Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Model Annealing: Data-Efficient Learning through Review and Consolidation

Ali Mohammadi Ruzbahani

ali.mohammadiruzbaha@ucalgary.ca

University of Calgary https://orcid.org/0009-0004-3287-552X

Rezvan Abouzadeh

MetaKet INC.

Research Article

Keywords: Model Annealing, Self-distillation, Data-efficient learning, Knowledge distillation, Adversarial training, Meta-learning Calibration

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11720123/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Model Annealing: Data-Efficient Learning through Review and Consolidation

Ali Mohammadi Ruzbahani and Rezvan Abouzadeh

***Abstract*—A small labelled dataset is memorized long before the training budget runs out, and the one-hot labels that remain to be fitted then mostly sharpen confidence. Model annealing spends the rest of the budget on review and consolidation, relearning from the network's own knowledge. Training is divided into episodes whose learning rate is raised again and cooled, every cooled state joins a memory of past selves, and each new episode learns from the labels, from the tempered predictions of this memory, and from adversarial dreams sought where student and memory disagree most. The memory averages predictions rather than weights, and the deployed network is the last student, so inference costs nothing extra. Across CIFAR-100, Tiny-ImageNet and Imagewoof, four architecture families and 5% to 100% of the labels, model annealing raises the top-1 accuracy of the AdamW-trained ConvNeXt, ViT and MLP-Mixer over cross-entropy training in 28 of 33 configurations, by 1.3 points on average, and is on par in accuracy with snapshot distillation, the closest competitor. On these families it is better than both in likelihood, calibration and robustness to corruptions and small adversarial perturbations. For an SGD-trained ResNet-18 it lowers accuracy but still improves robustness to small adversarial perturbations. Separating the reheated schedule from relearning shows that relearning helps every family and that the losses come from reheats, which undo most of this network's fit to its training set. How early the accuracy of a cross-entropy run saturates largely separates the AdamW families from ResNet-18 but does not predict the gain within them. With little data, relearning from the latest snapshot without dreams, as snapshot distillation does, compounds overconfidence in the AdamW families from episode to episode. The memory and the dreams each curb most of this drift, and together they remove it.**

***Index Terms*—Self-distillation, data-efficient learning, knowledge distillation, adversarial training, meta-learning, calibration.**

I. Introduction

THIS paper trains deep networks on small labelled datasets in a cycle borrowed from human study: read the material, review it and work problems on it, let it settle, and return to it to build new knowledge on what was retained. Evidence on how people learn favors working with material over rereading it. An assessment of study techniques rates self-testing, such as answering practice questions on the material, among the most useful and rereading among the least [1]. In one experiment, restudying foreign words that students could already recall did not improve their recall a week later, whereas recalling the words again improved it greatly [2]. In another, students who reread a prose passage were more confident of remembering it than students tested on it, yet a week later they recalled less [3]. New memories, moreover, stabilize only gradually after learning, a process known as consolidation [4]. The cycle serves here as an organizing analogy, not as a model of human memory.

Plain supervised training does only the reading, over and over. A deep network often memorizes a small labelled dataset long before its training budget is spent. From then on the one-hot labels tell it little that is new, and the remaining epochs mostly sharpen its confidence. Like the students who reread, the network grows more confident without growing more accurate, so small-data training can fail twice: accuracy plateaus early, and the predictions become overconfident. The regime is common wherever labels are expensive or cannot be pooled. Runtime anomaly detectors for cyber-physical systems are typically learned from normal operating data alone [5], and organizations that would gain from each other's data are often unwilling to share raw data, gradients or model parameters at all [6].

