *music* lead on both counts. *people speech* moves most, from third to seventh of the eight domains recorded on both, because nearly all of its pairs are same-source. Table 5 reports locality on both graphs, the density axis of Figure 4b uses all pairs above τ, Table 15 gives the density of each domain on both graphs, and every value of R uses E.

E The embedding space

Every quantity in the paper is defined in one embedding space: the grid is cut in its projection, the near-duplicate graph E is thresholded in it, and the repulsion score is computed in it. This appendix records how the space was chosen, what the choice implies, and what the projection behind the grid keeps (Appendix E.4). The comparison covers six video encoders on the clips of the Kandinsky-6 SFT pool that all six have embedded, 51,176 of the 67,175, with every vector L2-normalised. Neighbourhood and pairwise measures use one shared subsample of 20,000 of these clips (seed 123), so every encoder and every pair is measured on the same rows.

E.1 Six candidate spaces

Table 8 describes each space on its own, in three groups. The first measures spread. The Vendi score (Friedman & Dieng, 2023) of Qwen3-VL-Embedding-8B is 183, more than twice that of the next encoder, and the space needs 833 principal components to reach 95% of its variance, against 442 for the 2B model and 114 to 216 for the rest.

The second group measures how close clips sit. In the PE and V-JEPA 2 spaces two random clips already have cosine 0.64 to 0.69, and 38 to 79% of clips have a nearest neighbour above 0.90, so the near-duplicate threshold of Appendix B.2 would join most of the pool and a uniform grid would pack it into few cells. The Qwen3-VL and Cosmos spaces keep random pairs at 0.16 to 0.25, and the 8B model has the fewest clips above the threshold, 18%, and the largest mean distance to the ten nearest neighbours. Hubness (Radovanović et al., 2010), the skew of how often a clip appears

among the ten nearest neighbours of others, is lowest for Cosmos-Embed1 and the 8B model. A hub would collect near-duplicate edges and repulsion from clips it does not resemble.

The third group measures how well each space separates the eleven domain labels. The two Qwen3-VL models are level here, the 2B model marginally ahead on the linear probe, the silhouette (Rousseeuw, 1987) and the separation AUC, while V-JEPA 2, the only candidate trained without text, separates domains worst. We chose the 8B model for its spread and its low near-duplicate rate. Its domain structure is no better than that of the 2B model, and it doubles the dimension in which the repulsion score is computed, where a draw costs 0.42 to 0.48 ms (Appendix I.3).

Table 8: The six candidate encoders on the common set of 51,176 clips, with neighbourhood measures on a shared subsample of 20,000: Qwen3-VL-Embedding-8B and 2B (Li et al., 2026), Cosmos-Embed1 (NVIDIA et al., 2025b), PE-AV Large with its audio input on and off (PE-AV and PE-V) (Vyas et al., 2025), and V-JEPA 2 ViT-L (Assran et al., 2025). *Spread*: the Vendi score, the participation ratio of the covariance eigenvalues λ, $(\sum \lambda)^2 / \sum \lambda^2$, and the number of principal components for 95% of the variance, on 10,000 clips. *Closeness*: the mean cosine of 8,000 random pairs, the mean cosine distance to the ten nearest neighbours, the near-duplicate rate (nearest neighbour above cosine 0.90) and the skew of the ten-nearest-neighbour counts. *Domain structure*: the share of the ten nearest neighbours in the same domain, the separation AUC (the probability that a same-domain pair is more similar than a different-domain pair), the cosine silhouette of the domain labels, and a linear probe, macro-F1 on a held-out 30%. Bold marks the best value in each row. Values are for the common set and differ slightly from the whole-pool figures of Appendix B.1.

	Qwen 8B	Qwen 2B	Cosmos	PE-AV	PE-V	V-JEPA 2
dimension	4096	2048	768	1024	1024	1024
Spread						
Vendi score ↑	**183.0**	87.3	74.2	8.6	8.1	10.4
participation ratio ↑	**95.1**	67.1	67.9	64.8	53.0	53.3
components for 95% var. ↑	**833**	442	114	180	157	216
Closeness						
cosine of random pairs ↓	0.172	0.250	**0.160**	0.684	0.687	0.644
distance to 10 nearest ↑	**0.308**	0.232	0.283	0.117	0.097	0.137
near-duplicate rate ↓	**0.184**	0.287	0.266	0.613	0.793	0.384
hubness skew ↓	1.16	1.29	**1.07**	1.98	1.29	2.28
Domain structure						
same domain in 10 nearest ↑	**0.773**	**0.773**	0.762	0.734	0.772	0.667
separation AUC ↑	0.767	**0.775**	0.669	0.680	0.739	0.628
silhouette ↑	0.055	**0.068**	0.040	0.025	0.060	−0.004
probe macro-F1 ↑	0.801	**0.802**	0.782	0.722	0.761	0.663

E.2 Agreement between spaces

Figure 6a compares the spaces pairwise. Linear CKA (Kornblith et al., 2019) measures whether two spaces share their global geometry, and the Jaccard overlap of the ten-nearest-neighbour sets whether they agree on which clips are close. The two Qwen3-VL models share their geometry almost entirely, at a CKA of 0.92 and a rank correlation of pairwise distances of 0.79, yet they share only 46% of their ten nearest neighbours and agree on the single nearest neighbour for 53% of clips. Between the other pairs the overlap falls to 16 to 38%, and V-JEPA 2 shares 7 to 10% with every other space. Clusterings agree accordingly: k-means with eleven clusters matches across spaces at an adjusted Rand index (Hubert & Arabie, 1985) of 0.08 to 0.48.

