IN SEARCH OF LOST CONSISTENCY IN DATASET DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset Distillation (DD) condenses large datasets into compact synthetic ones, promising comparable performance at a fraction of the training cost. This would make hyperparameter optimization and neural architecture search feasible even for increasingly large foundation models. Recent DD methods report impressive accuracy, yet a critical question remains: *Does dataset distillation truly deliver on its promises?* We argue that accuracy alone is insufficient; equally crucial is consistency, the extent to which distilled data rank hyperparameters or architectures as the full dataset does. To study this, we introduce the *Dataset Distillation Testbed* (*DataDistillBed*), which spans 13 DD methods, five datasets up to ImageNet-1K, and diverse hyperparameter configurations and neural architectures. Our results show that DD delivers on its promises only in part. At matched training budgets, distilled data do not yet replace the full dataset. As proxies for search, however, they rank training hyperparameters largely as the full dataset does (median Spearman correlation 0.82 at 50 images per class), though architectures less faithfully (0.52). Case studies that use DD to retrace the design of ConvNeXt and EfficientNet show its practicality and limitations. We further analyze how consistency relates to accuracy, compression ratio, training cost, and individual hyperparameters. Building on these findings, we design a search strategy that uses distilled data as a cheap first stage and matches the strongest full-data baselines at a third to half the cost. Because accuracy depends strongly on training hyperparameters, we also propose an evaluation protocol that compares DD methods under shared, randomly drawn training configurations and reports consistency alongside accuracy. We hope that the release of *DataDistillBed* will catalyze more rigorous, reproducible, and practically effective DD methods.

1 INTRODUCTION

Dataset Distillation (DD) (Wang et al., 2018), also known as Dataset Condensation, synthesizes a compact training set, often only a few images per class, from which a model should learn nearly as well as from the full dataset. Its main appeal is efficiency: if a small synthetic set can stand in for the full data, every training run becomes cheap, and workloads that train many models, such as hyperparameter optimization (HPO) and neural architecture search (NAS), become affordable even for increasingly large models (Zhao et al., 2021; Zhao & Bilen, 2021; 2023; Ding et al., 2024). The field has advanced rapidly, with methods that scale to ImageNet-1K (Yin et al., 2023; Sun et al., 2024) and some that claim lossless distillation (Guo et al., 2024).

Despite this progress, a critical question remains: *does dataset distillation truly deliver on its promises?* The standard evaluation of DD does not answer it. It reports the test accuracy of a model trained on the distilled data with a recipe that each method chooses on its own (Li et al., 2025), often under-specified and sometimes costlier than a full training run on the original data. Such a number shows neither whether distilled data save training cost nor whether they help search. In HPO and NAS, distilled data serve as a cheap proxy: many candidate configurations are trained and compared on them, and only the best is trained on the full dataset. The proxy need not reach full accuracy, but it must rank the candidates as the full dataset would. We call this property *consistency* and argue that it is as crucial as accuracy for judging DD.

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.

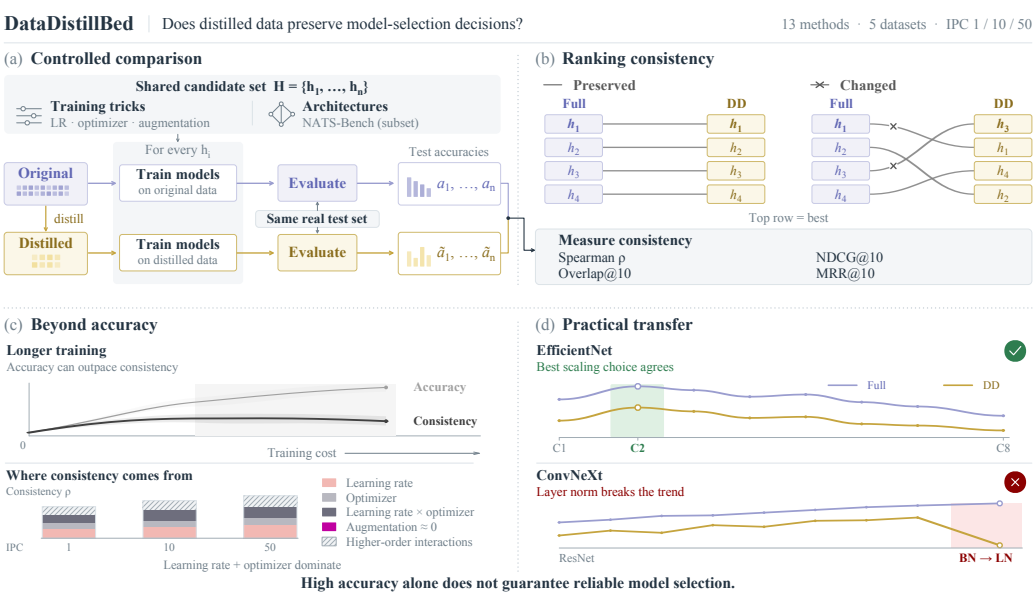

Figure 1: **Overview of DataDistillBed**. (a) The same candidate configurations (training tricks, NATS-Bench architectures) are trained on original and distilled data and tested on the same real test set. (b) Consistency is the agreement of the two accuracy rankings, globally (Spearman's ρ) and at the top (NDCG, Overlap, MRR at $k = 10$). (c) Accuracy and consistency can diverge: longer training raises accuracy while consistency saturates, and distilled data preserve the effect of only some hyperparameters (Figures 8 and 9). (d) Distilled data recover the best EfficientNet scaling but fail at the layer-normalization step of the ConvNeXt roadmap (Figure 4). Mini-plots are schematic.

Consistency, however, has rarely been measured: evidence on it is limited to architectures and mixed (Zhao et al., 2021; Zhao & Bilen, 2021; 2023; Cui et al., 2022; Ding et al., 2024), and to our knowledge, no study has systematically examined whether distilled data preserve the ranking of training hyperparameters, the main object of HPO. We therefore introduce **DataDistillBed** (Dataset Distillation Testbed), which evaluates DD beyond accuracy (Figure 1). Following DomainBed (Gulrajani & Lopez-Paz, 2020), it tests all methods under one randomized protocol: candidate configurations are sampled from a search space and trained from scratch on both the full and the distilled training sets, and the two resulting rankings are compared, globally with Spearman's correlation and at the top with three top-k metrics. We use two search spaces drawn from practice, training hyperparameters built on Bag of Tricks (He et al., 2019) and architectures from NATS-Bench (Dong et al., 2022). Because the testbed needs only the distilled data, any DD method can be evaluated in the same way. Its first release covers 13 methods and five datasets from CIFAR-10 to ImageNet-1K, and our experiments train more than 280,000 networks in total.

Our study shows that DD delivers on its promises only in part. At matched training budgets, the full dataset wins unless the budget is small, so distilled data do not yet replace it for training a model; their accuracy also depends strongly on the training recipe. As proxies for search, however, they are more useful than their accuracy suggests: they rank training hyperparameters largely as the full dataset does, though architectures less faithfully. Accuracy and consistency are related but not interchangeable; on CIFAR-10, for example, TESLA (Cui et al., 2023) yields higher accuracy than DSA but ranks configurations less faithfully. Consistency must therefore be measured directly.

Our contributions are as follows:

- We formalize consistency as an evaluation criterion for DD that complements accuracy, and build DataDistillBed to measure it at scale under a shared, randomized protocol (Section 3).
- We evaluate DD on both fronts: its accuracy under varying training recipes and at matched training budgets, and its consistency for training hyperparameters, architectures and two case studies that retrace the design of ConvNeXt (Liu et al., 2022b) and EfficientNet (Tan & Le, 2020). We also analyze how consistency relates to accuracy (Sections 4 and 5).

- We turn the measurements into actionable guidance: which hyperparameters can be searched on distilled data, a search strategy that uses distilled data as a cheap first stage and matches the strongest full-data baselines at a third to half the compute, and a protocol for reporting the accuracy of DD methods (Section 6).

2 Related works

Dataset Distillation. Dataset Distillation was introduced by Wang et al. (2018) to replace a large training set with a compact synthetic one, and this bilevel meta-learning formulation was later made tractable with kernel ridge regression (Nguyen et al., 2021). Subsequent methods can be grouped into gradient matching (Zhao et al., 2021; Zhao & Bilen, 2021; Kim et al., 2022; Liu et al., 2023), trajectory matching (Cazenavette et al., 2022; Guo et al., 2024; Du et al., 2023; Liu et al., 2024), and distribution or feature matching (Zhao & Bilen, 2023; Wang et al., 2022; Liu et al., 2025; Zhang et al., 2024; Wang et al., 2025; Cui et al., 2026). Recent work has also emphasized scaling and efficiency, including ImageNet-scale distillation (Yin et al., 2023), constant-memory optimization (Cui et al., 2023), frequency-domain compression (Shin et al., 2023), multi-backbone transfer (Shao et al., 2024a), self-supervised compression (Zhou et al., 2024a), and other efficient or robust variants (Shao et al., 2024b; Sun et al., 2024; Moser et al., 2025; Xiao & He, 2024; Xue et al., 2025; Liu et al., 2022a). Methods from 2025 and 2026 revisit how distilled data are represented, matched and synthesized (Shin et al., 2025; Chen et al., 2025; Zou et al., 2025; Tan & Slade, 2026; Chan-Santiago & Shah, 2026; Tan et al., 2026).

Benchmarks and Evaluation of Dataset Distillation. Existing evaluations of DD mostly judge distilled data by the quality of a model trained on them. Benchmarks compare methods under standardized protocols (Cui et al., 2022; Li et al., 2025; Zhong et al., 2026) and, together with further analyses (Qin et al., 2024; Dey et al., 2026; Yang et al., 2024), show that the reported accuracy depends heavily on how the model is trained, in particular on augmentation and soft labels; other benchmarks measure adversarial robustness, privacy and fairness (Wu et al., 2025; Zhou et al., 2024b; Chen et al., 2024). We ask a different question: whether distilled data rank candidate configurations as the full dataset does, which is what hyperparameter optimization and architecture search need. Evidence on this question is limited to architectures and mixed. Method papers report high correlations in a space of 720 ConvNets (Zhao et al., 2021; Zhao & Bilen, 2021; 2023), whereas DC-Bench (Cui et al., 2022) finds negative correlations for five methods and 100 NAS-Bench-201 architectures on CIFAR-10. HCDC (Ding et al., 2024) designs a new condensation method to preserve rankings and tests it only on architecture spaces. With DataDistillBed, we conduct, to our knowledge, the first large-scale study of whether existing DD methods preserve the ranking of training hyperparameters, with consistency measured both globally and at the top of the ranking.

3 Dataset Distillation Testbed

DataDistillBed tests whether distilled data rank candidate training configurations, such as hyperparameter settings or network architectures, in the same order as the full dataset. It adapts the code and experimental protocol of DomainBed (Gulrajani & Lopez-Paz, 2020), a testbed for domain generalization, to dataset distillation. Its first release comprises 13 DD methods, five datasets, two search spaces (Section 3.3) and four consistency metrics (Section 3.2).

3.1 Implementation of Testbed

Datasets. We use CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) at 32×32, Tiny-ImageNet (Le & Yang, 2015) at 64×64, ImageNette (Howard, 2019) (ten ImageNet classes) at 128×128, and ImageNet-1K (Deng et al., 2009) at 224×224. Distilled sets have 1, 10 or 50 images per class (IPC). All runs are tested on the real test split (the validation split for the ImageNet datasets).

DD methods. The 13 methods form two families. The eight *ConvNet-based* methods match what real data produce in a ConvNet: gradients for DC (Zhao et al., 2021) and DSA (Zhao & Bilen, 2021), feature distributions for DM (Zhao & Bilen, 2023) and NCFM (Wang et al., 2025), and training trajectories for MTT (Cazenavette et al., 2022), TESLA (Cui et al., 2023), ATT (Liu et al., 2024) and FreD (Shin et al., 2023). They are evaluated on the standard DD ConvNet (128 channels, instance

normalization, three to five blocks by resolution). The five *ResNet-based* methods, SRe^{2}L (Yin et al., 2023), SCDD (Zhou et al., 2024a), CVDD (Cui et al., 2026), G-VBSM (Shao et al., 2024a) and WMDD (Liu et al., 2025), synthesize images from networks pretrained on the full data. A ResNet-18 learns from their images by knowledge distillation (KD) (Hinton et al., 2015), with soft labels for every augmented batch from a ResNet-18 teacher pretrained on the full training set. We use the authors' released distilled datasets or regenerate them with the official code, and evaluate a method on a dataset only if one of the two is available (Appendix A).

Paired runs. For each dataset and family, we train every sampled configuration from scratch on the full training set and on every distilled set of the family. Unless noted otherwise, the two runs share the hyperparameters and random seed but not the number of epochs: 100 on the full data (50 on ImageNette, 300 on ImageNet-1K) and 50 to 1000 on distilled data. Only the distilled runs of a ResNet-based method use its teacher. Each side ranks the configurations by their final test accuracy.

Trial seeds. As in DomainBed, configurations are sampled at random (Bergstra & Bengio, 2012), and each experiment is repeated with three trial seeds, which redraw every random choice, from the configurations to the seeds of the runs. Unless noted otherwise, a trial seed has 100 configurations (20 on ImageNet-1K); consistency is measured within each trial seed and averaged over the three. In total, our experiments train more than 280,000 networks.

3.2 Consistency in Dataset Distillation

When distilled data serve hyperparameter optimization or architecture search, candidate models are compared on them and only the best is trained on the full dataset (Cui et al., 2022; Ding et al., 2024). The distilled data must therefore rank the candidates as the full dataset does. We call this agreement *consistency*; an evaluation with a single training recipe cannot measure it.