Model annealing trains a network through the whole cycle. Its budget is divided into equal episodes, and the first episode, ordinary supervised training on the labels, is the first reading. Each later episode is a new study session in which the network, now a student of its own past, starts with a reheat: the learning rate returns to its peak, which frees the network to leave the solution it settled into. The session is spent on relearning, which combines review and practice. To review, the student learns not only from the labels but also from the tempered predictions of a memory of every state it has cooled into, an ensemble of past selves whose older members fade geometrically but never vanish. Review differs from rereading in what it presents: a tempered distribution does not saturate the way a one-hot label does, and it still says how the classes relate, so review gives the student something to learn after the labels are memorized. To practice, the student works problems called dreams. A one-step adversary searches a small neighborhood of a training image for the input on which the student departs most from its memory, and the memory labels that input, so dreams are problems off the training set, sought where current and past knowledge disagree most.

While the student reviews and practices, the learning rate cools along a cosine to nearly zero and the network settles. Every episode, the first included, ends in consolidation: a frozen copy of its cooled state joins the memory that all later episodes review. The training data never change, so the new knowledge an episode adds is what the network learns in it beyond the earlier episodes; consolidation keeps it for every later review. The deployed model is the last student, so inference costs no more than for a single network.

A. Mohammadi Ruzbahani is with the Smart Cyber-Physical Systems (SCPS) Lab, University of Calgary, Calgary, AB, Canada, and is also the Chief Technology Officer of MetaKet Inc., Calgary, AB, Canada (e-mail: ali.mohammadiruzbaha@ucalgary.ca).

R. Abouzadeh is a Senior Engineer with MetaKet Inc., Calgary, AB, Canada (e-mail: rezvan@metaket.com).

Training a network to match its own earlier predictions has a history. Born-again networks distill a model into an identical copy over several generations [7], snapshot distillation and progressive self-knowledge distillation reuse earlier states of the same run as teachers [8], [9], and knowledge evolution re-initializes part of a network over many generations to help small datasets [10]. In image classification these methods were evaluated mostly on convolutional networks, and two questions that decide whether self-relearning is worth its cost remain open: which of its ingredients are responsible for the gains, and for which architectures and amounts of data the gains turn into losses. Model annealing gathers the ingredients into one framework in which each can be switched on and off, and we use it to answer both questions.

Our contributions are the following.

- Model annealing, a single-run procedure that reheats and cools the learning rate in equal episodes, consolidates every cooled state as a frozen member of a function-space memory of past selves, and relearns from this memory and from adversarial dreams, at the inference cost of one network.
- A controlled study over three benchmarks, four architecture families and four amounts of labelled data, with paired tests, in which model annealing improves accuracy and likelihood over cross-entropy training for the three AdamW-trained families, where it is on par with snapshot distillation in accuracy but better in likelihood, calibration and robustness, and improves robustness to small adversarial perturbations for all four.
- A decomposition into a schedule effect and a relearning effect, showing that relearning helps every family and that the losses on ResNet-18 come from reheats that undo most of its fit, together with evidence that early saturation of a plain run largely separates the AdamW-trained families, where model annealing pays, from ResNet-18 but does not predict the gain within them.
- Evidence that, for the AdamW-trained families with little data, relearning from the latest snapshot without dreams compounds overconfidence over episodes, that a memory of all past selves and the dreams each curb most of this drift and together remove it, and that the memory opposes the labels late in each episode, where it restrains memorization.

II. Related Work

Knowledge distillation trains a student to match the temperature-softened output distribution of a teacher [11]. Born-again networks showed that a student with exactly the teacher's architecture can outperform it, and that the procedure can be repeated over several generations [7]. Each generation, however, costs a full training run. Snapshot distillation avoids that cost by dividing one run into cycles and letting the network at the end of a cycle teach the next one [8], while progressive self-knowledge distillation softens the hard targets with the model's own predictions from the previous epoch, weighting them more as training proceeds [9]. Predictions can also replace parameters as the carrier of knowledge between models, as in query-based collaborative learning among mutually distrustful parties [6]. Model annealing belongs to this family but changes what the student relearns from. Its teacher is a memory of all past selves rather than the latest snapshot, and part of the supervision is spent on inputs chosen adversarially instead of on the training images alone.