Near-duplicate structure is therefore a property of the encoder as well as of the pool. The graph E, and with it every value of R, is defined in the Qwen3-VL-Embedding-8B space, and another encoder would define a different graph, so a reduction of co-exposure is a reduction in that space. The same measurements support evaluating distributions in V-JEPA 2 features (Appendix F.4). It is the space that agrees least with the one the sampler works in, at a CKA of 0.47 and a neighbour overlap of 0.10, so a gain measured there cannot come from the sampler optimising the metric's own distances.

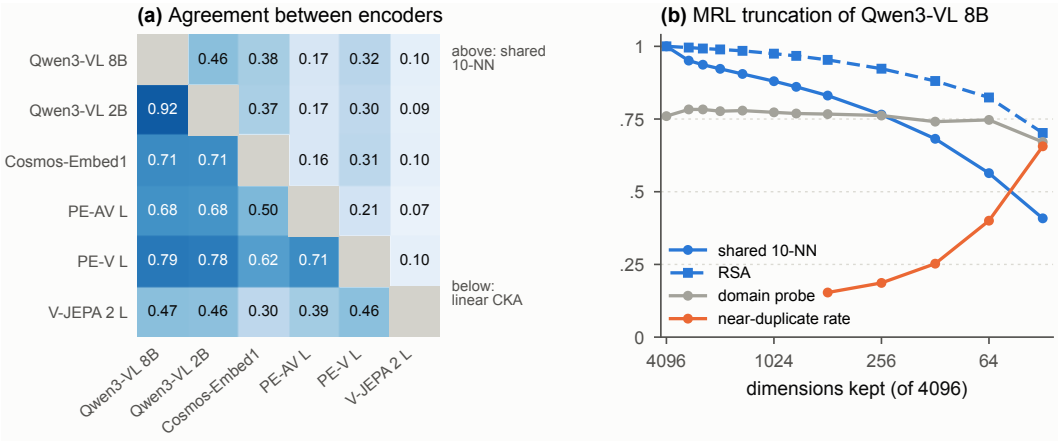


Figure 6: **(a)** Agreement between the six spaces on the shared subsample. Above the diagonal, the Jaccard overlap of the ten-nearest-neighbour sets; below it, linear CKA. **(b)** Qwen3-VL-Embedding-8B under MRL truncation to its first k coordinates, renormalised, on 23,087 clips: the shared ten nearest neighbours and RSA against the full space, the linear domain probe (macro-F1), and the share of clips whose nearest neighbour lies above cosine 0.90.

E.3 Matryoshka truncation of the chosen space

Figure 6b applies the truncation of Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) to Qwen3-VL-Embedding-8B: it keeps only the first k of the 4096 coordinates, renormalises, and compares each truncated space with the full one on the 23,087 clips of the pool that are at most five seconds long. Domain information survives almost untouched: the linear probe stays between 0.74 and 0.78 macro-F1 down to 128 coordinates. Neighbourhoods do not. The ten nearest neighbours agree with those of the full space for 88% of their members at 1,024 coordinates, 77% at 256 and 41% at 32, while the correlation of all pairwise similarities, representational similarity analysis (RSA) (Kriegeskorte et al., 2008), stays at 0.97 and 0.92 at the first two. Trustworthiness and continuity (Venna & Kaski, 2001) stay above 0.99 down to 64 coordinates, so the neighbours that change are replaced by clips that were close already. The share of clips with a neighbour above cosine 0.90 rises from 0.15 at 512 coordinates to 0.25 at 128 and 0.66 at 32: truncation pushes clips together, and a threshold set in the full space would join ever more of the pool.

Both the near-duplicate graph and the repulsion score depend on exact neighbourhoods, so both are computed in the full space. Scoring on the first 1,024 coordinates would keep 88% of each neighbour set at a quarter of the inner-product cost, a trade we did not take. MRL truncation is also not how the grid is built: the grid is cut in a PCA projection of the full space to four dimensions (Appendix E.4), which keeps a variance-ordered subspace rather than the leading native coordinates.