Definition. Every setting of a training run other than the data and the learned parameters is a hyperparameter, including the optimizer, the augmentation and the architecture. Let $\mathrm{Acc}(h, D)$ be the test accuracy of a network trained with configuration h of a search space H on a training set D. An ideal distilled dataset D_{syn} orders all of H as the full training set D_{full} does, so that the best configuration is the same on both. As H is too large to evaluate, we sample $H_s \subset H$ at random and define consistency as the similarity of the two rankings on it:

$$\mathrm{Sim}\big(\{\mathrm{Acc}(h, D_{\mathrm{full}})\}_{h\in H_s},\ \{\mathrm{Acc}(h, D_{\mathrm{syn}})\}_{h\in H_s}\big). \tag{1}$$

Metrics. We measure this similarity globally with Spearman's rank correlation ρ and at the top of the ranking, with $k = 10$, by NDCG@k (quality of the ordered top list), Overlap@k (recovery of the true top k) and MRR@k (how early the distilled picks appear in the full-data ranking; Appendix B).

3.3 Search Spaces

Tricks search space. Our space of training hyperparameters builds on Bag of Tricks (He et al., 2019), whose refinements improved ResNet-50 (He et al., 2016) on ImageNet and reflect common practice. Its 12 dimensions are batch size, learning rate, optimizer, weight decay, warm-up, learning-rate schedule, no-bias decay, distillation temperature, augmentation, label smoothing, mixup or CutMix, and mixed precision (Appendix C).

Architecture search space. We use the topology search space of NATS-Bench (Dong et al., 2022) (15,625 cells) on CIFAR-10 and CIFAR-100. Each configuration pairs an architecture with two training switches drawn at random, DSA augmentation (Zhao & Bilen, 2021) and mixup (Zhang et al., 2018); the other hyperparameters are fixed (Section 4.2.2).

4 Experiments

We first examine what the accuracy reported by DD methods measures (Section 4.1), and then test whether distilled data deliver on their promise as a proxy for search (Section 4.2).

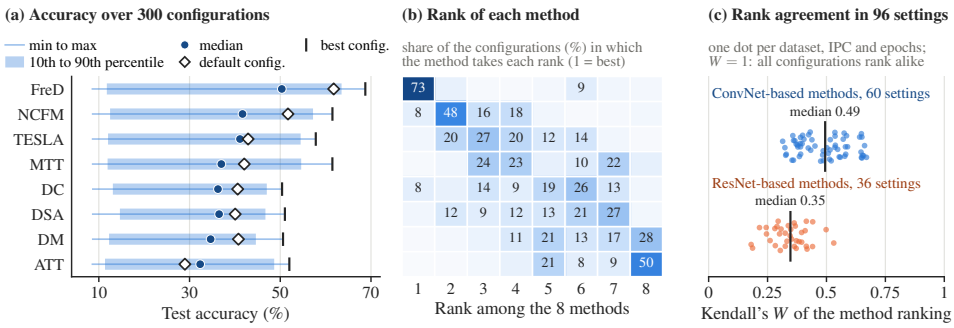

Figure 2: **The accuracy of distilled data, and the ranking of DD methods, depend on the training hyperparameters.** (a) Accuracy of each method under the same 300 sampled configurations (CIFAR-10, ConvNet, IPC 10, 300 epochs). (b) How often each method takes each rank when all eight share a configuration. (c) Kendall's W of the method rankings across configurations in 96 settings.

4.1 What Does Reported Accuracy Mean?

Two observations qualify the ever higher accuracy that recent DD methods report.

4.1.1 Sensitivity to training hyperparameters

The accuracy of a distilled dataset is that of a network trained on it, with hyperparameters that each DD method chooses on its own (optimizer, learning rate, schedule, augmentation, epochs). These choices differ from paper to paper (Li et al., 2025) and are often under-specified: only 4 of the 13 papers state the number of training epochs, which the other nine set in their code at 40 to 2,000 (Table 4). To measure their effect, we train the CIFAR-10 sets (IPC 10) of the eight ConvNet-based methods under the same 300 sampled configurations (Figure 2). For every method, accuracy differs by 32 to 52 points between the 10th and 90th percentile configurations, whereas the medians of the eight methods lie within 18 points of each other (Figure 2a). The ranking of the methods changes with the configuration too (Figure 2b): FreD ranks first under 73% of the configurations, but MTT is third or fourth under 47% and seventh under 22%. Across the 96 combinations of dataset, IPC and training length with sufficient coverage, Kendall's W (Kendall & Babington Smith, 1939) of the method rankings has a median of 0.49 for the ConvNet-based and 0.35 for the ResNet-based methods, far from the 1 of a configuration-independent ranking (Figure 2c). Full-data training is similarly sensitive, so the lesson is not that distilled data are fragile, but that an accuracy, and any leaderboard of such numbers, is meaningful only with its training configuration and tuning effort (Section 6.3).

4.1.2 Accuracy at matched training budgets

DD promises comparable accuracy at a fraction of the training cost, so the fair comparison with the full dataset is at equal training budgets, which Figure 3 counts in processed training images. Distilled sets lead only at small budgets: on CIFAR-10 the last lead disappears at about 3.4% of a full training run, and on CIFAR-100 with ResNet-18 no method leads at any budget. On ImageNette, whose training set has only 9,469 images, the lead lasts longer, but the full dataset still overtakes every distilled set before 100%. Beyond the crossover the gap grows: at 200%, the best distilled set trails by 4.4 to 16.3 points (Appendix D). Nor do the methods' own protocols stay at small budgets: their 36 official evaluation budgets (triangles) range from 3% to 800% of a full run; 32 lie beyond the crossover, and 11 amount to at least one full run (Table 4).

Distilled data thus cannot yet replace the full dataset for training an equally accurate model, and a reported accuracy is hard to interpret without its training configuration. A proxy for search, however, need not reach full accuracy; it only needs to rank candidate configurations as the full dataset does. Nor need the most accurate distilled dataset be the best proxy (Section 5).

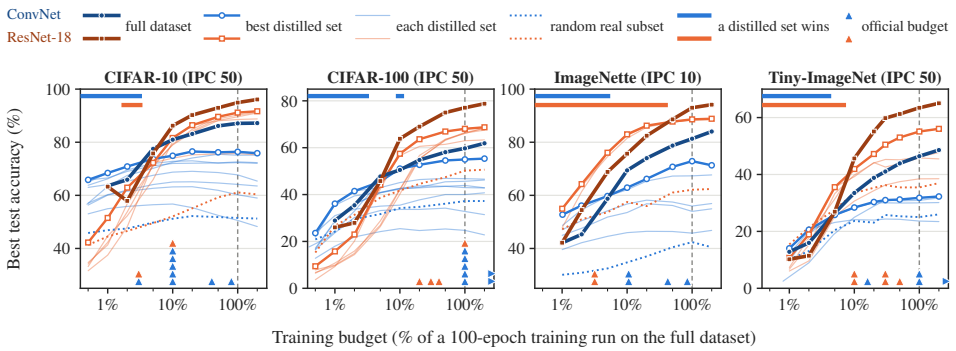

Figure 3: **Distilled data versus the full dataset at matched training budgets.** A budget of $b\%$ is $b\%$ of a 100-epoch run on the full dataset, that is, $b \times N$ processed training images (N: number of training images in the full dataset). Each point is the best final test accuracy over 10 fixed configurations, each trained from scratch at that budget. Dashed line: one full training run. Triangles: the official evaluation budget of each method at this IPC (arrows: beyond 200%; Table 4). ResNet-based distilled sets train with a teacher, which the axis does not count. Details: Appendix D.

Table 1: **Tricks consistency of all 13 DD methods at IPC 50.** Spearman's ρ over all configurations; NDCG, Overlap (Ovl.) and MRR over the ten best (chance: NDCG 0.47 to 0.73, Ovl. 0.10, MRR 0.05; MRR cannot exceed 0.29). Means over three trial seeds of about 100 configurations (ResNet-18 on Tiny-ImageNet: one seed; std of ρ over seeds at most 0.09). Distilled-data epochs: 350 on CIFAR, 200 (ConvNet) and 300 (ResNet-18) on Tiny-ImageNet. Bold: best per family. *Random subset*: 50 real images per class, 300 epochs, mean of three subsets.

	CIFAR-10				CIFAR-100				Tiny-ImageNet			
Method	ρ	NDCG	Ovl.	MRR	ρ	NDCG	Ovl.	MRR	ρ	NDCG	Ovl.	MRR
ConvNet-based methods (ConvNet, trained without a teacher)												
ATT	0.833	0.988	0.70	0.240	0.852	0.915	0.43	0.173	0.843	0.872	0.43	0.121
DC	0.824	0.978	0.53	0.187	0.853	0.929	0.37	0.110	–	–	–	–
DM	0.833	0.987	0.63	0.211	**0.952**	**0.984**	0.67	0.261	–	–	–	–
DSA	0.838	0.986	0.57	0.196	0.949	**0.984**	0.67	**0.269**	–	–	–	–
FreD	0.872	0.984	0.63	0.243	0.813	0.927	0.47	0.182	–	–	–	–
MTT	0.820	0.982	0.60	0.226	0.883	0.944	0.47	0.210	0.924	0.933	0.63	0.227
NCFM	**0.914**	**0.992**	**0.73**	**0.272**	0.932	**0.984**	**0.70**	0.262	0.871	0.900	0.40	0.115
TESLA	0.816	0.986	0.60	0.231	0.837	0.887	0.27	0.170	**0.926**	**0.948**	**0.67**	**0.232**
ResNet-based methods (ResNet-18, trained with knowledge distillation from a teacher)												
CVDD	**0.681**	**0.960**	**0.27**	**0.080**	0.811	0.894	0.33	0.143	0.851	0.974	0.60	0.264
G-VBSM	0.635	0.946	0.23	0.076	0.804	0.928	0.47	**0.195**	0.876	**0.981**	0.60	0.266
SCDD	0.664	0.946	0.23	0.073	0.843	0.955	0.50	0.189	–	–	–	–
SRe^{2}L	0.654	0.947	0.20	0.070	**0.844**	0.959	**0.53**	0.193	0.857	0.979	**0.80**	**0.279**
WMDD	0.652	0.948	0.23	0.074	0.841	**0.960**	**0.53**	0.193	**0.890**	0.917	0.60	0.255

4.2 Does Dataset Distillation Deliver on Its Promise?

We test this promise, a consistent proxy for hyperparameter and architecture search, in the tricks and NAS search spaces and in two practical case studies.

4.2.1 Tricks evaluation

Table 1 reports all four metrics at IPC 50. The distilled sets preserve the global order of the configurations well (ρ from 0.64 to 0.95, NDCG from 0.87 to 0.99) but recover the best configurations less reliably: on CIFAR-10, the ResNet-based methods reach an Overlap of 0.20 to 0.27 and an MRR of 0.07 to 0.08, against 0.10 and 0.05 at chance. The most consistent ConvNet-based method is NCFM on CIFAR-10, DM and DSA on CIFAR-100 (NCFM when trained for 200 epochs or fewer), and TESLA and MTT on Tiny-ImageNet. A random real subset of the same size is a strong reference:

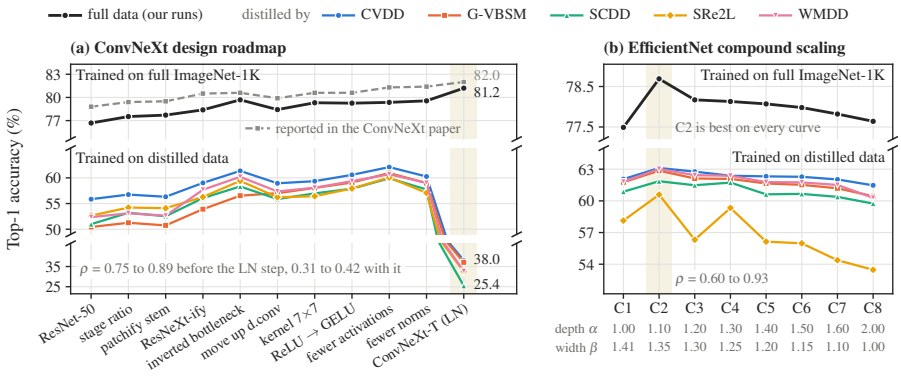

Figure 4: **Practical case studies on ImageNet-1K.** (a) Along the ConvNeXt design roadmap, distilled data track the full-data trend until the final step, where introducing LayerNorm (LN) helps on full data but makes every distilled set collapse. (b) Across eight fixed-resolution EfficientNet scaling configurations, full and distilled data agree on the best one, C2. ρ: Spearman correlation with our full-data runs (range over the five DD methods).

every ConvNet-based set ranks the CIFAR-10 configurations better than it, but on CIFAR-100 four have a lower ρ (all but TESLA still recover more of the ten best configurations), and at 100 training epochs, the random subset matches the median distilled set on both datasets. Consistency generally rises with IPC, with a median ρ over the 26 method and dataset pairs of 0.75, 0.80 and 0.82 at IPC 1, 10 and 50 (Figure 7 in Appendix E); averaged over training lengths, NCFM is the most consistent at the larger IPCs and FreD at IPC 1 (for both, IPC is an equivalent storage budget).

Higher-resolution datasets. Many configurations fail to train on the full data of ImageNette and ImageNet-1K (29% and 23% stay below twice the chance accuracy, against 10 to 19% on the other datasets), so we also report ρ_{v} over the configurations that do train (Table 10 in Appendix E). On ImageNette, the distilled sets separate the failing configurations from the others (ρ from 0.69 to 0.87) but rank the viable ones much less faithfully (ρ_{v} from 0.22 to 0.39 at IPC 1 and from 0.50 to 0.65 at IPC 10, whereas the two differ by at most 0.08 on the other datasets). On ImageNet-1K, where we rank the configurations of the three seeds together, the four distilled sets reach a ρ of 0.65 to 0.67 and a ρ_{v} of 0.77 to 0.79, and seven of the ten configurations that each selects are among the ten best.