A second line of work restarts or perturbs optimization deliberately. Warm restarts of the learning rate improve the anytime performance of stochastic gradient descent [12]. Shrinking the weights and adding noise before training resumes closes the generalization gap that warm-started networks otherwise suffer [13], and knowledge evolution splits a network into a retained part and a re-initialized part, retraining it over many generations with clear gains on small datasets [10]. The name of our method comes from simulated annealing [14], where controlled heating followed by slow cooling lets a search leave poor local minima before it settles.

How the past states of a run should be combined is usually answered in weight space. Reptile moves an initialization toward the weights reached after a short inner training loop [15], and averaging the weights visited under a cyclical or high constant learning rate yields flatter solutions that generalize better [16]. Language models have recently been placed in the role of the meta-learner, writing training-data-free weight updates from the snapshots of a centralized run [17] and aggregation rules over client models in federated learning [18]; in the centralized case, the update the model arrived at in every run was an average of recent weight snapshots. Such averages are safe only between networks joined by a path of low loss. Copies of a network trained from one state under different mini-batch noise share a linear path of this kind only when that state lies sufficiently far into training, and copies trained from initialization do not [19]. Snapshot ensembles avoid the question by averaging the predictions of the networks reached at the end of each learning-rate cycle [20]. Model annealing keeps such an ensemble as its memory, uses it only as a teacher, and distills it into a single network within the same budget.

The adversarial component draws on two ideas. Generative adversarial networks cast learning as a minimax game between a generator and a discriminator [21], and adversarial belief matching transfers this game to distillation without data by training a generator to produce images on which student and teacher disagree [22]. Virtual adversarial training searches a small neighborhood of every input for the perturbation that changes the model's own prediction the most and penalizes that change [23]. Model annealing keeps the search local, as virtual adversarial training does, while the disagreement it maximizes is the one between the student and its consolidated past self.

Label smoothing also replaces a one-hot target by a distribution and improves both generalization and calibration [24]. The mass it spreads over the wrong classes is uniform, so it encodes nothing about which classes resemble each other. Comparing against it separates the value of structured, self-generated targets from the effect of target entropy as such.

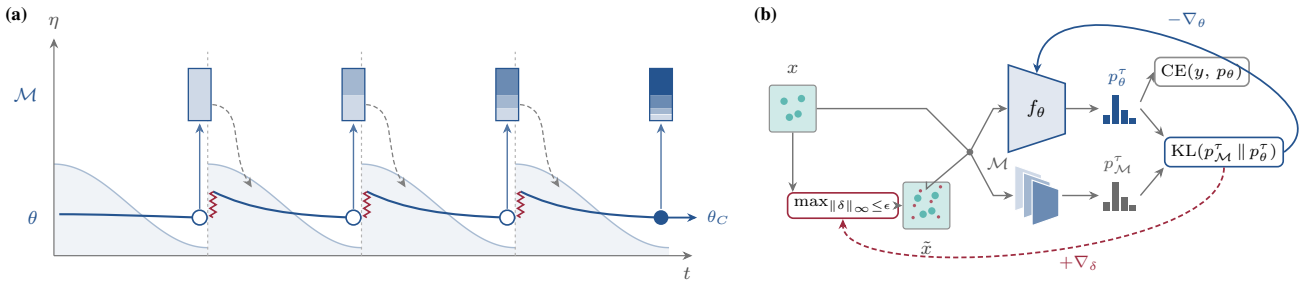


Fig. 1. Model annealing. (a) Reheated episodes and the memory of cooled snapshots. (b) One relearning step.