E.4 The projection

What the projection keeps. Table 9 reports a linear probe on the eleven domain labels at each projection dimension, normalised by the probe's accuracy in the full 4096-dimensional space (0.846). Retention is smooth and there is no elbow: $d = 3$ keeps 66% of the probe accuracy, $d = 4$ keeps 70%, $d = 8$ keeps 84%, and $d = 64$ is already at 97%. Nothing in this table selects a dimension, which is why the choice is made on the sampler's outcome instead (Section 5). Note that explained variance is a poor guide here: $d = 4$ captures 6.3% of the variance while keeping 70% of the full embedding's probe accuracy.

Stability of the projection. The projection is computed by a randomized range finder (Halko et al., 2011) with $d + 10$ Gaussian test vectors, no power iterations and seed 123, so it approximates the top principal subspace rather than computing it exactly; its four axes explain 6.3% of the variance, in shares of 2.67%, 1.73%, 1.00% and 0.87%. Refitting it with the same seed on two disjoint halves of the pool and comparing the two four-dimensional subspaces gives principal angles (Björck &

Table 9: Retention of domain-label information under PCA projection, 67,175 clips, Qwen3-VL-Embedding-8B (Li et al., 2026) features. Explained variance is as recorded; its dip from $d = 6$ to $d = 8$ cannot occur for nested PCA components and marks those two values as unreliable, and at $d = 4096$ it is 1 by construction. The silhouette (Rousseeuw, 1987) is that of the domain labels.

d	2	3	4	6	8	16	64	256	4096
probe accuracy	0.530	0.560	0.593	0.680	0.708	0.766	0.822	0.843	0.846
probe retention	0.627	0.662	0.701	0.804	0.837	0.906	0.972	0.997	1.000
explained var.	0.042	0.053	0.063	0.083	0.082	0.139	0.333	0.630	1.000
kNN-10 label agr.	0.421	0.496	0.542	0.635	0.674	0.724	0.753	0.754	0.749
silhouette	−0.088	−0.070	−0.020	−0.019	0.007	0.023	0.038	0.030	0.026

Golub, 1973) of $2.74°$, $3.52°$, $4.78°$ and $5.09°$, with singular values from 0.999 to 0.996, so a grid fitted on one half lies close to the grid fitted on the other. The two fits share their test vectors, so this measures the stability of the projection as deployed and not the identifiability of the population's principal subspace, which a spectrum this flat makes hard to pin down. Principal angles also compare subspaces and not axes. An axis-aligned grid is sensitive to a rotation inside the subspace that a clustering would ignore, so we make no claim that depends on a particular axis. The mechanism does not need one: near-duplicates are co-located by being near-identical vectors, which any sufficiently fine projection achieves, and not by the axes being well chosen.

F Evaluation protocol

F.1 Prompts and generation

Every trained configuration generates one clip for each of 500 text prompts, and a per-domain configuration generates from the soup of its eleven jobs. Generation runs in text-to-audio-video mode with 50 sampling steps and guidance weight 5.0 and produces 31 latent frames at 512 pixels, which decode to 121 frames, the 5.04-second window the encoder also sees. The seed of each clip is derived from its prompt, so two configurations differ in their weights and not in their sampling noise, and every comparison is paired by prompt.

F.2 VBench

We score the video stream on five VBench dimensions (Huang et al., 2024): imaging quality, measured with MUSIQ (Ke et al., 2021); aesthetic quality, with the LAION aesthetic predictor (Schuhmann et al., 2022); subject consistency, with DINO features (Caron et al., 2021); background consistency, with CLIP features (Radford et al., 2021); and dynamic degree, with RAFT optical flow (Teed & Deng, 2020). None of them reads the prompt, which lets them run on our clips through a file that lists them with our prompts, instead of VBench's prompt suite. The prompt-conditioned dimensions cannot: they expect VBench's short prompts or accept at most 77 text tokens, and our prompts are long captions in the style of the training data that could not be shortened without changing what the models are compared on. Motion smoothness was not scored. VBench stores subject consistency per clip as a sum over 120 frame comparisons, while its overall score divides by their number; we divide the per-clip sums by 1.2, which puts subject consistency on the same 0–100 scale as imaging quality. Aesthetic quality, background consistency and dynamic degree are on 0–1.

F.3 Intervals and the seed floor

Each difference between two configurations is the mean over the 500 prompts of the paired per-clip difference, with a 95% percentile interval from 4,000 bootstrap resamples of the prompts (Efron & Tibshirani, 1993). Resampling prompts captures the noise of the evaluation and not the noise of training, and a configuration retrained with a new seed already differs from itself by more than that interval resolves. Two configurations were therefore retrained with a second data seed, production deduplication with random sampling and full + GridReS at ($d = 3$, $M = 32$), and the seed floor of each dimension is the larger of the two gaps (Table 11). A star in Table 3 marks an interval that excludes zero, and bold one that lies entirely beyond the floor. On OpenVid-100k, random

sampling was rerun with three further seeds, which were scored on aesthetic quality and background consistency and, for one seed, on subject consistency.