4.2.2 NAS EVALUATION

Each architecture configuration is trained on the distilled data and for 30 epochs on the full dataset. The two training switches typically raise the accuracy of a run on distilled data by a few points but lower that of the 30-epoch full-data run by 3 to 5 points. A ranking that mixes architectures and switches is therefore dominated by the switches, and its Spearman correlation stays near zero for most methods (Appendix F). We thus compute the correlation within each augmentation setting and average over the settings, which isolates the architecture ranking.

Architecture rankings are preserved less well than hyperparameter rankings (Table 11 in Appendix F): the median correlation over methods rises from 0.17 at IPC 1 to 0.52 at IPC 50, against 0.75 to 0.82 in the tricks space, and several methods stay near or below zero at IPC 1. On CIFAR-100 at IPC 50, all five ResNet-based methods reach 0.75 to 0.95, possibly because knowledge distillation provides a more stable training signal; on CIFAR-10 the two families perform comparably.

4.2.3 CASE STUDIES

We also test two practical design processes on ImageNet-1K: the ConvNeXt design roadmap (Liu et al., 2022b), from the ResNet-50 baseline to ConvNeXt-T, and the compound scaling of EfficientNet-B0 (Tan & Le, 2020) over depth and width. Full-data runs use standard supervised learning and distilled-data runs use knowledge distillation; all other training settings are shared, except that the EfficientNet variants train for 100 full-data and 300 distilled-data epochs.

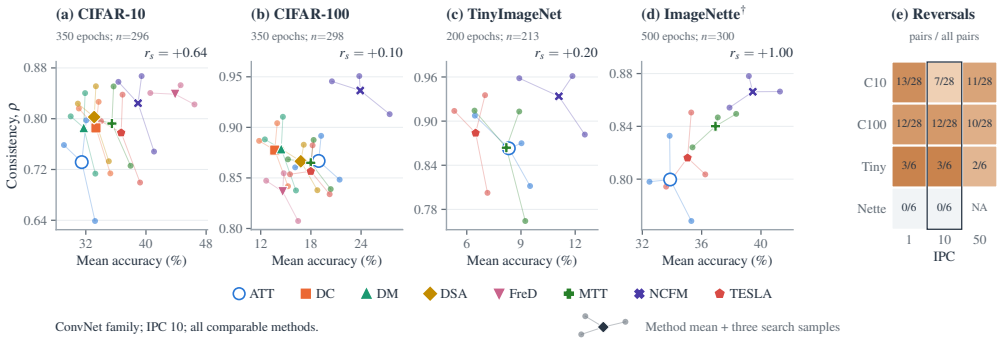

Figure 5: **Higher accuracy does not guarantee higher rank consistency.** (a) to (d): Test accuracy and consistency ρ of the ConvNet-based methods at IPC 10. Small dots: three configuration samples, which all methods share (not independently distilled datasets); large markers: method means, with spokes to their samples; r_s: rank correlation of the method means; n: configuration-trial pairs; local axis ranges. (e) Reversed method pairs out of all pairs, from the method means within each dataset and IPC; the outlined column corresponds to (a) to (d). †Older search space.

Along the ConvNeXt roadmap (Figure 4a), distilled data track the full-data trend until the final step replaces batch normalization with layer normalization: this helps on the full data, but every distilled-data curve drops sharply, so DD misses the overall improvement of the roadmap. Across the fixed-resolution EfficientNet configurations (Figure 4b; Table 13), full and distilled data agree on the best one, C2, so distilled data can guide depth and width scaling at the native resolution. DD can thus be reliable for structural search within the resolution and normalization regime of distillation.

Across these evaluations, DD delivers on its promise only in part: distilled data rank training hyperparameters largely as the full dataset does, but architectures less faithfully.

5 Rethinking the Pursuit of Accuracy in Dataset Distillation

Is the most accurate distilled dataset also the best search proxy? If so, accuracy leaderboards would suffice. In the tricks space, we show they do not, across methods and along the training budget.

Across methods. Figure 5 compares the ConvNet-based methods at IPC 10 on four datasets, with the pipeline, IPC and training budget fixed within a panel. Accuracy and consistency are related but not interchangeable. On CIFAR-10, TESLA is more accurate than DSA (36.7% versus 33.1%) but less consistent ($\rho = 0.778$ versus 0.803); on CIFAR-100, FreD is more accurate than DC (14.6% versus 13.7%) but less consistent (0.836 versus 0.878). Both reversals hold in all three configuration samples. Reversals of the method order appear in nine of the eleven available ConvNet settings, including all three IPCs of Tiny-ImageNet (Figure 5e), whereas on ImageNette the two orders agree. Accuracy can track consistency, but this cannot be assumed: choosing a distilled dataset by its accuracy can select a less faithful proxy, so consistency has to be evaluated directly.

Along the training budget. On CIFAR-10 and CIFAR-100, longer training on the distilled data keeps raising accuracy, whereas consistency stabilizes or declines after 200 to 300 epochs (Figure 8 in Appendix H). Many training epochs are thus unnecessary when consistency is the goal, and for the ConvNet-based methods the extra accuracy even comes at the cost of consistency. Rather than chasing accuracy over hundreds or thousands of epochs, DD methods that aim at efficient training should seek higher consistency with fewer training steps.

6 Actionable Insights

The tricks runs on CIFAR-10, CIFAR-100 and Tiny-ImageNet (ConvNet, and ResNet-18 with knowledge distillation, KD; about 300 configurations per setting) also yield three practical lessons.

6.1 WHICH HYPERPARAMETERS CAN BE SEARCHED ON DISTILLED DATA?

Because the search space samples every hyperparameter independently, ρ computed on ranks splits into one term per hyperparameter, one term for the interaction of learning rate and optimizer, and a remainder of higher-order interactions (Figure 9 in Appendix I). The learning rate, the optimizer and their interaction carry 70% of ρ with the ConvNet pipeline and 77% with the ResNet-18 pipeline; augmentation, the schedule, weight decay and all other hyperparameters contribute almost nothing, and higher-order interactions supply the rest. Distilled data thus preserve the optimization side of the search space but not the regularization side. The proxy rejects poor optimizers but cannot tell AdamW from Adam and NAdam, and with ResNet-18 it over-values heavy augmentation; letting it choose the optimizer or the augmentation costs up to 7 and 6 points of full-data accuracy. A distilled proxy can therefore be trusted for the learning-rate scale, for rejecting optimizers and for combining the two, while regularization hyperparameters should be fixed or re-tuned on real data.

6.2 DISTILLED DATA AS THE FIRST STAGE OF A CHEAPER SEARCH

Since every configuration was trained on the full dataset and on every distilled set, any search strategy can be simulated by table lookup, as in tabular HPO benchmarks (Eggensperger et al., 2022; Pineda Arango et al., 2021). A strategy spends a compute budget, counted in full-data 100-epoch runs including all distilled evaluations, and recommends one configuration, whose regret is its full-data accuracy gap to the best configuration in the pool. We compare full-data baselines (random search (Bergstra & Bengio, 2012), TPE (Bergstra et al., 2011), and successive halving (Jamieson & Talwalkar, 2016) and Hyperband (Li et al., 2018) over early stopping) with a *ladder* built from our findings: all candidates train at IPC 1 for 50 epochs, about a thousandth of a full-data run each; the best are promoted to IPC 10 for 100 epochs, then to 20 full-data epochs and finally to 100. The rise of consistency with IPC orders the stages, its early saturation keeps the distilled stages short, and the full-data stages re-rank the survivors on the hyperparameters that distilled data cannot see. The ladder is the best strategy at every budget (Figure 10 in Appendix J): its regret is 1.7, 1.2 and 0.5 points at 3, 5 and 10 full-data runs, the best full-data baseline needs 2.2 to 3.1 times more compute to match it, and it beats successive halving in 91 to 100% of the 33 conditions. Run live on CIFAR-100 with MTT and WMDD (five pools of 300 new configurations each), the ladder was the most accurate in all four settings, 4.5 and 2.0 points ahead of successive halving at 5 full-data runs (Appendix J). These results concern selection among about 300 configurations and exclude the one-time distillation cost.

6.3 HOW SHOULD DD METHODS REPORT ACCURACY?

Section 4.1.1 showed that a reported accuracy depends on the training configuration. It also depends on the tuning budget: the best of 10 random recipes (sampled configurations) is about 3 points (ConvNet) or 8 points (ResNet-18 with KD) below the best of 300. Ranked under one recipe each, as in a table that collects numbers from different papers, methods agree with the fully tuned leaderboard at a Kendall's τ (Kendall, 1938) of only 0.1, against 0.78 (ConvNet) and 0.58 (ResNet-18 with KD) with ten recipes shared by all methods (Appendix K). We therefore recommend that DD methods (i) report the best and the median accuracy over $k \geq 10$ recipes drawn with a fixed seed from a published search space and shared by all methods; (ii) select the reported recipe on a validation split of the real training data and state the tuning compute; (iii) state the training budget in processed images and report at the budget that the claim is about; (iv) report the accuracy under the best full-data recipe, which a practitioner obtains without re-tuning; and (v) report the full-data regret of the selected recipe, which is free on benchmark datasets because the shared recipes are part of the full-data sweep.

7 CONCLUSION

DataDistillBed evaluates dataset distillation beyond final accuracy by measuring whether distilled datasets preserve the model-selection decisions made on full data. Distilled datasets can provide useful ranking signals, but their consistency varies across datasets, IPC budgets, model families and search spaces, which makes it a necessary complement to accuracy when judging whether DD is useful for practical acceleration. Future work should scale this benchmark to broader tasks, incorporate adaptive hyperparameter search, and develop DD methods that optimize ranking consistency directly rather than treating it as a byproduct of accuracy.

AI USE STATEMENT

We did not use generative AI tools for any of the tasks with required disclosure under the ICLR 2027 AI Policy for Authors; in particular, the research ideas, methodology, experimental design, method implementations, data processing, and interpretation of results in this work are entirely the authors' own. We used generative AI assistants for tasks with recommended disclosure: editing the paper for readability, LaTeX formatting, creating and revising figure-plotting code, and formatting bibliography entries. In addition, LLM helped us in reviewing our code and give out suggestions in fixing bugs. We have reviewed all AI-assisted work; in particular, all edits were checked by the authors, and all bibliography entries were manually verified against their official bibliographic records. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL http://jmlr.org/papers/v13/bergstra12a.html.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10718–10727, June 2022.

Jeffrey A. Chan-Santiago and Mubarak Shah. Learnability-guided diffusion for dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41657–41666, June 2026.

Yanda Chen, Gongwei Chen, Miao Zhang, Weili Guan, and Liqiang Nie. Curriculum coarse-to-fine selection for high-ipc dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20437–20446, June 2025.

Zongxiong Chen, Jiahui Geng, Derui Zhu, Qing Li, Sonja Schimmler, and Manfred Hauswirth. Towards trustworthy dataset distillation: A benchmark of privacy, fairness and robustness. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2024. doi: 10.1109/IJCNN60899.2024.10650522.

Nick Craswell. Mean reciprocal rank. *Encyclopedia of Database Systems*, pp. 1703–1703, 2009.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020. doi: 10.1109/CVPRW50498.2020.00359.

Jiacheng Cui, Zhaoyi Li, Xiaochen Ma, Xinyue Bi, Yaxin Luo, and Zhiqiang Shen. Dataset distillation via committee voting, 2026. URL https://arxiv.org/abs/2501.07575.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 810–822. Curran Associates, Inc., 2022. doi: 10.52202/068431-0059. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/052e22cfdd344c79634f7ec76fa03e22-Paper-Datasets_and_Benchmarks.pdf.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to ImageNet-1K with constant memory. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6565–6590. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/cui23e.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Priyam Dey, Aditya Sahdev, Sunny Bhati, Konda Reddy Mopuri, and R. Venkatesh Babu. Rethinking dataset distillation: Hard truths about soft labels, 2026. URL https://arxiv.org/abs/2604.18811.

Mucong Ding, Yuancheng Xu, Tahseen Rabbani, Xiaoyu Liu, Brian Gravelle, Teresa Ranadive, Tai-Ching Tuan, and Furong Huang. Calibrated dataset condensation for faster hyperparameter search, 2024. URL https://arxiv.org/abs/2405.17535.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2185–2194, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1224. URL https://aclanthology.org/D19-1224/.

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3634–3646, 2022. doi: 10.1109/TPAMI.2021.3054824.

Timothy Dozat. Incorporating Nesterov momentum into Adam, 2016. URL https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ.

Jiawei Du, Yidi Jiang, Vincent Y. F. Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3749–3758, June 2023.

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo, 2022. URL https://arxiv.org/abs/2109.06716.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. URL https://arxiv.org/abs/1706.02677.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization, 2020. URL https://arxiv.org/abs/2007.01434.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching, 2024. URL https://arxiv.org/abs/2310.05773.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567, 2019. doi: 10.1109/CVPR.2019.00065.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from Imagenet, 2019. URL https://github.com/fastai/imagenette.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 240–248, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/jamieson16.html.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 06 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81. URL https://doi.org/10.1093/biomet/30.1-2.81.

M. G. Kendall and B. Babington Smith. The problem of m rankings. *The Annals of Mathematical Statistics*, 10(3):275–287, 1939.

Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11102–11118. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kim22c.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Stanford CS231N course project report, 2015. URL http://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL http://jmlr.org/papers/v18/16-558.html.