III. Model Annealing

Model annealing trains a student network f_θ and keeps a memory $\mathcal{M}$ of the states it has cooled into. The budget of T optimizer steps is divided into C episodes of equal length. In every episode the learning rate is warmed up to the same peak $\eta_{\max}$ and then decays along a cosine to nearly zero (Fig. 1(a)). Because the learning rate sets the scale of the gradient noise, raising it again at the start of an episode reheats the system, and the weights become free to leave the basin in which the previous episode came to rest. Only the cooled end of an episode enters the memory, a step we call consolidation. A network sampled at a high learning rate makes a poor teacher: for a ResNet-18 on CIFAR-10, the mean gradients of consecutive epochs at a high learning rate were found to be almost orthogonal [17], so the weights there move mostly with mini-batch noise. The first episode is ordinary supervised training. Each later episode relearns from the labels, from the memory and from adversarial dreams, and every episode, the first included, ends by adding its final state to the memory.

A. Relearning from a Memory of Past Selves

After c episodes the memory holds the snapshots $\theta_1, \dots, \theta_c$ and predicts

$$p^\tau_{\mathcal{M}}(x) = \sum_{j=1}^{c} w_{c,j}\, p^\tau_{\theta_j}(x), \qquad w_{c,j} = \begin{cases} (1-\mu)^{c-1}, & j = 1, \\ \mu\,(1-\mu)^{c-j}, & j \geq 2, \end{cases} \tag{1}$$

where p^τ is the softmax at temperature τ. The newest snapshot carries weight μ, and the influence of older ones decays geometrically without ever vanishing. During the next episode the student minimizes

$$\mathcal{L}(\theta) = (1-\alpha)\,\mathrm{CE}\big(y, p_\theta(x)\big) + \alpha\,\tau^2\,\mathrm{KL}\big(p^\tau_{\mathcal{M}}(x) \,\|\, p^\tau_\theta(x)\big) + \beta\,\tau^2\,\mathrm{KL}\big(p^\tau_{\mathcal{M}}(\tilde{x}) \,\|\, p^\tau_\theta(\tilde{x})\big), \tag{2}$$

where x is an augmented training image with label y and $\tilde{x}$ is a dream. The memory sees the same augmented view as the student, so it always speaks about the image the student is looking at. The supervision for a training image is therefore the mixture $q = (1-\alpha)\,e_y + \alpha\,p^\tau_{\mathcal{M}}(x)$ instead of the one-hot vector e_y. Its entropy is zero for cross-entropy and constant for label smoothing; here it varies from image to image and is shaped by what the network learned in all earlier episodes.

The weights in (1) are the ones a Reptile step $\phi_c = \phi_{c-1} + \mu\,(\theta_c - \phi_{c-1})$ [15] would assign to the snapshots if they were averaged in weight space. We average predictions instead. A weight average is meaningful when the averaged points share one basin, as successive iterates of a single trajectory at a small learning rate do [16]. Reheating, however, carries the weights far enough that the straight path between consecutive snapshots can cross a ridge of high loss [19]. Predictions can be averaged regardless, as snapshot ensembles do with the networks found at the end of their learning-rate cycles [20]. With $\mu = 1$ the memory reduces to the latest snapshot, the teacher of snapshot distillation [8]. The memory serves training only. The network returned at the end is the last student, a single model whose inference cost equals that of every baseline.

B. Dreams

Dreams are produced by an adversary that searches an ℓ_∞ ball around a training image for the point where the student departs most from its memory,

$$\tilde{x} = \Pi_\epsilon\big[x + \delta_0 + \tfrac{5}{4}\epsilon\,\mathrm{sign}(g)\big], \tag{3}$$

where $g = \nabla_\delta\,\mathrm{KL}\big(p^\tau_{\mathcal{M}}(x+\delta_0) \,\|\, p^\tau_\theta(x+\delta)\big)$ is evaluated at $\delta = \delta_0$, δ_0 is drawn uniformly from the ball, the memory's prediction is held fixed during the ascent step, and Π_ϵ projects back onto the ball and the valid pixel range. The adversary ascends the discrepancy that the third term of (2) descends. The two players form a game in the spirit of generative adversarial training [21], played locally around real images rather than in the output space of a generator. The memory labels every dream with its tempered distribution, so the student receives supervision at points that lie outside the training set and were selected to maximize the disagreement between its current knowledge and what it knew before. Algorithm 1 summarizes the procedure.