F.4 Distribution metrics

Precision and recall (Kynkäänniemi et al., 2019) with density and coverage (Naeem et al., 2020) are computed in V-JEPA 2 features (Assran et al., 2025) with $k = 5$, 500 generations per configuration against 3,893 real clips, averaged over 10 resamples of the reference. We avoid the Qwen3-VL space on purpose: the grid, the graph and the score all live there, and the score maximises exactly the distances these metrics would read, so scoring in that space would flatter the sampler, and V-JEPA 2 is the candidate space that agrees least with it (Appendix E.2). As a ceiling, held-out real clips scored as if generated reach a coverage of 0.453 at 500 samples, 0.679 at 1,000 and 0.877 at 2,000.

F.5 Draw-distribution simulations

Distributional results replay the sampler offline on the embedding index, with no model in the loop. On the Kandinsky-6 SFT pool the studies use six domains, *animals*, *cartoons*, *food*, *nature*, *people action* and *people speech*, each sampled on its own as in a per-domain job, at budgets of 0.10, 0.25 and 0.50 of the domain's size; on OpenVid-100k the whole pool is one job. Each run is paired with a random permutation at the same domain, seed and budget, and its R is computed on the cross-source graph restricted to the job. A run in which either side draws no pair of E has no defined ratio; such runs are dropped and counted rather than clipped, because clipping them into a logarithm biases the aggregate (Appendix H.6). The reported value is the geometric mean of the paired ratios, with a 95% percentile interval over runs. Pool coverage is the fraction of clips drawn at least once, and source coverage the fraction of the pool's source videos with at least one clip drawn. SemDeDup and facility location reduce the pool once and then draw from what remains by random permutation. ClusterClip draws k-means clusters uniformly and caps how often a clip can be drawn, so it keeps the pool but, drawing small clusters as often as large ones, reaches less of it; on the Kandinsky-6 SFT pool CABS-DM uses the eleven domain labels as its concept set, so inside a per-domain job, which holds one domain, it draws as random sampling does. The facility-location baseline selects, once per domain, the subset that greedily maximises the summed similarity of every clip to its most similar selected clip in the embedding space (Wei et al., 2015), keeping 12% of each domain, so its coverage saturates at 0.12 once the budget exceeds the subset. Seeds and run counts are given in each caption and in Appendix H.

G Extended results

This appendix collects the ablations and trained configurations behind Sections 4.3 and 5 that do not fit in the main text.

G.1 Queue length and projection basis

R falls from 0.505 at $M = 32$ to 0.467 at 128 and 0.451 at 1024 on the Kandinsky-6 SFT pool, and from 0.854 to 0.790 on OpenVid-100k, where 128 is best. At $M = 32$ the queue is full at 93.0% of draws on OpenVid-100k against 51.9% on the Kandinsky-6 SFT pool, so saturation explains one gain and a longer memory the other, at 0.42 to 0.48 ms per draw up to $M = 256$. The lower half of Table 10 gives the occupancy-matched comparison of projection bases discussed in Appendix A.7.

G.2 Settings held fixed

Several parts of the design were held fixed rather than varied, and we list them so that the ablations are read for what they cover. The cells feeding a queue are either one cell or the 3^d block (Appendix H); the repulsion distance is always taken in the full 4096-dimensional space and never in the projection; the embedding is never truncated before projection; the reported runs always scale the projection by one scalar; and no reported run visits cells in a round-robin tour instead of drawing them independently, which leaves the CommonIT contrast of Appendix A.5 open. Each of these can be tested in simulation, without a training run.

Table 10: $R/R_{\text{rand}} \downarrow$, the geometric mean of the paired ratio with zero-R runs dropped, pooled over $B \in \{0.10, 0.25, 0.50\}N$. Queue length: four Kandinsky-6 domains at $\alpha \in \{0.5, 0.75\}$ with three seeds (66–68 runs per row, 4–6 dropped), and OpenVid-100k at $\alpha \in \{0.75, 1.0\}$ with three seeds (18 runs per row, none dropped). Projection basis: two Kandinsky-6 domains at $d = 4$, with the random projection's cell side bisected to the per-domain occupancy of the PCA grid, 15–17 runs per cell with 1–3 dropped. The basis comparison reverses sign between the two domains.

Component	Setting	Kandinsky-6 SFT pool	OpenVid-100k
Queue length	$M = 32$	0.505	0.854
	$M = 128$	0.467	**0.790**
	$M = 256$	0.457	0.792
	$M = 512$	0.452	0.799
	$M = 1024$	**0.451**	0.802
		tech	*nature*
Projection basis	PCA	**0.552**	0.515
	random, occupancy matched	0.575	**0.490**

G.3 VBench for further configurations

Table 11 reads the per-domain configurations against production curation, including those Table 3 leaves out, all at 5,000 steps, and gives the retrains behind the seed floor. On the Kandinsky-6 SFT pool, trained per domain, neither the configuration of Section 2 nor the full pool drawn at random, the grid without repulsion at $d = 3$ or ClusterClip differs from production curation beyond the seed floor on any dimension. On OpenVid-100k, GridReS and ClusterClip are within that pool's seed spread of random sampling on every dimension scored for the extra seeds.