Zekai Li, Xinhao Zhong, Samir Khaki, Zhiyuan Liang, Yuhao Zhou, Mingjia Shi, Ziqiao Wang, Xuanlei Zhao, Wangbo Zhao, Ziheng Qin, Mengxuan Wu, Pengfei Zhou, Haonan Wang, David Junhao Zhang, Jia-Wei Liu, Shaobo Wang, Dai Liu, Linfeng Zhang, Guang Li, Kun Wang, Zheng Zhu, Zhiheng Ma, Joey Tianyi Zhou, Jiancheng Lv, Yaochu Jin, Peihao Wang, Kaipeng Zhang, Lingjuan Lyu, Yiran Huang, Zeynep Akata, Zhiwei Deng, Xindi Wu, George Cazenavette, Yuzhang Shang, Justin Cui, Jindong Gu, Qian Zheng, Hao Ye, Shuo Wang, Xiaobo Wang, Yan Yan, Angela Yao, Mike Zheng Shou, Tianlong Chen, Hakan Bilen, Baharan Mirzasoleiman, Manolis Kellis, Konstantinos N. Plataniotis, Zhangyang Wang, Bo Zhao, Yang You, and Kai Wang. Dd-ranking: Rethinking the evaluation of dataset distillation, 2025. URL https://arxiv.org/abs/2505.13300.

Dai Liu, Jindong Gu, Hu Cao, Carsten Trinitis, and Martin Schulz. Dataset distillation by automatic training trajectories, 2024. URL https://arxiv.org/abs/2407.14245.

Haoyang Liu, Yijiang Li, Tiancheng Xing, Peiran Wang, Vibhu Dalal, Luwei Li, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric, 2025. URL https://arxiv.org/abs/2311.18531.

Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1100–1113. Curran Associates, Inc., 2022a. doi: 10.52202/

068431-0081. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/07bc722f08f096e6ea7ee99349ff0a86-Paper-Conference.pdf.

Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17314–17324, October 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022b. doi: 10.1109/CVPR52688.2022.01167.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. URL https://arxiv.org/abs/1608.03983.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Brian B. Moser, Federico Raue, Sebastian Palacio, Stanislav Frolov, and Andreas Dengel. Unlocking dataset distillation with diffusion models, 2025. URL https://arxiv.org/abs/2403.03881.

Timothy Chieu Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *ICLR 2021*, 2021. URL https://openreview.net/forum?id=l-PrrQrK0QR.

Sebastian Pineda Arango, Hadi Jomaa, Martin Wistuba, and Josif Grabocka. Hpo-b: A large-scale reproducible benchmark for black-box hpo based on openml. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/ec8956637a99787bd197eacd77acce5e-Paper-round2.pdf.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046. URL https://doi.org/10.1137/0330046.

Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A label is worth a thousand images in dataset distillation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 131946–131971. Curran Associates, Inc., 2024. doi: 10.52202/079017-4194. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ee45939f4403e8abd63a15a29a9c055b-Paper-Conference.pdf.

Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16709–16718, June 2024a.

Shitong Shao, Zikai Zhou, Huanran Chen, and Zhiqiang Shen. Elucidating the design space of dataset condensation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 99161–99201. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3146. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b3ac2a18635fb355100c85f0f2044b89-Paper-Conference.pdf.

Donghyeok Shin, Seungjae Shin, and Il-chul Moon. Frequency domain-based dataset distillation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 70033–70044. Curran Associates, Inc., 2023. doi: 10.52202/075280-3069. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ddbbcd937d63d5c6b935c07b1a8222ec-Paper-Conference.pdf.

Donghyeok Shin, HeeSun Bae, Gyuwon Sim, Wanmo Kang, and Il-Chul Moon. Distilling dataset into neural field, 2025. URL https://arxiv.org/abs/2503.04835.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9390–9399, June 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Haoru Tan, Wang Wang, Sitong Wu, Xiuzhe Wu, Yang-Tian Sun, Chirui Chang, Shaofeng Zhang, and Xiaojuan Qi. Dataset distillation by influence matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19654–19664, June 2026.

Hong Ye Tan and Emma Slade. Dataset distillation as pushforward optimal quantization, 2026. URL https://arxiv.org/abs/2501.07681.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL https://arxiv.org/abs/1905.11946.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12186–12195, 2022. doi: 10.1109/CVPR52688.2022.01188.

Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25570–25580, June 2025.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2018. URL https://arxiv.org/abs/1811.10959.

Yifan Wu, Jiawei Du, Ping Liu, Yuewei Lin, Wei Xu, and Wenqing Cheng. Dd-robustbench: An adversarial robustness benchmark for dataset distillation. *IEEE Transactions on Image Processing*, 34:2052–2066, 2025. doi: 10.1109/TIP.2025.3553786.

Lingao Xiao and Yang He. Are large-scale soft labels necessary for large-scale dataset distillation? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 16406–16437. Curran Associates, Inc., 2024. doi: 10.52202/079017-0524. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1da9ca7e9cef4b1af63913f05d1630a4-Paper-Conference.pdf.

Eric Xue, Yijiang Li, Haoyang Liu, Peiran Wang, Yifan Shen, and Haohan Wang. Towards adversarially robust dataset distillation by curvature regularization, 2025. URL https://arxiv.org/abs/2403.10045.

William Yang, Ye Zhu, Zhiwei Deng, and Olga Russakovsky. What is dataset distillation learning?, 2024. URL https://arxiv.org/abs/2406.04284.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 73582–73603. Curran Associates, Inc., 2023. doi: 10.52202/075280-3219. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e91fb65c6324a984ea9ef39a5b84af04-Paper-Conference.pdf.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019. doi: 10.1109/ICCV.2019.00612.

Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy, 2024. URL https://arxiv.org/abs/2312.15927.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. URL https://arxiv.org/abs/1710.09412.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12674–12685. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhao21a.html.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6503–6512, 2023. doi: 10.1109/WACV56688.2023.00645.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl.

Xinhao Zhong, Shuoyang Sun, Xulin Gu, Chenyang Zhu, Bin Chen, and Yaowei Wang. Rectified decoupled dataset distillation: A closer look for fair and comprehensive evaluation, 2026. URL https://arxiv.org/abs/2509.19743.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017. URL https://arxiv.org/abs/1708.04896.

Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need, 2024a. URL https://arxiv.org/abs/2404.07976.

Zheng Zhou, Wenquan Feng, Shuchang Lyu, Guangliang Cheng, Xiaowei Huang, and Qi Zhao. Beard: Benchmarking the adversarial robustness for dataset distillation, 2024b. URL https://arxiv.org/abs/2411.09265.

Yawen Zou, Guang Li, Duo Su, Zi Wang, Jun Yu, and Chao Zhang. Dataset distillation via vision-language category prototype. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2941–2950, October 2025.

A Reproduction of Existing Dataset-Distillation Methods

This section documents our reproduction of prior dataset-distillation (DD) methods before they are used in the consistency benchmark. The goal of this stage is not to reproduce our full benchmark pipeline, but to obtain a faithful distilled training set for each baseline method and to check that training on that distilled set reaches a reasonable accuracy compared with the method's reported number. Only after this reproduction check do we use the distilled data as the proxy dataset in the ranking-consistency experiments.

A.1 Artifact Layout and Missing Large Files

The DataDistillBed repository separates the reproduction stage from the consistency evaluation stage. The top-level `readme.md` states that `DD_data` stores reproduced DD data from DC, DM, CAFE, TESLA, and LinBA-style ConvNet methods at IPC 1, 10, and 50, and that the reproduced accuracy is recorded during generation with the original method code. It also states that `DD_domainbed_results` stores the downstream DomainBed (Gulrajani & Lopez-Paz, 2020) evaluations produced from `DD_data`. The scripts under `tools/cifar10/` point to these external locations, e.g. `/home/mjc/DD_data/...` and `/home/mjc/DD_domainbed_results/...`.

The checked DataDistillBed artifact directory itself does not contain these large reproduced datasets: `consistent-dataset-distillation-2/distilled_data` is only a placeholder in the current artifact. Consequently, most raw reproduced datasets and full DomainBed result folders must be released separately or regenerated from the method-specific scripts. The local DataDistillBed folder does contain:

- method repositories or copied official code under `projects_pool/`;
- method-specific reproduction/evaluation configurations under `tools/configs/reproduce/`;
- DataDistillBed wrappers for running reproduced data through the unified evaluator under `tools/`;
- a small local ATT sanity check in `data/output/`; and
- ATT generation logs under `projects_pool/ATT/wandb/`.

This distinction is important: absence from `distilled_data` in the repository does not mean the method was not evaluated; it means the heavy reproduced data are not packaged in this folder.

A.2 General Reproduction Protocol

For each baseline, we follow the closest available official implementation. We prefer the authors' released distilled images when they are provided, because this removes unnecessary variation from the synthesis stage. If released images are unavailable but synthesis scripts are provided, we run the authors' distillation code with their recommended hyperparameters and record the evaluation accuracy produced during that run. If neither distilled data nor a faithful synthesis procedure is available for a dataset/IPC pair, that pair is excluded from the benchmark rather than approximated with a different setting.

The reproduced data are then converted into the unified DataDistillBed loader format. For tensor-based methods, this usually means storing `images_best.pt` and `labels_best.pt`, or method-specific image/label tensors. For image-folder methods with soft labels, such as CVDD, G-VBSM, SRe^{2}L, SCDD, and WMDD, we keep the image folders together with the FKD, mixup, cutmix, or soft label files expected by the original method. The corresponding JSON files in `tools/configs/reproduce/` specify how DataDistillBed loads and evaluates each method.

A.3 Reported vs. Reproduced Accuracy

Table 2 summarizes the reproduction sanity checks for CIFAR-10, CIFAR-100, and Tiny-ImageNet. Each cell is *reproduced accuracy / reported accuracy*, in percent. The reproduced value is the

Table 2: **Reported vs. reproduced DD accuracy.** Values are reproduced / reported Top-1 accuracy (%). Missing entries indicate that the local DataDistillBed artifact does not contain a corresponding comparison record.

Method	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	IPC 1	IPC 10	IPC 50	IPC 1	IPC 10	IPC 50	IPC 1	IPC 10	IPC 50
ATT	45.98 / 48.3	64.26 / 67.7	61.73 / 74.5	16.88 / 26.1	37.06 / 44.2	44.23 / 51.2	–	–	–
CVDD	26.11 / 27.4	46.28 / 54.7	70.58 / 76.9	18.95 / 16.5	57.66 / 61.8	64.17 / 69.4	15.15 / 20.1	49.36 / 53.0	55.61 / 61.2
DC	30.23 / 28.3	50.27 / 44.9	53.05 / 53.9	14.56 / 12.8	25.78 / 25.2	19.19 / –	–	–	–
DM	25.79 / 26.0	48.30 / 48.9	61.19 / 63.0	9.22 / 11.4	28.76 / 29.7	43.25 / 43.6	–	–	–
DSA	29.95 / 28.8	50.37 / 52.1	60.69 / 60.6	14.86 / 13.9	30.93 / 32.3	43.03 / 42.8	–	–	–
FreD	59.46 / 60.6	69.99 / 70.3	74.75 / 75.8	34.31 / 34.6	26.67 / 42.7	31.61 / 47.8	17.53 / 19.2	– / 24.2	– / 26.4
G-VBSM	18.71 / –	53.82 / 53.5	80.62 / 59.2	27.82 / 25.9	59.70 / 59.5	65.27 / 65.0	–	–	48.01 / 47.6
MTT	45.69 / 46.3	63.03 / 65.3	69.66 / 71.6	22.08 / 24.3	37.44 / 40.1	45.86 / 47.7	9.26 / 8.8	20.30 / 23.2	28.37 / 28.0
NCFM	47.91 / 49.5	69.68 / 71.8	76.88 / 77.4	30.68 / 34.4	49.01 / 48.7	54.15 / 54.7	14.05 / 18.2	23.63 / 26.8	25.74 / 29.6
SCDD	17.26 / –	40.05 / –	75.81 / –	12.63 / –	42.70 / –	46.01 / 53.4	–	– / 31.6	– / 45.9
SRe^{2}L	18.46 / –	– / –	77.75 / –	11.02 / –	45.00 / 23.48	49.44 / 51.35	–	–	40.96 / 41.1
TESLA	45.48 / 48.5	56.91 / 66.4	64.67 / 72.6	15.27 / 24.8	32.60 / 41.7	44.12 / 47.9	4.55 / –	14.11 / –	26.35 / –
WMDD	24.40 / –	41.02 / –	78.06 / –	14.64 / –	45.74 / –	48.13 / –	–	–	44.25 / –

accuracy obtained in our reproduction or our closest DataDistillBed evaluation setting; the reported value is the number from the corresponding original method paper or released README. A dash means that either the dataset/IPC pair was not evaluated by the original method in our benchmark scope, or we did not find the corresponding reproduced/result record in the local artifact.

Several entries show substantial gaps between the reproduced value and the reported number. We keep these values in the table rather than filtering them out, because the reproduced data used by DataDistillBed must be auditable even when it does not exactly match the original report. In particular, some large-scale methods rely on released image folders, soft labels, teacher ensembles, or FKD metadata that are not included in the local artifact; for those cases, the table should be updated once the external `DD_data` and `DD_domainbed_results` folders are attached.