IV. Experimental Setup

The evaluation spans three image classification benchmarks of increasing resolution, four architecture families and four amounts of labelled data, with every comparison paired on identical data, initialization and batch order.

CIFAR-100 [25] contains 50,000 training and 10,000 test images of 100 classes at 32×32 pixels. Tiny-ImageNet is

Algorithm 1 Model annealing

Require: data $\mathcal{D}$, steps T, episodes C, temperature τ, weights α, β, radius ϵ, memory rate μ

initialize θ and an empty memory $\mathcal{M}$
for $c = 1, \dots, C$ **do**
  for T/C steps, learning rate warmed up to $\eta_{\max}$ and cosine-decayed **do**
    draw and augment a mini-batch (x, y) from $\mathcal{D}$
    if $c = 1$ **then**
      $\ell \leftarrow \mathrm{CE}(y, p_\theta(x))$
    else
      form dreams $\tilde{x}$ for half of the batch by (3)
      $\ell \leftarrow$ objective (2)
    end if
    take an optimizer step on ℓ
  end for
  add a frozen copy of θ to $\mathcal{M}$ and reweight it by (1)
end for
return θ

a 200-class subset of ImageNet [26] with 100,000 training images at 64×64 pixels; its test labels are not public, so its official validation split serves as the test set. Imagewoof[1] draws ten visually similar dog breeds from ImageNet, with 9,025 training and 3,929 test images that we resize to 144 pixels and crop to 128. From each official training set a fixed, stratified 10% is held out for validation, and nested class-balanced subsets of 5%, 10%, 25% and 100% of the remainder form the training sets, the full set being omitted for Tiny-ImageNet. Test images are evaluated once, after training. Every decision that depends on data, from learning rates to checkpoint selection and temperature scaling, is made on the validation split, and normalization statistics come from each run's own training subset. A search for exact and near-duplicate images finds 33, 7 and 3 test images of CIFAR-100, Tiny-ImageNet and Imagewoof with a counterpart in the training data; on the de-duplicated test sets no paired difference in top-1 accuracy between model annealing and a baseline changes sign. Out-of-distribution detection uses SVHN [27] as far-distribution data for all three benchmarks and the disjoint Imagenette classes as near-distribution data for Imagewoof.

All networks are trained from scratch. ResNet-18 [28] stands for convolutions with batch normalization, ConvNeXt [29] in an 8.6 M-parameter configuration for depthwise convolutions with layer normalization, ViT-Tiny [30] for self-attention, and an eight-block MLP-Mixer [31] of width 384 for pure token and channel mixing. Their stems are adapted so that ResNet-18 sees a 32×32 feature map, ConvNeXt a 16×16 map and the two token-based models 64 patches, whatever the input resolution. ResNet-18 is optimized with Nesterov SGD (momentum 0.9, weight decay 5×10^{-4}) and the other families with AdamW [32] (weight decay 0.05, gradient norm clipped at 1, drop-path rate 0.1). Batches hold 256 images, the learning rate warms up linearly over 5% of training and then follows a cosine decay, and a run lasts the larger of 100 epochs and 4,000 steps so that small subsets still receive enough updates. The base learning rate of each family was selected by a cross-entropy sweep with one cosine cycle on the CIFAR-100 validation split at 10% and 25% of the data and is shared by every method: 1.6 for ResNet-18, the largest value tried, and 2.5×10^{-4}, 5×10^{-4} and 10^{-3} for ConvNeXt, ViT and MLP-Mixer, the value for ConvNeXt being the smallest tried. The sweep used a slightly longer budget than the main runs, 120 epochs and at least 5,000 steps. Augmentation consists of reflection-padded random crops and horizontal flips at 32 and 64 pixels, and of random resized crops with scale between 0.35 and 1 plus flips at 128 pixels.