Table 11: VBench for further trained configurations at 5,000 steps, 500 generations each, paired by prompt. Imaging quality and subject consistency are on a 0–100 scale, the rest on 0–1; VBench stores subject consistency per clip as a sum over 120 frame pairs, which we divide by 1.2. The top block gives the seed gaps: the difference between a configuration and its retrain with a second seed, whose larger value on each dimension is the seed floor of Table 3, and the largest difference among four random-sampling seeds on OpenVid-100k, on the dimensions scored for them. The OpenVid-100k runs of GridReS used $d = 3$ and $M = 32$ (Table 1).

	imaging ↑	aesthetic ↑	subject ↑	background ↑	dynamic ↑
Seed gaps, same configuration retrained with a second seed					
dedup + random	0.800	0.0012	0.365	0.0015	0.020
full + GridReS ($d = 3$, $M = 32$)	0.647	0.0021	0.294	0.0002	0.004
random, OpenVid-100k, 4 seeds	—	0.0052	0.465	0.0016	—
Kandinsky-6 SFT pool per domain, difference from dedup + random (production)					
full + random	−0.90	−0.0021	+0.45	+0.0025	−0.030
full + grid ($d = 3$)	−0.50	−0.0004	+0.31	+0.0012	−0.040
full + GridReS	−0.26	−0.0029	+0.09	+0.0012	−0.022
ClusterClip	−0.37	+0.0004	+0.18	+0.0008	−0.016
OpenVid-100k, difference from random sampling					
ClusterClip	+0.47	+0.0001	+0.36	+0.0009	−0.018
GridReS, $\alpha = 1$	+0.11	−0.0038	+0.17	+0.0006	−0.012

G.4 Precision, recall, density and coverage

Table 12 reads the trained models in V-JEPA 2 features, the candidate space that agrees least with the one the sampler works in (Appendix E.2). Trained per domain, GridReS improves on production curation on all four metrics, by 0.0158 in precision, 0.0204 in recall, 0.0280 in density and 0.0025 in coverage. The precision and density gains exceed the larger seed gap; the recall gain has the same sign in all ten reference resamples but lies within that gap. GridReS also improves on ClusterClip by 0.0242 in precision and 0.0412 in density, beyond both seed gaps, and differs from the full pool

drawn at random by less than the seed gap on every metric. The $d = 8$ control trails it on all four, by 0.0196 in precision and 0.0335 in density.

As a single job the ordering reverses. GridReS trails its random control on all four metrics, by 0.0270 in precision, 0.0213 in recall, 0.0516 in density and 0.0140 in coverage, beyond the larger seed gap on all but recall, although it leads the same control on VBench imaging quality (Section 4.3). Coverage is read against the ceiling of Appendix F.4: every configuration reaches 0.168 to 0.182, so the column ranks configurations against one another and does not say how much of the achievable coverage they reach.

Table 12: Trained models on the Kandinsky-6 SFT pool: precision and recall (Kynkäänniemi et al., 2019) with density and coverage (Naeem et al., 2020) in V-JEPA 2 (Assran et al., 2025), $k = 5$, 500 generations against $3{,}893$ real clips, averaged over 10 reference resamples (Appendix F.4). GridReS rows use the configuration of Section 2 unless marked. Bold marks the best value among the per-domain models. The seed gaps compare a configuration with its retrain with a second seed; their larger value on each metric is the reference used in the text.

Configuration	precision ↑	recall ↑	density ↑	coverage ↑
Per domain: eleven jobs souped into one model				
dedup + random *(production)*	0.7188	0.6401	0.4809	0.1760
full + random	0.7288	0.6527	0.4934	0.1762
full + grid ($d = 3$)	0.7138	0.6476	0.4784	0.1783
full + GridReS	**0.7346**	**0.6605**	**0.5089**	**0.1785**
full + GridReS, $d = 8$	0.7150	0.6554	0.4754	0.1695
ClusterClip	0.7104	0.6536	0.4677	0.1707
Single job: one model on the whole pool, 0.60 epochs				
random sampling	0.7258	0.6555	0.5067	0.1818
GridReS	0.6988	0.6342	0.4550	0.1678
Seed gaps, same configuration retrained with a second seed				
dedup + random	0.0024	0.0094	0.0091	0.0018
full + GridReS ($d = 3$, $M = 32$)	0.0114	0.0337	0.0233	0.0068

H Draw-distribution results in detail

H.1 Setting the exponent

For $\alpha < 1$, cell c is exhausted at budget B, in expectation, when $n_c \le \left(B / \sum_{c'} n_{c'}^{\alpha}\right)^{1/(1-\alpha)}$, so the trade-off between concentration and exhaustion can be read from the occupancy histogram before training. On the Kandinsky-6 SFT pool the exhausted fraction falls from 25.7% at $\alpha = 0$ to 15.0% at 0.5 and to none from 0.75 on, and on OpenVid-100k from 44.9% at $\alpha = 0$ to none at 1 (Tables 13 and 14). The two pools land on opposite sides. On the Kandinsky-6 SFT pool R is flat from $\alpha = 0$ to 0.5, at 0.488, 0.481 and 0.495 with overlapping intervals, and rises beyond, and $\alpha = 0.5$, the setting of Section 2, is the end of that range with the fewest exhausted cells. On OpenVid-100k R falls all the way to $\alpha = 1$.