A.4 ConvNet-Based Reproductions

DC, DM, and DSA-style baselines. For the classical ConvNet baselines, we use the DatasetCondensation-style entry points in DataDistillBed: `main.py` for gradient-matching style methods and `main_DM.py` for distribution matching. These scripts generate synthetic tensors and report accuracy during generation. Example commands are:

```
python main.py --dataset CIFAR10 --model ConvNet --ipc 10 \
  --save_path /path/to/DD_data/CIFAR10_10_DC \
  --data_path /path/to/data --num_eval 3

python main_DM.py --dataset CIFAR100 --model ConvNet --ipc 50 \
  --dsa_strategy color_crop_cutout_flip_scale_rotate --init real \
  --lr_img 1 --num_exp 5 --num_eval 3 \
  --save_path /path/to/DD_data/DM_cifar100_ipc50 \
  --data_path /path/to/data
```

The DataDistillBed scripts under `tools/cifar10/` show how those reproduced datasets are later evaluated. For example, `run_DCipc1.sh`, `run_DCipc10.sh`, and `run_DCipc50.sh` expect `DD_data/CIFAR10_1_DC`, `DD_data/CIFAR10_10_DC`, and `DD_data/CIFAR10_50_DC`. Similarly, `run_DM.sh`, `run_DM10.sh`, and `run_DM50.sh` expect DM reproduced data at IPC 1, 10, and 50.

MTT and TESLA. For trajectory-matching methods, we follow the official two-stage workflow: first generate expert trajectories on the original dataset, then distill synthetic images by matching student training trajectories to the experts. The copied MTT README recommends 100 ConvNet experts trained for 50 epochs, e.g.

```
python buffer.py --dataset=CIFAR100 --model=ConvNet \
  --train_epochs=50 --num_experts=100 --zca \
```

```
--buffer_path=/path/to/buffer --data_path=/path/to/data
```

The TESLA README follows the same structure but uses TESLA's constant-memory distillation. Its example CIFAR commands include IPC 1 on CIFAR-10 and IPC 50 on CIFAR-100. DataDistillBed contains downstream wrappers such as `tools/cifar10/run_CAFEipc*.sh` for TESLA-style reproduced CIFAR-10 data paths named `DD_data/tesla_cifar10_ipc1`, `DD_data/tesla_cifar10_ipc10`, and `DD_data/tesla_cifar10_ipc50`. These wrappers should be interpreted as the consistency-evaluation stage over already reproduced TESLA data, not as the TESLA synthesis command itself.

ATT. ATT is the one baseline for which DataDistillBed contains local generation logs. The ATT README gives the exact commands used for CIFAR-10 and CIFAR-100. The local wandb summaries show completed CIFAR-10 generation runs:

Condition	Iterations	Accuracy	EMA accuracy
CIFAR-10, IPC 1	6000	45.25	45.75
CIFAR-10, IPC 10	60000	65.15	66.08
CIFAR-10, IPC 50	partial/local logs	47.12–47.86	48.06–49.05

These numbers are generation-time ConvNet evaluations from ATT's original logging. The IPC 1 reproduced dataset was also passed once through the DataDistillBed DomainBed wrapper, yielding `env1_in_acc = 45.98%` after 1000 epochs in `data/output/672fa1f16872ae7a02f369cd4213cd77/results.jsonl`. A second local DataDistillBed wrapper job for ATT CIFAR-10 IPC 10 failed because `./distilled_data/att/dataset/CIFAR10_ipc10/images_best.pt` was not present in the packaged folder; this is consistent with the large reproduced data being stored outside DataDistillBed.

FreD and NCFM. For FreD and NCFM, we use the authors' released code and/or released condensed tensors rather than reimplementing the synthesis objective. The FreD README reports ConvNet/TM accuracies for CIFAR-10, CIFAR-100, Tiny-ImageNet, and related low-resolution settings; the CIFAR and Tiny-ImageNet reported values are used as the target sanity range when checking our reproduced or downloaded data. NCFM provides both training scripts and downloadable condensed datasets. In DataDistillBed, NCFM evaluation is configured by files such as `NCFM_cifar10.json`, `NCFM_cifar100.json`, and `NCFM_TinyImageNet200.json`. The loader decodes NCFM tensors before passing them to the unified evaluator, because NCFM's effective IPC is a storage or compressed-domain budget rather than a raw image count identical to DC/DM.

A.5 ResNet/Soft-Label Reproductions

SRe^{2}L and SCDD. For SRe^{2}L-family methods, we follow the squeeze–recover–relabel–validate pipeline described in the copied SRe^{2}L and SCDD README files. For CIFAR-100, the SRe^{2}L README reports a 200-epoch ResNet-18 squeeze model with 78.72% top-1 accuracy before recovery. For TinyImageNet200, the README provides download links for released IPC 50 and IPC 100 distilled images. The DataDistillBed wrapper `tools/tiny/sre2l.sh` evaluates the released TinyImageNet200 SRe^{2}L IPC 50 data with knowledge distillation using `tools/configs/low_res_r18_sre2l-default.json`.

SCDD follows the same broad structure but uses self-supervised compression before recovery. Its README documents CIFAR-100 recovery and relabel commands. In our benchmark, SCDD is included where compatible reproduced data and relabel artifacts are available; otherwise it is omitted rather than approximated from SRe^{2}L.

CVDD and G-VBSM. CVDD and G-VBSM provide released distilled images and soft-label/relabel artifacts. We therefore treat them primarily as data-ingestion reproductions: download the authors' released images, organize them in the expected folder layout, and keep the FKD paths aligned with the images. DataDistillBed configuration files such as `CVDD_cifar10_ipc10.json`,

`CVDD_TinyImageNet200_ipc50.json`, and `GVBSM_cifar100_ipc50.json` store the corresponding `fkd_path` and gradient-accumulation settings. The G-VBSM README reports strong official targets such as 38.7% on CIFAR-100 IPC 10 and 47.6% on TinyImageNet IPC 50, which serve as sanity checks for the downloaded and relabeled data.

WMDD. WMDD is evaluated as a ResNet/knowledge-distillation method. Its DataDistillBed configuration files under `tools/configs/tricks/` and `tools/configs/reproduce/` force a ResNet-family student and knowledge-distillation evaluation. As with the other large-scale soft-label methods, the reproduced data are expected outside the packaged `distilled_data` placeholder.

A.6 From Reproduced Data to Consistency Evaluation

After a baseline's distilled set passes the reproduction sanity check, DataDistillBed evaluates it with the same downstream protocol as the original dataset. The scripts in `tools/cifar10/`, `tools/tiny/`, and `tools/imagenette/` call `domainbed.scripts.sweep` with matched `hparams_seed` and `trial_seed`. The outputs are stored in `DD_domainbed_results` in the original experiment setup. The helper `tools/collect2.sh` then compares an original-data result directory with a reproduced-distilled-data result directory to compute accuracy and rank correlation.

Thus the reproduction stage has two checks. First, the method's own code should produce a distilled set whose training accuracy is close to the method's reported result. Second, the DataDistillBed wrapper should be able to train downstream models on that reproduced set without changing the method-specific data format or soft-label assumptions. Only the second output is used for the consistency metrics in the main paper.

B Evaluation Metrics

We assess consistency by comparing the accuracy rankings induced by the distilled and original datasets with four complementary metrics that target different questions: global agreement and the utility of a short actionable list. For all top-k metrics we report $k = 10$, which provides a practical shortlist size while yielding stable estimates across seeds.

Spearman's rank correlation. Our primary metric (Spearman, 1904) measures whether the distilled ranking preserves the pairwise order of hyperparameters from the original ranking. It is invariant to monotone transformations of accuracy and ranges in $[-1, 1]$ (1: perfect agreement; 0: no monotonic relation; -1: perfect inverse). Hence, it reflects global ranking consistency across the entire search space.

NDCG@k. Normalized Discounted Cumulative Gain (Järvelin & Kekäläinen, 2002) treats the original-dataset accuracies as relevance scores and evaluates how well the distilled top-k orders the most promising configurations:

$$\mathrm{NDCG@}k = \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k}, \quad \mathrm{DCG@}k = \sum_{i=1}^{k} \frac{\mathrm{rel}_i}{\log_2(i+1)}, \tag{2}$$

where rel_i is the original accuracy of the hyperparameter ranked at position i by the distilled dataset. High NDCG@k means the distilled list places truly strong configurations near the top (larger discount for lower ranks).

Overlap@k. The set-level hit rate $|\mathrm{Top}_k^{\mathrm{distilled}} \cap \mathrm{Top}_k^{\mathrm{original}}|/k$ measures how many of the truly best k configurations are recovered, ignoring their internal order.

MRR@k. Mean Reciprocal Rank (Craswell, 2009) of the distilled top-k under the original ranking,

$$\mathrm{MRR@}k = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{r_i}, \tag{3}$$

where r_i is the position in the original ranking of the i-th distilled top-k item. Larger values indicate that the distilled picks tend to appear very early in the true ranking; since the k picks are distinct, the

maximum is H_k/k with H_k the k-th harmonic number (0.29 for $k = 10$), reached when the distilled top-k coincides with the original top-k.

Collectively, these metrics capture global ranking agreement, precision at top positions, and quality of individual top choices.

C TRICKS HYPER-PARAMETER SEARCH SPACE

The tricks search space in DataDistillBed is designed to cover training choices that are commonly tuned in practical image-classification pipelines while still remaining fully bounded and auditable. Table 3 gives the hyper-parameter search space used in our experiments. Continuous variables are sampled from bounded uniform or log-uniform distributions; categorical variables are sampled from explicit finite sets.

The categorical options are standard techniques from the image-classification literature. The optimizers are SGD, averaged SGD (Polyak & Juditsky, 1992), Adam (Kingma & Ba, 2017), AdamW (Loshchilov & Hutter, 2019) and NAdam (Dozat, 2016), all taken from the shared implementation; `cosine` is the cosine annealing schedule of Loshchilov & Hutter (2017), applied without restarts, and the linear warm-up follows the gradual scheme of Goyal et al. (2018). The augmentation policies are DSA (Zhao & Bilen, 2021), crop-and-flip, AutoAugment (Cubuk et al., 2019), and RandAugment (Cubuk et al., 2020) combined with random erasing (Zhong et al., 2017); the batch-level mixing options are mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019); and label smoothing follows Szegedy et al. (2016).

Table 3: **Tricks hyper-parameter search space in DataDistillBed.** Defaults are used when `hparams_seed=0`; random values are generated deterministically from `hparams_seed` and `trial_seed`. Log-uniform notation means sampling uniformly in the exponent.

Hyper-parameter	Default	Search domain	Role in training
`lr`	5×10^{-3}	5×10^{u}, $u \sim \mathcal{U}[-5, -1]$; range $[5 \times 10^{-5}, 5 \times 10^{-1}]$	Base learning rate. Under the constant warm-up variant the run trains at 0.01 of this value, so the effective range is $[5 \times 10^{-7}, 5 \times 10^{-1}]$.
`optim`	`adamw`	{`adam`, `sgd`, `asgd`, `adamw`, `nadam`}	Optimizer family.
`batch_size`	256	$\lfloor 2^{u} \rfloor$, $u \sim \mathcal{U}[7, 11]$; range 128–2048	Mini-batch size.
`scheduler`	`cosine`	{`steplr`, `cosine`, `polynomiallr`, `linearlr`, `constant`}	Learning-rate decay policy, defined over the full training length.
`use_warm_up`	`True`	{`True`, `False`}	Enables a warm-up phase; the decay policy then starts after warm-up and keeps the full training length as its horizon.
`warm_up_ratio`	0.05	$\mathcal{U}[0.01, 0.20]$	Fraction of training assigned to warm-up.
`lr_warmup_method`	`linear`	{`linear`, `constant`}	`linear`: the learning rate rises linearly from 0.01 towards the base value during warm-up; `constant`: the learning rate is scaled by 0.01 for the entire run.
`weight_decay`	10^{-4}	10^{u}, $u \sim \mathcal{U}[-6, -2]$; range $[10^{-6}, 10^{-2}]$	Weight-decay coefficient; with `asgd` it takes effect only together with `no_bias_decay`.
`no_bias_decay`	`True`	{`True`, `False`}	Excludes bias and normalization parameters from weight decay.
`KD_temperature`	5	integers in {1,…,20}	Temperature for distillation soft targets.
`augment_type`	`none`	{`none`, `dsa`, `CropFlip`, `AutoAug`, `RandAugErasing`}	Image augmentation policy.
`mix_type`	`None`	{`mixup`, `cutmix`, `mixup_or_cutmix`, `None`}	Batch-level mixing augmentation.
`use_label_smoothing`	`True`	{`True`, `False`}	Enables label smoothing.
`label_smoothing`	0.2	$\mathcal{U}[0.1, 0.5]$	Smoothing coefficient when enabled.
`use_amp`	`False`	{`True`, `False`}	Mixed-precision training switch.

For a fair comparison, all algorithms in the same method class and dataset share the same sampled hyper-parameter space. Method-specific JSON configs only fix compatibility constraints such as whether a baseline belongs to the ConvNet or ResNet/teacher-distillation group; they do not give one DD method a larger tuning space than another method in the same group. Unless otherwise stated, we sample 100 `hparams_seed` values per condition. The ImageNet-1K full-data runs always used mixed precision, which the distilled runs sample; Table 10 excludes the four pairs whose full-data

run diverged with it while the distilled run, trained without it, did not (the default configuration in all three trial seeds and one other).

Why this search space is bounded. A possible concern is that optimizer and learning-rate choices are "not well limited". We address this by bounding every continuous knob and enumerating every categorical knob. The base learning rate spans four orders of magnitude, from 5×10^{-5} to 5×10^{-1}, which is deliberately broad enough to include conservative AdamW-style settings and aggressive SGD-style settings; the constant warm-up variant extends the effective range down to 5×10^{-7}, so part of the space trains too slowly to be useful at our budgets, as in a realistic search. Similarly, the optimizer set is not open-ended: it contains five widely used first-order optimizers available in the shared implementation. The purpose of DataDistillBed is not to find the globally best optimizer ever proposed, but to test whether a distilled dataset preserves the ranking of realistic training configurations under a controlled, identical space.