Model annealing is compared with cross-entropy training, label smoothing with $\epsilon = 0.1$ [24], SGDR with four cycles [12], and snapshot distillation [8] using four cycles and the same temperature and distillation weight as model annealing. Ablations on CIFAR-100 with 25% of the data, most of them repeated at 10%, replace the memory by the latest snapshot, remove the dreams, average the snapshots in weight space instead of function space, add Gaussian weight noise at every reheat, halve the peak learning rate from one episode to the next, or make every episode twice or half as long as the one before it; the front-loaded schedule is also run with SGDR. A cross-entropy run with twice the number of steps serves as a compute reference, and a diagnostic variant distills from a cross-entropy teacher trained on the entire training pool, which separates the quality of the teacher from the act of distillation. Model annealing uses $C = 4$, $\tau = 4$, $\alpha = \beta = \frac{1}{2}$, $\mu = \frac{1}{2}$ and $\epsilon = 4/255$, with dreams on half of every batch. The adversary evaluates the student in inference mode. These values were fixed before any experiment; the form of the memory and the use of dreams without weight noise were chosen on the validation split of CIFAR-100 at 10% and 25% data. Beyond top-1 accuracy we report negative log-likelihood, expected calibration error with 15 bins [33], the area under the ROC curve for out-of-distribution detection with the maximum softmax probability [34], and accuracy under five synthetic corruptions at three severities in the spirit of [35]. Paired differences are tested across configurations with the Wilcoxon signed-rank test, with Holm's correction over all comparisons of a table.

Training on the GPU is not bitwise deterministic, so identical computations can end in different networks. The first episode is the same computation for model annealing and snapshot distillation, from the same initialization on the same batches, yet the validation accuracies the two reach at its end differ with a standard deviation of 0.7 points over the 44 configurations, of 0.3 points for the AdamW families, and by at most 3.4 points. We use this spread as a yardstick for which differences between methods are meaningful.

V. Results

Table I reports the test accuracy of every configuration, Fig. 2 plots it against the amount of labelled data, Fig. 3 maps the gain of model annealing configuration by configuration, and Table II condenses the paired differences between model annealing and each baseline. The three families trained with

[1] https://github.com/fastai/imagenette

AdamW and ResNet-18, the only family trained with SGD, respond to model annealing in opposite directions, so we report them separately.

A. Accuracy, Calibration and Robustness

For ConvNeXt, ViT and MLP-Mixer, model annealing is more accurate than cross-entropy training in 28 of 33 configurations, by 1.3 points on average and by up to 4.3 points for ConvNeXt on CIFAR-100 with 10% of the data. It also exceeds label smoothing by 0.9 points and SGDR, whose learning-rate schedule it shares, by 1.3 points. Snapshot distillation is the closest competitor at 1.3 points above cross-entropy, and in accuracy model annealing is on par with it, with an average difference close to zero (Table II). Model annealing is ahead of it in 19 of 27 configurations with at most a quarter of the data (+0.14 points on average, less than the spread between identical computations described in Section IV) and behind it when all the data are used (-0.56 points). Model annealing is better calibrated and more robust than snapshot distillation, with a negative log-likelihood lower by 0.26 nats, an expected calibration error lower by 5.5 points, and accuracy higher by 0.2 points under corruptions and by 1.8 points under the PGD attack described below, all four differences significant (Table II).

Relearning from soft targets changes more than the top prediction. Top-5 accuracy rises by 3.3 points over cross-entropy and by 5.0 points over label smoothing. The negative log-likelihood falls by 1.58 nats relative to cross-entropy and remains 0.16 nats lower once both models are temperature-scaled on the validation split, so the improvement is more than a rescaling of confidence. The expected calibration error drops by 18.7 points, although label smoothing is better calibrated before temperature scaling in 20 of 33 configurations. The two kinds of soft target also differ in what they encode. On CIFAR-100 a uniform spread would place 4% of the non-target probability on the four other classes of the true superclass. Cross-entropy places 23.3% there, label smoothing 11.9%, and model annealing 23.4%: label smoothing weakens the similarity structure among the wrong classes, and a memory of past selves preserves it.