H.2 Sweeps over the exponent

Table 13 gives the numbers behind Figure 4a with the columns that must be read together: R answers the redundancy question, source coverage says whether the reduction was bought by touching less of the pool, and Vendi says what happened to effective diversity, which falls as α rises toward 1, while R is lowest at $\alpha = 0.25$ and cannot be separated from $\alpha = 0$ and 0.5. At $\alpha = 1$ the grid alone reads 1.086 and not 1. The marginal is uniform there, but drawing cells independently gives a cell a binomial rather than a hypergeometric number of draws, which co-draws two members of one cell more often, by a factor of about $n_c/(n_c - 1)$, and 22.8% of the pairs of E share a cell (Table 5). The R columns drop zero-R runs, at most 11 of 144 per cell, and the marginal drops a comparison if any of its three values is zero; source coverage and Vendi are geometric-mean ratios of grid + repulsion to random over all runs. On OpenVid-100k Vendi relative to random rises from 0.881 at $\alpha = 0$ to 1.011 at $\alpha = 1$ (Table 14), the opposite of the Kandinsky-6 SFT pool. At the configuration of the

trained OpenVid-100k runs, $d = 3$ with $\ell = 0.10$, $M = 32$ and a harmonic-mean score, GridReS at $\alpha = 1$ reads 0.829 [0.802, 0.853] over the same 6 seeds and 3 budgets, at a pool coverage of 0.283, the lowest of the dimensions swept at that queue length and score, against 0.911 to 0.974 for $d = 4$, 5 and 8 at the cell sides of that sweep.

Table 13: Kandinsky-6 SFT pool at the geometry of Section 2 ($d = 4$, $\ell = 0.1429$, $M = 128$), 6 domains $\times$ 8 seeds $\times$ 3 budgets for each α and each of the two samplers, 1,440 runs. R columns are geometric means of the paired ratio to a random permutation with 95% bootstrap intervals. The *repulsion marginal* column is the paired ratio of grid + repulsion to grid only at the same α, domain, seed and budget, and is the direct test of whether repulsion adds anything. Table 2 draws its grid rows from ten seeds with their own random reference, where $\alpha = 0.5$ reads 0.749 and 0.497.

α	cells exh.	grid only	grid + repulsion	repulsion marginal	src / Vendi
0.00	25.7%	0.638 [0.565, 0.717]	0.488 [0.431, 0.550]	0.759 [0.714, 0.804]	1.040 / 1.146
0.25	21.7%	0.669 [0.607, 0.738]	**0.481** [0.433, 0.531]	0.722 [0.682, 0.762]	1.039 / 1.126
0.50	15.0%	0.737 [0.671, 0.811]	0.495 [0.450, 0.543]	0.686 [0.641, 0.734]	1.041 / 1.103
0.75	0.0%	0.836 [0.774, 0.901]	0.534 [0.493, 0.577]	0.645 [0.595, 0.694]	1.031 / 1.070
1.00	0.0%	1.086 [1.005, 1.172]	0.663 [0.607, 0.725]	**0.601** [0.561, 0.642]	1.018 / 1.018

Table 14: OpenVid-100k at the geometry of Section 2 ($d = 4$, $\ell = 0.1248$, $M = 128$), 6 seeds $\times$ 3 budgets, no zero-R runs, geometric mean of the paired ratio to a random permutation. Repulsion helps at every α, the weighting at $\alpha < 1$ hurts monotonically, and grid + repulsion clears random from $\alpha = 0.5$ on. Pooled over occupancies, queue lengths and scores, the preference between $\alpha = 0.75$ and 1 depends on the target occupancy (Appendix H.3).

α	0.00	0.25	0.50	0.75	1.00
cells exhausted	44.9%	40.1%	28.1%	10.0%	0.0%
grid only	1.402	1.277	1.170	1.090	1.024
95% CI	[1.369, 1.436]	[1.246, 1.306]	[1.146, 1.194]	[1.066, 1.113]	[1.000, 1.049]
grid + repulsion	1.207	1.060	0.917	0.830	**0.754**
95% CI	[1.169, 1.242]	[1.014, 1.101]	[0.871, 0.963]	[0.787, 0.871]	[0.719, 0.789]
repulsion marginal	0.861	0.830	0.784	0.761	0.737
Vendi / random	0.881	0.908	0.938	0.975	1.011

H.3 Budget and parameter marginals

At the configuration of Section 2, R/R_{rand} of GridReS on the Kandinsky-6 SFT pool is 0.448 at $B = 0.10N$, 0.482 at $0.25N$, 0.560 at $0.50N$ and exactly 1 by construction whenever every stream has made a whole number of passes.