Design choices. Q: Does the wide learning-rate interval make the benchmark noisy? A: The interval is intentionally log-uniform because learning rates act multiplicatively. Collapsed runs are part of realistic HP search; on the higher-resolution datasets, where they are most frequent, we additionally report ρ_{v} over the configurations that train on the full data (Table 10), to avoid inflated agreement from configurations that fail on both original and distilled data.

Q: Learning rate, optimizer, and batch size are strongly coupled. Why does DataDistillBed define them as separate search dimensions? A: We agree that these hyperparameters interact. In practice, however, no one knows in advance which learning-rate, optimizer, and batch-size combination will be optimal for a given dataset, architecture, and distilled-data method. That uncertainty is exactly why this remains a hyperparameter search problem. DataDistillBed **does not assume** that learning rate, optimizer, and batch size act independently in the final training outcome; it only parameterizes the search space along interpretable axes so that configurations can be sampled, evaluated, and ranked. The resulting sampled point is still a joint configuration, e.g., a specific optimizer with a specific learning rate and batch size. Our consistency metric then asks whether the distilled dataset preserves the ranking of these joint configurations compared with the original dataset.

Q: Could a method look bad only because its favorite optimizer is missing? A: This risk is reduced by using the same optimizer set for all methods in the same class. Adding method-specific optimizers would confound proxy quality with extra tuning privilege. Our question is whether the distilled data ranks a common engineering search space consistently with the original data.

Q: Why include Adam variants and SGD in the same space? A: The decomposition in Figure 9 shows that `optim`, alone and in combination with the learning rate, is a dominant factor of the ranking. Excluding optimizer would hide an important source of practical variation; including a finite optimizer set lets us measure whether distilled datasets preserve that sensitivity.

Q: Are knowledge distillation and model family extra advantages for ResNet-based methods? A: They are treated as method-class constraints rather than per-method bonuses. ConvNet-based methods are evaluated with ConvNet-style training, while ResNet-based methods are evaluated with ResNet-18 and teacher supervision because those methods depend on teacher-provided labels or features. Comparisons are therefore made within matched classes and datasets.

D Budget-Matched Comparison: Protocol, Numbers and Checks

Budgets. Figure 3 compares distilled data and the full dataset at equal numbers of processed training images. A budget of b full-data epochs corresponds to $b \times N$ images, where N is the size of the full training set (50,000 for CIFAR-10 and CIFAR-100, 9,469 for ImageNette), so a distilled set of n images is trained for bN/n epochs. Figures 3 and 6 show this budget as $b\%$ of a 100-epoch training run on the full dataset; the tables keep full-data epochs. For FreD and NCFM, n is the number of decoded images (for example 2,000 on CIFAR-10 at IPC 50), so their position on the axis is their real cost. Budgets follow a 1-2-5 grid from 0.5 (distilled data only) to 200 full-data epochs. Every budget is a separate run from scratch with a learning-rate schedule set for that budget; no point is a checkpoint of a longer run. Figure 6 shows the same runs by method, and Tables 5 to 7 give the numbers.

Configurations. Each point reports the best final test accuracy over 10 hyperparameter configurations that were fixed before the runs. On CIFAR they were taken from our earlier sweeps of 300 configurations per setting: for a distilled set, the five best at the longest swept budget and the five best at the shortest, and, beyond the longest swept budget, the ten best of the longest sweep; for the full dataset, the two best at each of the seven swept budgets (1 to 100 epochs). On ImageNette, where no such sweeps exist, 40 to 44 candidate configurations per curve (the most frequent ones in the CIFAR lists) were run at 5 and 20 full-data epochs, and the lists were chosen from those runs in the same way. The statistic is an oracle, since the maximum is taken on the test set, and it is applied identically to distilled and full data.

What the axis does not count. The axis counts training images, not compute. The ResNet-based methods train their distilled sets with online knowledge distillation from the method's teacher, which adds one teacher forward pass per batch (24 to 43 percent more time per image in our measurements); the full dataset and the random subset are trained without a teacher. The cost of producing the distilled set and of training the teacher is not counted either. All three omissions favor the distilled sets. For SRe2L, SCDD and WMDD on CIFAR the original teacher checkpoint was no longer available; we use a public ResNet-18 teacher trained with the same recipe (95.0 and 78.4 percent test accuracy on CIFAR-10 and CIFAR-100) and, for SRe2L, images regenerated with the released recovery script.

Official budgets. Table 4 places the evaluation protocol of each method on the axis of Figure 3 (the triangles there), at the IPCs shown. The number of epochs comes from the paper where it states it (SRe^{2}L, SC-DD, CV-DD and G-VBSM) and otherwise from the default settings of the released code, which we checked at a fixed commit (the shared DatasetCondensation code for DC, DSA and DM, and the repository of each other method). Of the 36 settings, 32 lie beyond the last budget at which a distilled set of the same family beats the full dataset, and 11 amount to at least one 100-epoch run: eight to exactly one (1,000 epochs on a set a tenth the size of the training set) and three to more than two (FreD on CIFAR-100 and Tiny-ImageNet, NCFM on CIFAR-100). The exceptions are DC and CV-DD on CIFAR-10, at 3% against a crossover at 3.4%, and CV-DD and WMDD on ImageNette, at 3.2%, where the distilled sets trained with a teacher lead up to 42%. The ResNet-based methods also train their teachers on the full dataset, which none of these budgets counts; CV-DD, for example, trains a committee of five models on CIFAR-100 for 200 epochs each.

Checks. Table 8 compares the new runs with the earlier sweeps. Runs whose configuration and budget also exist in the earlier sweeps reproduce them to -0.21 points on average when teacher and data are unchanged, and the curves with the replacement teacher show no systematic shift ($+0.06$ and -0.19 points). At the budgets covered by the earlier sweeps, the best of the 10 fixed configurations matches the best of all 300 to $+0.01$ points on average. Table 9 probes budgets far beyond the earlier sweeps: on CIFAR-10 with ResNet-18, none of the next 20 configurations beats the fixed list by more than 0.2 points at 200 epochs. Of the 3,996 runs of the grid, 3 diverged and 3 crashed (a step schedule whose step size is zero at budgets of one or two epochs); the affected points are the best of the remaining configurations.

Table 4: **Official evaluation budgets at the IPCs of Figure 3.** Epochs of training on the distilled set in each method's own protocol and, in parentheses, the same budget in percent of a 100-epoch run on the full dataset, the unit of Figure 3 (decoded images counted for FreD and NCFM). Source: P, stated in the paper; C, only in the released code. –: setting not reported by the method. Last row: the budget up to which the best distilled set of each family beats the full dataset in Figure 3 (ConvNet-based / ResNet-based).

Method	Source	CIFAR-10 IPC 50	CIFAR-100 IPC 50	ImageNette IPC 10	Tiny-ImageNet IPC 50
ConvNet-based methods					
DC	C	300 (3.0%)	–	–	–
DSA	C	1000 (10%)	1000 (100%)[a]	–	–
DM	C	1000 (10%)	1000 (100%)	–	1000 (100%)
MTT	C	1000 (10%)	1000 (100%)	1000 (11%)	1000 (100%)
TESLA	C	1000 (10%)	1000 (100%)	–	–
ATT	C	1000 (10%)	1000 (100%)	1000 (11%)	–
FreD	C	1000 (40%)	1000 (256%)	1000 (42%)	1000 (320%)[b]
NCFM	C	2000 (80%)	2000 (800%)	2000 (84%)	40 (16%)[c]
ResNet-based methods					
SRe^{2}L	P	–	400 (40%)	–	100 (10%)
SC-DD	P	–	200 (20%)	–	500 (50%)
CV-DD	P	300 (3.0%)[d]	300 (30%)[d]	300 (3.2%)	300 (30%)
G-VBSM	P	1000 (10%)	1000 (100%)	–	100 (10%)
WMDD	C	–	–	300 (3.2%)[e]	300 (30%)[e]
A distilled set wins up to		3.4% / 3.4%	11.6% / never	5.5% / 42%	4.4% / 7.5%

[a]DSA does not report CIFAR-100 at IPC 50; the number is from the DM paper (same authors), which ran the released DSA code. [b]Decoding factor 3.2 as stated in the paper; its hyperparameter table and code imply a different one. [c]The released configuration, committed after the paper. [d]CV-DD's main table; its long-training rows on CIFAR use 1000 epochs. [e]The paper keeps the settings of SRe^{2}L; the released code trains for 300 epochs.

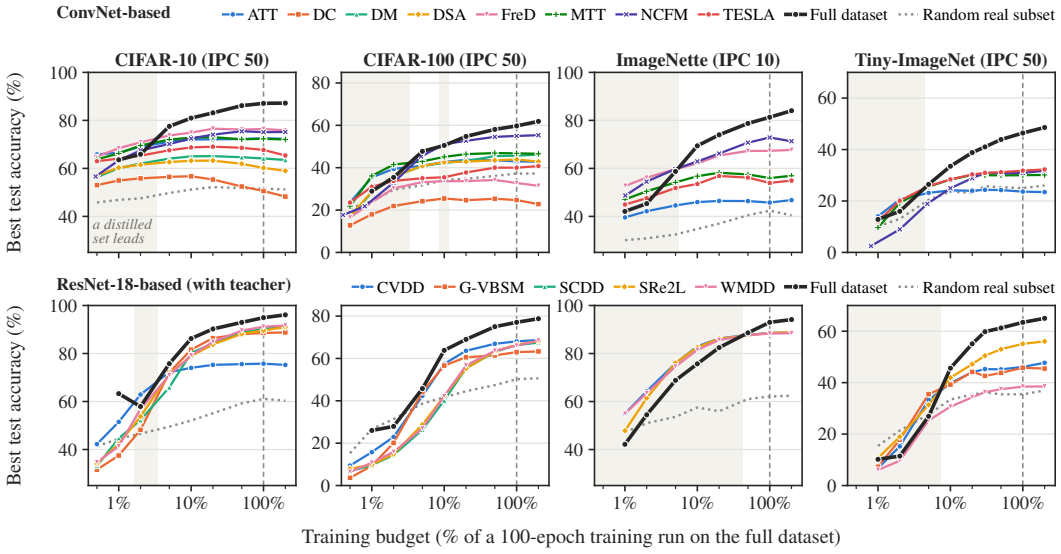

Figure 6: **The budget-matched comparison by method.** Same runs and axes as Figure 3, one panel per dataset and model family, every method with its own color and marker. Shading marks the budgets at which some distilled set beats the full dataset; the dashed line marks 100%, one 100-epoch run on the full dataset. The distilled sets of the ResNet-based methods are trained with the method's teacher, which the axis does not count. The numbers are in Tables 5 to 7.

Table 5: **Numbers behind the budget-matched figure, CIFAR-10 (IPC 50).** Best final test accuracy (%) over the 10 fixed configurations of each curve; every budget is a separate run from scratch. Budgets are in full-data epochs (one epoch = 50,000 training images). Bold: the best distilled set of the family at that budget. ResNet-family distilled sets train with their method's teacher; full data and the random subset do not.

Training budget (full-data epochs)	0.5	1	2	5	10	20	50	100	200
ConvNet									
Full dataset	–	63.6	65.9	77.5	81.0	83.1	86.1	87.1	87.2
Random real subset	45.8	46.9	47.5	49.8	51.2	52.2	51.8	51.5	51.3
ATT	**65.8**	66.5	69.0	71.1	72.0	72.1	72.2	72.6	72.3
DC	53.0	55.0	55.8	56.5	56.7	55.4	52.5	50.6	48.2
DM	56.3	60.1	62.0	64.1	65.0	65.1	64.6	64.1	63.5
DSA	57.0	60.3	61.5	62.6	63.2	63.2	61.9	60.2	59.0
FreD	64.7	**68.4**	**70.9**	**73.7**	**74.9**	**76.5**	**76.2**	**76.4**	**75.9**
MTT	63.8	66.2	69.6	72.0	72.8	73.0	72.1	72.3	72.0
NCFM	56.6	63.3	67.5	70.2	72.5	74.1	75.5	75.1	75.2
TESLA	63.0	64.1	65.3	67.5	68.7	69.0	68.6	67.7	65.4
ResNet-18 (low resolution)									
Full dataset	–	63.3	57.9	75.8	86.2	90.2	93.0	94.9	96.1
Random real subset	41.3	44.5	46.6	49.5	52.1	55.1	59.2	61.1	60.4
CVDD	**42.2**	**51.5**	**62.9**	**72.3**	74.0	75.3	75.6	75.8	75.2
G-VBSM	31.6	37.5	48.2	72.1	**81.6**	**86.4**	88.1	88.6	88.8
SCDD	33.2	44.7	52.3	65.9	80.2	84.2	88.9	90.2	91.3
SRe2L	34.2	42.3	53.8	71.4	79.0	83.7	88.2	89.6	91.0
WMDD	34.7	41.3	56.9	70.9	79.2	84.9	**89.6**	**91.2**	**91.7**

Table 6: **Numbers behind the budget-matched figure, CIFAR-100 (IPC 50).** Best final test accuracy (%) over the 10 fixed configurations of each curve; every budget is a separate run from scratch. Budgets are in full-data epochs (one epoch = 50,000 training images). Bold: the best distilled set of the family at that budget. ResNet-family distilled sets train with their method's teacher; full data and the random subset do not.