Robustness improves for every family. Accuracy under corruptions exceeds that of cross-entropy in 36 of 44 configurations, by 1.0 points on average, and accuracy under a ten-step PGD attack with $\epsilon = 1/255$ in 42 of 44, by 5.2 points. For ResNet-18 the gain under attack holds (+9.0 points) even though its clean accuracy falls, while the gain under corruptions shrinks to +0.3 points and is not significant. The AUROC for detecting SVHN images as out-of-distribution changes by +0.9 points, which is not significant.

ResNet-18 is the exception in clean accuracy. Model annealing trails cross-entropy by 1.8 points and label smoothing, the most accurate method for this family in 9 of 11 configurations, by 2.7 points. Snapshot distillation also falls behind cross-entropy (-0.9 points). The analysis below traces both losses to the reheated schedule rather than to relearning.

B. Where Relearning Helps, and Why

Fig. 4 follows a single configuration through training. Cross-entropy fits its training images within the first half of the budget, its validation accuracy stops rising, and its validation NLL climbs for the rest of the run, while that of snapshot distillation steps up at every reheat and that of model annealing stays close to its first-episode level. The first episode of model annealing ends after a quarter of the budget, and its snapshot becomes the first teacher. Fig. 5(a) relates the gain over cross-entropy to the share of its final validation accuracy that a cross-entropy run has reached at that point. For the AdamW families the share lies between 76% and 118%, and in 18 of 33 configurations the accuracy after one quarter is higher than at the end. In these configurations the remaining three quarters of the budget buy memorization rather than accuracy, and relearning puts them to better use. ResNet-18 has reached 44% to 84% by then, so the two ranges overlap between 76% and 84%. Across all configurations, share and gain have a Spearman correlation of 0.54 ($p < 0.001$), but across the configurations of the AdamW families only +0.07. The overall correlation thus rests on the difference between ResNet-18 and the AdamW families, and among the AdamW configurations the share does not predict the gain.

Model annealing differs from cross-entropy training in two respects, the reheated schedule and relearning from the memory. SGDR has the first without the second, so SGDR minus cross-entropy isolates the schedule and model annealing minus SGDR isolates relearning (Fig. 5(b)). Relearning raises accuracy for every family, by 0.7 to 1.6 points on average and in 33 of 44 configurations. The schedule is close to neutral for the AdamW families (-0.1 to +0.3 points) but costs ResNet-18 3.2 points on average and up to 9.7 points with 10% of the data. The difference lies in what a reheat does to each network (Fig. 5(c)). When the SGD learning rate of ResNet-18 returns to its peak, the training cross-entropy of the network jumps from 0.13 to 2.54 nats, against 3.76 at the start of training, and its accuracy on training batches, 98% on average before the reheat, falls to between 26% and 45%. The AdamW families, whose accuracy on training batches averages 78% before a reheat, lose at most 7 points of it per reheat, averaged over the reheats of each configuration. Between consecutive snapshots the weights of ResNet-18 move by 42% to 140% of their norm, against 6% to 32% for the other families. Every episode of this network thus begins far below the fit the previous one reached, and its snapshot records a short training run rather than a refinement of the previous one.

Front-loading the schedule tests this account (Fig. 5(d)). When the four episodes take 8/15, 4/15, 2/15 and 1/15 of the budget, the first snapshot of ResNet-18 comes within 1.4 points of the validation accuracy of a full cross-entropy run, against 5.0 points for the first of four equal episodes. The second episode, still long enough to recover from its reheat, adds 0.9 points in 2 of 2 configurations. The two short episodes that follow end before the network has recovered, and the last snapshot falls 6.5 points below the second. Relearning thus helps ResNet-18 too, but only within episodes that outlast the recovery from a reheat. The AdamW families hardly register a

TABLE I
TEST TOP-1 ACCURACY (%). BOLD: BEST METHOD PER MODEL AND COLUMN.