Over the parameter grid at $d = 4$, with three seeds, which crosses target occupancy, α, M, the score and the neighbourhood, each contrast paired on everything else: the nearest-entry score against the harmonic mean is 0.912 on the Kandinsky-6 SFT pool and 0.985 on OpenVid-100k; the 3^d neighbourhood against a single cell is 0.927 and 0.970; $M = 128$ against $M = 32$ is 0.967 and 0.985; $\alpha = 0.5$ against $\alpha = 1.0$ is 0.727 on the Kandinsky-6 SFT pool (0.581 against 0.798 over 1,214 pairs). Target occupancy 10 is best on the Kandinsky-6 SFT pool (0.660 against 0.721 at 6 and 0.677 at 20) and 20 on OpenVid-100k (0.842 against 0.863 at 10 and 0.880 at 5). On OpenVid-100k, $\alpha = 0.75$ reads 0.847, 0.853 and 0.876 at occupancies 5, 10 and 20, against 0.913, 0.873 and 0.809 for $\alpha = 1.0$.

H.4 Per domain

Table 15 puts the per-domain simulation values beside the near-duplicate density of each domain and the values read from the training draw logs. In simulation, the domain with the densest near-duplicates, *animals*, gains most on both counts of density. On the live logs R moves toward 1, its value at every whole pass, as passes accumulate, from 0.664 at 0.60 passes per stream on *people action* to 1.056 at 4.43 on *tech*, and *animals*, at 1.02 passes, sits almost exactly at a whole pass. The geometric mean of the eight live values is 0.883. Across all eleven domains the density with same-source pairs ranges from 3.321 on *animals* to 0.033 on *interiors*, a factor of 100 (Table 7).

Table 15: Per-domain values on the Kandinsky-6 SFT pool. R/R_{rand} of GridReS comes from the simulations of Table 2, ten seeds and three budgets, with 26 to 30 of 30 runs per domain kept after zero-R runs are dropped. Pairs per clip are counted inside the domain. The live columns come from the training draw logs, which run past the evaluated checkpoint to about 80,000 draws per job; passes are per rank's stream, one eighth of the epochs.

	simulation	pairs per clip		live draw log		
Domain	R/R_{rand} (95% CI)	all	cross-source	epochs	passes	R/R_{rand}
animals	0.267 [0.235, 0.299]	3.321	0.740	8.13	1.02	0.988
people speech	0.587 [0.490, 0.693]	0.998	0.044	6.10	0.76	0.786
people action	0.431 [0.363, 0.505]	0.265	0.050	4.77	0.60	0.664
nature	0.566 [0.471, 0.669]	0.246	0.065	14.00	1.75	0.796
cartoons	0.648 [0.517, 0.806]	0.080	0.052	20.67	2.58	1.046
food	0.651 [0.562, 0.751]	0.072	0.030	12.36	1.55	0.885
music	—	1.830	0.655	26.11	3.26	0.923
tech	—	0.700	0.280	35.41	4.43	1.056

H.5 Run counts behind Table 2

Kandinsky-6 sampler rows keep 166 to 179 of 180 runs over ten seeds, the grid alone 178, GridReS 170 and ClusterClip 166, and SemDeDup keeps 81; facility location keeps 15 of 90 over five seeds. OpenVid-100k rows keep all of their 12 or 18 runs. Sampler rows agree with the ratio of means within 0.10, and for SemDeDup and facility location the ratio of means is lower still.

H.6 Reading R

R is informative only below one pass per stream, where full + GridReS reads 0.448, 0.482 and 0.560 at $B = 0.10N$, $0.25N$ and $0.50N$. It also falls when fewer clips are drawn, since a pair cannot be co-drawn if neither endpoint is drawn, so every value of R is reported beside its coverage. At $d = 3$ and $\ell = 0.10$ (Table 6), clipping zero-R runs into a logarithm instead of dropping them would move GridReS from 0.777 to 0.495 and invert the ranking of d.

H.7 Coverage of the single-job run

The single job reads $R = 0.394$ on its own draw logs against 1.016 for its random control, but it reaches 0.196 of the pool where the control reaches 0.337, and a pair cannot be co-drawn if neither endpoint is drawn. Two nulls bracket the part of the gap that coverage explains. Spread uniformly, the lower coverage γ would scale the reachable pair fraction as γ^2, a null of 0.338 that the measurement does not beat. The sampler concentrates draws on cells and near-duplicates lie inside cells, so its true pair reach exceeds γ^2, and on the slope of one paired comparison of coverage and R, $R \propto \gamma^{0.48}$, the null is 0.771 and roughly half the reduction survives; other budgets and values of α give slopes from -0.42 to 2.19. The two bounds do not separate, so the coverage-matched claim rests on the offline measurements of Table 2, where coverage is equal by construction. The run's 27,200 logged draws are 0.405 epochs of the pool, or 0.051 passes per stream.