Training budget (full-data epochs)	0.5	1	2	5	10	20	50	100	200
ConvNet									
Full dataset	–	28.9	35.4	47.7	50.5	54.8	58.1	59.7	61.9
Random real subset	20.9	24.2	29.6	31.7	34.2	34.8	36.1	37.2	37.2
ATT	**23.6**	35.4	39.4	40.9	42.7	43.7	43.5	43.0	42.8
DC	12.8	17.9	21.9	24.2	25.5	24.6	25.3	24.7	22.8
DM	16.7	28.4	35.9	40.8	42.3	43.1	45.5	45.7	46.2
DSA	17.1	28.7	36.0	40.8	42.6	42.9	43.5	43.9	42.9
FreD	16.5	23.7	30.2	33.2	33.7	33.7	34.2	32.8	31.4
MTT	21.8	**36.1**	**41.5**	43.0	45.1	46.4	46.9	46.8	46.6
NCFM	17.7	21.8	33.0	**45.3**	**51.1**	**52.7**	**54.5**	**55.0**	**55.3**
TESLA	23.6	31.0	33.8	35.0	35.5	37.7	40.0	40.0	40.7
ResNet-18 (low resolution)									
Full dataset	–	26.0	27.9	45.8	63.8	69.0	75.0	77.1	78.8
Random real subset	15.3	26.7	31.4	38.7	41.7	44.5	47.4	50.2	50.6
CVDD	**9.4**	**15.8**	**23.0**	42.4	**57.4**	**63.6**	**66.9**	**68.1**	**68.7**
G-VBSM	3.7	9.3	20.1	**44.1**	56.6	60.6	61.4	63.0	63.3
SCDD	6.8	9.6	14.4	26.4	40.2	55.3	63.1	66.3	67.5
SRe2L	7.9	9.9	14.9	28.8	42.2	55.2	63.4	66.5	68.2
WMDD	6.4	10.8	15.9	26.9	42.2	56.7	63.6	66.1	68.4

Table 7: **Numbers behind the budget-matched figure, ImageNette (IPC 10).** Best final test accuracy (%) over the 10 fixed configurations of each curve; every budget is a separate run from scratch. Budgets are in full-data epochs (one epoch = 9,469 training images). Bold: the best distilled set of the family at that budget. ResNet-family distilled sets train with their method's teacher; full data and the random subset do not.

Training budget (full-data epochs)	0.5	1	2	5	10	20	50	100	200
ConvNetD5 (128 px)									
Full dataset	–	42.1	45.3	58.7	69.4	74.0	78.7	81.3	84.0
Random real subset	–	30.1	30.9	32.5	34.8	37.0	40.5	42.4	40.5
ATT	–	39.6	42.2	44.6	46.0	46.5	46.4	45.8	46.8
FreD	–	**52.7**	**56.1**	59.7	62.1	65.4	67.3	67.3	67.7
MTT	–	47.1	50.4	54.2	56.8	58.2	57.5	56.0	56.9
NCFM	–	48.7	54.5	**59.7**	**62.9**	**66.2**	**70.7**	**72.9**	**71.3**
TESLA	–	44.9	47.6	51.8	53.5	56.8	56.2	54.0	54.9
ResNet-18 (224 px)									
Full dataset	–	42.2	54.5	68.8	75.7	82.5	88.6	93.0	94.1
Random real subset	–	47.2	51.0	53.6	57.5	56.0	61.0	62.1	62.4
CVDD	–	**55.0**	**64.2**	75.9	**83.0**	**86.2**	**87.8**	88.6	88.7
SRe2L	–	47.9	61.4	**76.0**	82.6	86.2	87.5	**88.6**	**88.8**
WMDD	–	54.9	63.5	74.6	81.5	85.7	87.6	88.3	88.5

Table 8: **Checks of the budget-matched protocol on CIFAR.** Left: every new run whose configuration and budget also exist in the earlier 300-configuration sweeps, new minus old final accuracy in points. Right: at the budgets that the earlier sweeps cover, the best of the 10 fixed configurations (new runs) minus the best of all 300 (old runs). 'New teacher': the public replacement teacher for SCDD, SRe2L and WMDD on CIFAR; 'new images': the regenerated SRe2L sets.

	Same configuration, new − old				Best of 10 (new) − best of 300 (old)			
Group	pairs	mean	s.e.	$\leq$1 pt	points	mean	median	range
Same teacher, same data	1,005	-0.21	0.07	79%	101	$+0.01$	$+0.05$	-2.6 to $+2.1$
New teacher (SCDD, WMDD)	120	$+0.06$	0.24	40%	12	-0.33	-0.28	-4.0 to $+5.2$
New teacher, new images (SRe2L)	60	-0.19	0.33	27%	6	-0.14	$+0.23$	-2.2 to $+1.5$

Table 9: **Long-budget probe on CIFAR-10 (ResNet-18 with teacher).** The fixed list of a curve holds the 10 best configurations of the longest earlier sweep (4 full-data epochs). The probe runs the next 20 (ranks 11 to 30) at 50 and at 200 full-data epochs, that is, 12 and 50 times past that sweep. Accuracy in %. Across all 85 long-budget points of the CIFAR grid, the winner came 73 times from ranks 1 to 5 of the earlier sweep, 7 times from ranks 6 and 7, and 5 times from ranks 8 to 10.

Method	Budget	Best of the list (10)	Best outside (20)	Outside − list
CVDD	50	75.58	75.45	-0.13
CVDD	200	75.25	75.43	$+0.18$
WMDD	50	89.57	88.13	-1.44
WMDD	200	91.66	91.61	-0.05

E TRICKS CONSISTENCY ACROSS IPCS AND AT HIGHER RESOLUTION

Figure 7 shows the tricks consistency of Section 4.2.1 at every IPC, and Table 10 reports it on ImageNette and ImageNet-1K.

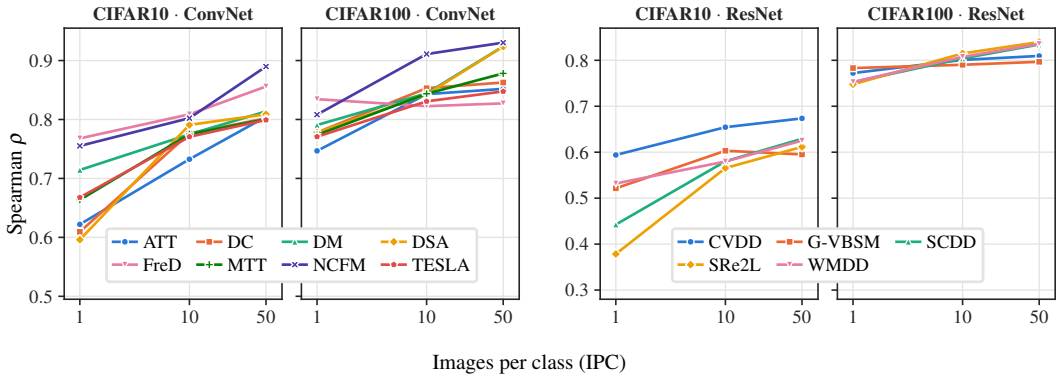

Figure 7: **Trend of tricks consistency across IPCs**. Spearman's ρ in the four dataset and family settings, averaged over the training lengths shared by the methods of each panel (std per point at most 0.09, median 0.03). Consistency generally rises with IPC.

Table 10: **Tricks consistency on ImageNette and ImageNet-1K.** ρ_v: ρ over the configurations that train on full data. ImageNette: older search space. ImageNet-1K: 300 epochs, 56 configurations of three seeds ranked together. Bold: best per block.

	Spearman		Top 10		
Method	ρ	ρ_v	NDCG	Ovl.	MRR
ImageNette, ConvNet, IPC 1, 1000 epochs					
ATT	0.719	0.287	0.875	0.20	0.067
MTT	0.718	0.280	0.833	0.17	0.053
NCFM	**0.755**	**0.387**	**0.938**	**0.27**	**0.076**
TESLA	0.693	0.218	0.759	0.07	0.039
ImageNette, ConvNet, IPC 10, 500 epochs					
ATT	0.800	0.499	0.945	0.27	0.082
MTT	0.840	0.613	0.942	**0.30**	0.113
NCFM	**0.866**	**0.653**	**0.963**	**0.30**	**0.119**
TESLA	0.816	0.548	0.912	0.23	0.072
Chance	0	0	0.608	0.10	0.052
ImageNet-1K, ResNet-18 + KD, IPC 50					
CVDD	**0.675**	0.771	0.922	0.70	0.254
G-VBSM	0.658	0.780	0.921	0.70	0.254
SRe^{2}L	0.649	0.776	0.953	0.70	**0.255**
WMDD	0.673	**0.786**	**0.959**	0.70	**0.255**
Chance	0	0	0.528	0.18	0.082

F NATS-BENCH RESULTS

Table 11 gives the NATS-Bench consistency of Section 4.2.2, and Table 12 extends it for every method and IPC. In the NATS-Bench sweep, each sampled configuration fixes one topology together with two training switches, DSA augmentation (on in 48 percent of the configurations) and mixup

(on in 33 percent), drawn independently of the architecture and applied identically on the distilled and the original data. We draw 100 configurations per trial seed and train the distilled sets for 400 epochs at IPC 1 and 200 epochs at IPC 10 and 50, and the original data for 30 epochs. DSA raises the accuracy of a run on distilled data by 4 points in the median setting (up to 16) and mixup by 1.5 points, while both lower that of the 30-epoch run on the original data, by 3.0 and 3.1 points on CIFAR-10 and by 4.9 and 5.5 points on CIFAR-100; together the two switches explain about half of the rank variance of a distilled-data sweep (interquartile range 41 to 65 percent) but 9 to 18 percent of that of the original data. The consistency ρ of Table 11 therefore ranks architectures within each of the four (DSA, mixup) settings, which hold about 36, 32, 16 and 16 configurations per trial seed, and averages the four Spearman correlations with these counts as weights. The table also reports the correlation of the joint ranking, in which the switches enter the comparison, and the correlations within the two largest settings alone.

Table 11: **NAS consistency on NATS-Bench.** Spearman's ρ between the rankings of 100 architectures per seed on distilled and on full data (30 epochs), within each augmentation setting, averaged over the settings and three seeds (std up to 0.17; Appendix F). Bold: best of each family.

		ConvNet-based methods									ResNet-based methods					
Dataset	IPC	ATT	DC	DM	DSA	FreD	MTT	NCFM	TESLA	*Med.*	CVDD	G-VBSM	SCDD	SRe^{2}L	WMDD	*Med.*
CIFAR-10	1	0.22	0.09	−0.27	0.18	**0.34**	−0.09	0.10	0.01	*0.09*	0.01	−0.21	**0.16**	0.06	−0.20	*0.01*
	10	−0.06	0.21	0.12	0.28	**0.66**	0.31	0.27	0.31	*0.27*	0.17	0.02	**0.31**	0.29	0.26	*0.26*
	50	**0.55**	0.23	0.40	0.38	0.54	0.51	0.40	0.52	*0.45*	0.30	0.32	**0.55**	0.54	0.52	*0.52*
CIFAR-100	1	0.21	0.17	−0.10	0.26	**0.65**	0.30	0.25	0.36	*0.26*	0.23	0.15	0.16	**0.30**	0.27	*0.23*
	10	0.59	0.18	0.27	**0.65**	0.43	0.51	0.36	0.41	*0.42*	0.38	0.41	0.44	**0.47**	**0.47**	*0.44*
	50	**0.63**	0.06	0.60	0.52	−0.17	0.44	0.24	0.23	*0.34*	0.75	**0.95**	0.75	0.75	0.78	*0.75*

Table 12: **NATS-Bench consistency on CIFAR100 and CIFAR10.** Spearman correlation between the rankings on distilled and on original data, mean over three trial seeds. ρ is the value of Table 11: computed within each augmentation setting (DSA on or off, mixup on or off) and averaged over the settings with their numbers of configurations as weights. ρ_{joint} ranks all configurations of a trial seed together, so that the two switches enter the ranking; ρ_{plain} and ρ_{DSA} are computed within the configurations trained without augmentation and within those trained with DSA but without mixup. For FreD and NCFM, IPC is the equivalent storage budget. CIFAR-100 NCFM at IPC 50 has a single trial seed (97 configurations), and CIFAR-100 CVDD at IPC 50 was trained without knowledge distillation.

Method	IPC	CIFAR100				CIFAR10			
		ρ	ρ_{joint}	ρ_{plain}	ρ_{DSA}	ρ	ρ_{joint}	ρ_{plain}	ρ_{DSA}
ATT	1	0.206	-0.118	-0.068	0.312	0.222	-0.039	-0.136	0.529
	10	0.590	0.143	0.530	0.582	-0.061	-0.157	-0.040	-0.277
	50	0.629	0.207	0.464	0.936	0.552	0.299	0.261	0.726
CVDD	1	0.226	0.038	0.063	0.309	0.013	-0.121	-0.132	0.117
	10	0.376	0.171	0.142	0.555	0.168	-0.015	0.062	0.366
	50	0.747	0.165	0.611	0.885	0.304	-0.009	0.032	0.612
DC	1	0.172	-0.149	-0.251	0.408	0.092	-0.113	-0.344	0.339
	10	0.179	-0.166	-0.137	0.079	0.208	-0.053	-0.277	0.416
	50	0.059	-0.339	0.117	-0.308	0.234	-0.065	-0.266	0.400
DM	1	-0.098	-0.302	-0.117	-0.124	-0.271	-0.377	-0.217	-0.427
	10	0.273	0.121	0.083	0.106	0.124	-0.067	-0.210	0.222
	50	0.598	0.204	0.104	0.902	0.396	0.213	0.020	0.665
DSA	1	0.263	-0.087	-0.096	0.487	0.185	-0.049	-0.218	0.393
	10	0.649	0.370	0.548	0.621	0.284	0.060	-0.110	0.415
	50	0.515	0.124	0.288	0.674	0.379	0.216	0.011	0.599
FreD	1	0.653	0.098	0.712	0.481	0.343	0.005	-0.043	0.621
	10	0.432	-0.006	0.317	0.584	**0.661**	0.204	0.446	0.772
	50	-0.166	-0.302	0.056	-0.284	0.541	0.170	0.431	0.810
G-VBSM	1	0.153	0.041	-0.038	0.233	-0.214	-0.262	-0.408	-0.138
	10	0.411	0.347	0.261	0.473	0.023	-0.150	-0.163	0.083
	50	**0.947**	0.858	0.941	0.937	0.320	0.129	0.165	0.461
MTT	1	0.302	-0.156	0.119	0.348	-0.092	-0.183	-0.282	-0.014
	10	0.515	0.146	0.285	0.569	0.310	-0.013	-0.069	0.493
	50	0.444	0.135	-0.030	0.878	0.507	0.191	0.140	0.703
NCFM	1	0.253	-0.039	-0.145	0.324	0.095	-0.089	-0.134	0.175
	10	0.361	0.065	-0.271	0.849	0.265	0.127	-0.225	0.605
	50	0.240	-0.171	-0.506	0.825	0.399	0.170	-0.229	0.885
SCDD	1	0.158	-0.118	-0.056	0.195	0.160	-0.069	-0.325	0.545
	10	0.440	0.203	0.352	0.540	0.305	0.018	0.090	0.343
	50	0.749	0.306	0.499	0.905	0.551	0.126	0.448	0.581
SRe^{2}L	1	0.299	0.008	0.061	0.499	0.056	-0.093	-0.140	0.129
	10	0.472	0.348	0.371	0.602	0.290	0.013	0.021	0.389
	50	0.749	0.422	0.540	0.904	0.544	0.152	0.503	0.563
TESLA	1	0.356	-0.014	0.078	0.480	0.014	-0.155	-0.232	0.012
	10	0.407	0.158	0.091	0.600	0.314	-0.011	-0.028	0.517
	50	0.234	-0.151	-0.274	0.775	0.519	0.303	0.251	0.685
WMDD	1	0.269	-0.003	0.013	0.451	-0.197	-0.293	-0.439	-0.068
	10	0.472	0.424	0.375	0.572	0.264	-0.106	0.034	0.484
	50	0.779	0.503	0.586	0.913	0.521	0.266	0.445	0.619

G EfficientNet Scaling Configurations

Table 13 lists the eight configurations of Figure 4b. All are derived from EfficientNet-B0 at its native resolution ($\gamma = 1$) and trade width for depth; $\alpha\beta^2$ stays between 1.94 and 2.03, the constraint of compound scaling (Tan & Le, 2020).

Table 13: Scaling configurations for compound scaling coefficients in EfficientNet experiments.

Index	α (depth)	β (width)	γ (resolution)
C1	1.00	1.41	1.00
C2	1.10	1.35	1.00
C3	1.20	1.30	1.00
C4	1.30	1.25	1.00
C5	1.40	1.20	1.00
C6	1.50	1.15	1.00
C7	1.60	1.10	1.00
C8	2.00	1.00	1.00

H Accuracy and Consistency along the Training Budget

Figure 8 shows, for Section 5, how the accuracy and the consistency of the distilled data change as training on them runs longer, on CIFAR-10 and CIFAR-100.

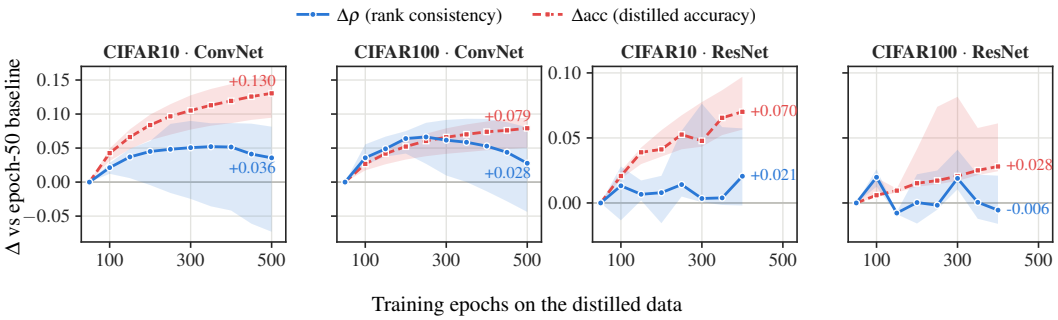

Figure 8: **Accuracy and consistency as training on the distilled data runs longer.** Change of distilled-data accuracy (Δacc) and rank consistency ($\Delta\rho$) relative to 50 epochs, median over methods and IPCs (bands: interquartile range). Accuracy keeps improving, while consistency saturates by 200 to 300 epochs and then declines for ConvNet methods and stays small and unstable for ResNet methods.

I Where the Consistency Comes From

Figure 9 splits the consistency of Section 6.1 into the contributions of hyperparameter groups.

J Simulated and Live Runs of the Ladder

Figure 10 shows the simulated search of Section 6.2.

In the simulation of Section 6.2, screening with the most consistent proxy (IPC 50, 300 epochs) never pays: it costs as much as 4 to 20 full-data epochs, which rank at least as well. Section 6.2 also runs the ladder live on CIFAR-100 (Table 14). MTT and WMDD were chosen before the live run as the ConvNet and ResNet-18 methods with the lowest simulated ladder regret on CIFAR-100, averaged over the nine budgets of Figure 10 (0.93 points for MTT against 1.40 to 2.29 for the other ConvNet methods; 1.43 for WMDD against 2.26 to 4.38). Each pool holds 300 configurations newly drawn

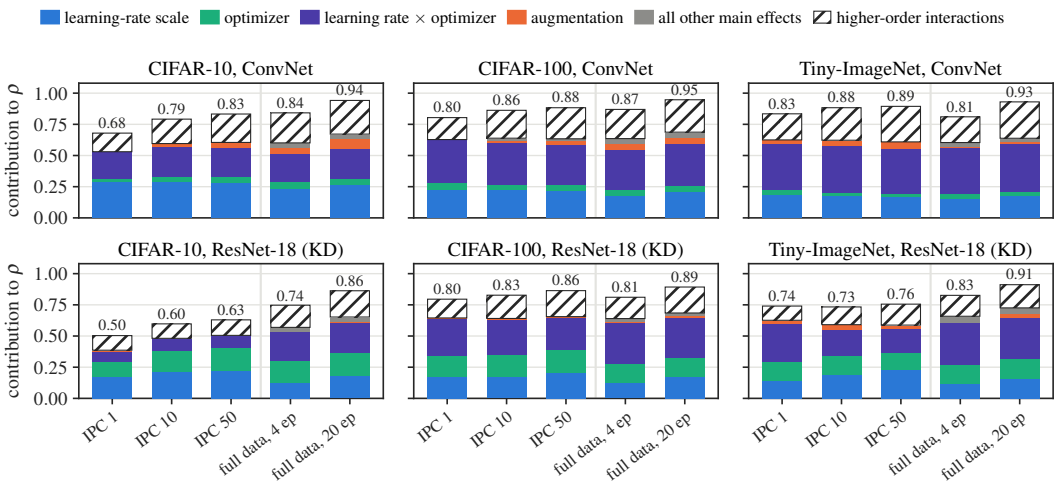

Figure 9: **Where the consistency comes from.** Consistency ρ of cheap evaluations with the 100-epoch full-data ranking, split by hyperparameter group: distilled data at IPC 1, 10 and 50 (mean over methods and training epochs) and full-data runs stopped after 4 and 20 epochs. The learning rate, the optimizer and their interaction carry most of it.

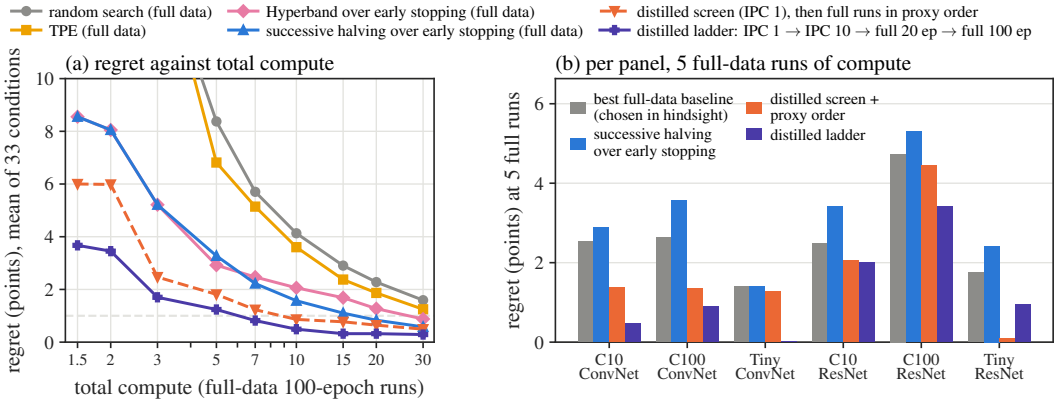

Figure 10: **Distilled data as the first stage of a hyperparameter search.** (a) Regret of the recommended configuration (points of full-data accuracy below the best in the pool) against total compute, distilled evaluations included; mean over 33 conditions (dataset, pipeline, DD method). (b) Regret at a compute of 5 full-data runs, per dataset and pipeline.

from the search space of the pipeline. Random search, successive halving and the ladder spend their budgets under the cost model of the simulation, and the configuration each recommends is trained on the full training set for 100 epochs. The 10,000 CIFAR-100 test images are split once into two class-stratified halves of 5,000; every selection uses one half and the table reports the other. The analysis uses all five pools of each pipeline, and no recommended configuration diverged. The lead of the ladder over successive halving is largest at 5 full-data runs (4.5 points for MTT, ahead in all five pools, one-sided sign test $p = 1/32$, the smallest possible with five pools; 2.0 points for WMDD) and smaller at 15 runs (2.0 and 0.3 points), as more full-data runs also let successive halving approach the best configuration.

K What a Reported Accuracy Depends On

Figure 11 gives the evidence behind the reporting protocol of Section 6.3, from the tricks runs of Section 6. The best of 10 random recipes is about 3 points (ConvNet) or 8 points (ResNet-18 with KD) below the best of all 300, and the best of 30 about half that; reporting accuracy against the

Table 14: **Live run of the ladder on CIFAR-100.** Mean test-half accuracy (%) of the recommended configuration over five pools of 300 new configurations; in parentheses, its regret (points) against the best test-half accuracy among the configurations of the same pool that any strategy trained for 100 epochs (not against all 300). Budgets in full-data 100-epoch runs, distilled evaluations included. Last column: pools in which the ladder is ahead of / tied with / behind successive halving.

Pipeline	Budget	Random search	Successive halving	Ladder	Ladder vs. SH
MTT (ConvNet)	5	52.64 (7.45)	55.30 (4.79)	**59.77** (0.32)	5 / 0 / 0
	15	55.99 (4.10)	58.10 (2.00)	**60.09** (0.00)	4 / 1 / 0
WMDD (ResNet-18, KD)	5	68.70 (6.66)	72.14 (3.22)	**74.12** (1.24)	4 / 0 / 1
	15	71.72 (3.63)	74.58 (0.77)	**74.85** (0.50)	2 / 2 / 1

number of recipes tried, as Dodge et al. (2019) recommend, would make this dependence visible. When every method is scored under its own recipe, the leaderboard agrees with the fully tuned one at a Kendall's τ of only 0.1, and one recipe shared by all methods reaches 0.3 to 0.4. Ten shared recipes, with each method at its best, reach 0.78 (ConvNet) and 0.58 (ResNet-18 with KD), as much as twenty to thirty recipes drawn separately per method, because the paired design removes the luck of the draw.

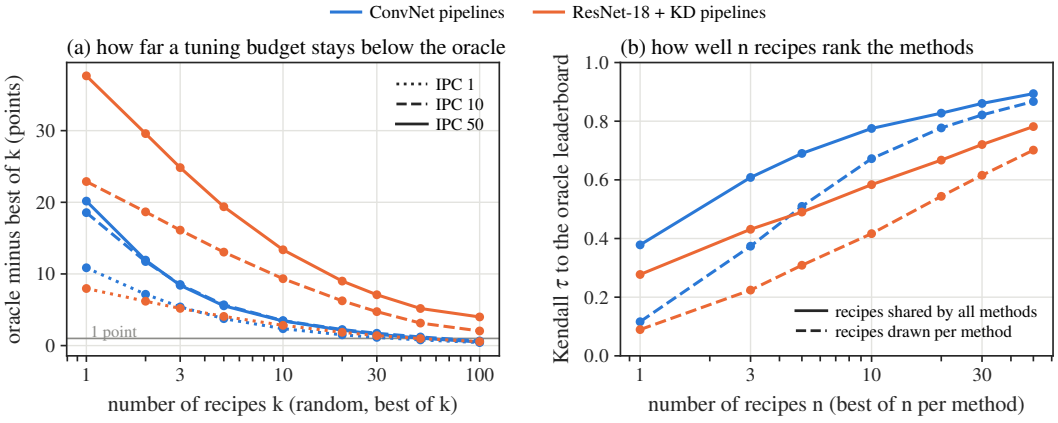

Figure 11: **What a reported accuracy depends on.** A recipe is one sampled training configuration. (a) Accuracy lost against the best of all 300 recipes when only the best of k random recipes is reported. (b) Agreement (Kendall's τ) between a leaderboard built from the best of n recipes per method and the leaderboard with every method fully tuned, with the recipes shared by all methods or drawn separately for each.