I Implementation and compute

I.1 Code

The implementation of GridReS is available at `https://github.com/94yvwj2hrj-eng/gridres`. The sampler follows Algorithm 1. Its state, the queues, the unserved sets and the draw counts, is checkpointed with the model, and every draw is logged with its rank and step, which is what the live-log measurements of Appendix H.4 read.

I.2 Configuration

Table 16 lists the configuration of every reported run.

Table 16: Configuration of the reported runs. Unless a row says otherwise, the Kandinsky-6 SFT pool and OpenVid-100k share it.

Encoder	Qwen3-VL-Embedding-8B, 4096 dimensions, unit-norm rows, computed once and cached.
Projection	PCA to $d = 4$ fitted on the whole pool with seed 123, centred at the median and divided by the 99.9th percentile of $\|z\|$.
Cell side	Bisected to a target mean occupancy, 10 on the Kandinsky-6 SFT pool ($\ell = 0.1429$, 6,719 occupied cells) and 20 on OpenVid-100k ($\ell = 0.1248$, 4,899 cells).
Cell weighting	$\alpha = 0.5$ on the Kandinsky-6 SFT pool and 1.0 on OpenVid-100k.
Memory	One first-in-first-out queue per cell of the last $M = 128$ draws from its 81-cell neighbourhood, one per rank, checkpointed with the model.
Score	Full-space cosine distance to the nearest queue entry (1 if empty), plus $0.1/(1+n_x)$, n_x the draws of x so far.
Within-cell draw	One unserved member, in proportion to its score; a served cell retires until all are spent. One stream per rank over the whole pool.
Training	Kandinsky-6 Lite, 3B parameters, 5,000 steps (OpenVid-100k also 10,000) on 8 GPUs, one clip per GPU per step; eleven souped per-domain jobs or one single job (0.60 epochs).
Optimisation	Learning rate 10^{-5} (10^{-6} for audio-side groups), weight decay 0.001, betas $(0.9, 0.95)$, gradient clip 1.0, 500 warm-up steps, EMA 0.999, caption dropout 0.1. On OpenVid-100k the audio loss is off and the audio and fusion groups are frozen.
Evaluation	500 prompts, 50 steps, guidance 5.0, five VBench dimensions, 4,000 bootstrap re-samples, seed floor from two retrains (Appendix F).
($d = 3$, $M = 32$) runs	$\ell = 0.10$, harmonic mean of queue distances weighted by 0.9^{age} and a soft cell membership $\exp(-\|z-\bar{z}_c\|^2/\ell^2)$; with replacement within cells on the Kandinsky-6 SFT pool, by permutation on OpenVid-100k (all its trained runs). The Kandinsky-6 grid-only control uses the same grid.

I.3 Compute

Every training job runs on one node with eight A100 80 GB GPUs. A per-domain configuration is eleven such jobs, souped into one model, and a single job is one; generation for evaluation uses the same GPU type. The sampler runs in the data loader of each rank. A draw costs 0.33 ms at $d = 3$ and 2.54 ms at $d = 8$, and 0.42 to 0.48 ms at $d = 4$ for queue lengths from 32 to 256, which is small next to a training step. The queues of one rank hold at most 128 entries for each of the 6,719 occupied cells of the Kandinsky-6 SFT pool, 860,032 in all. Embedding the pool with Qwen3-VL-Embedding-8B is a one-off offline cost, and no encoder runs during training.

J Example clips

Figure 7 extends Figure 3D; clips in which a person is identifiable are left out.

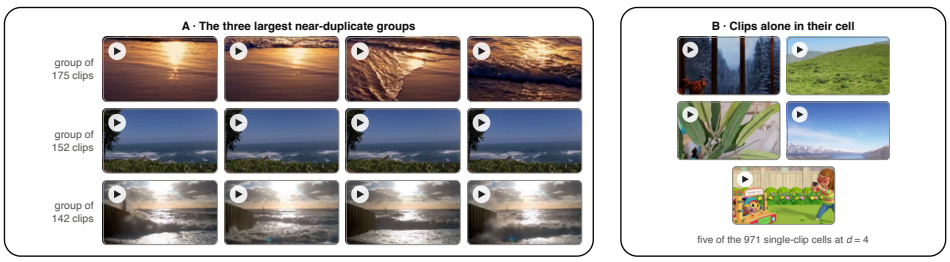


Figure 7: Example clips from OpenVid-100k. **(A)** Four members of each of the three largest near-duplicate groups, connected components of the cosine > 0.94 graph whose members show the same scene at different moments or near-identical footage. **(B)** Clips alone in their cell at $d = 4$, whose per-clip weight $n_c^{\alpha-1}$ is largest for any $\alpha < 1$.