		CIFAR-100				Tiny-ImageNet			Imagewoof			
Model	Method	5%	10%	25%	100%	5%	10%	25%	5%	10%	25%	100%
ResNet-18	Model annealing	25.4	39.7	56.4	74.3	17.2	27.0	43.3	28.5	36.1	55.7	76.9
	Cross-entropy	26.8	43.0	59.3	75.2	18.3	29.0	43.6	**30.5**	39.3	56.6	78.8
	Label smoothing	**28.9**	**44.8**	**61.1**	**75.7**	**19.0**	**30.5**	**44.6**	29.5	**39.5**	56.6	**79.9**
	SGDR	19.3	33.3	58.1	74.1	11.2	26.6	43.8	28.0	35.3	56.4	79.4
	Snapshot KD	25.9	40.2	57.2	75.4	17.5	28.0	43.5	28.1	37.4	**57.9**	79.2
ConvNeXt	Model annealing	**17.7**	**24.6**	31.0	45.5	**8.7**	13.2	19.7	24.6	27.5	**36.7**	44.8
	Cross-entropy	15.8	20.2	30.3	44.1	7.6	13.1	20.1	23.2	26.6	35.4	46.8
	Label smoothing	16.5	21.0	30.1	45.1	7.4	13.1	19.9	23.7	27.7	35.3	**47.1**
	SGDR	15.8	20.7	30.7	44.9	7.4	**13.6**	**20.3**	23.6	27.2	35.5	46.4
	Snapshot KD	17.5	22.9	**31.4**	**46.0**	8.0	**13.6**	19.9	**24.6**	**27.9**	36.6	46.2
ViT	Model annealing	**15.7**	24.0	**37.3**	59.4	11.6	**18.6**	**28.3**	20.7	**25.8**	**36.0**	52.0
	Cross-entropy	15.2	22.3	35.4	57.2	10.5	16.5	25.2	20.7	24.9	34.0	51.1
	Label smoothing	15.4	23.2	35.1	57.2	10.7	16.3	26.3	**21.6**	25.3	33.5	50.5
	SGDR	14.8	21.7	35.3	57.2	9.9	16.3	25.3	20.9	25.3	33.9	51.4
	Snapshot KD	15.7	**24.2**	37.0	**59.7**	**11.7**	18.0	28.1	20.2	25.6	35.9	**53.0**
MLP-Mixer	Model annealing	**17.4**	**25.1**	**39.6**	57.7	**11.7**	**19.3**	**28.5**	23.4	26.3	37.6	**55.8**
	Cross-entropy	17.1	24.1	36.3	56.1	10.4	15.6	24.7	22.6	**27.7**	38.4	53.3
	Label smoothing	17.0	24.6	38.0	56.8	11.6	17.0	26.5	22.8	27.4	38.2	54.2
	SGDR	16.8	23.4	35.7	54.9	10.6	15.7	24.3	**23.6**	27.6	**38.6**	53.7
	Snapshot KD	17.3	24.9	39.3	**58.1**	11.7	19.0	28.4	22.9	26.3	38.3	55.6

TABLE II
MODEL ANNEALING MINUS EACH BASELINE, AVERAGED OVER CONFIGURATIONS (NLL IN NATS, OTHERS IN POINTS). *: $p < 0.05$, WILCOXON TEST WITH HOLM CORRECTION. WINS: CONFIGURATIONS WITH HIGHER TOP-1 ACCURACY.

Families	vs.	Top-1	NLL	ECE	Corr.	PGD	OOD	Wins
ConvNeXt, ViT, Mixer	Cross-entropy	+1.31*	−1.58*	−18.7*	+1.27*	+4.0*	+0.5	28/33
	Label smoothing	+0.90*	+0.15	+3.6	+1.00*	+3.0*	+0.1	27/33
	SGDR	+1.30*	−1.42*	−17.1*	+1.18*	+3.9*	+0.5	26/33
	Snapshot KD	+0.01	−0.26*	−5.5*	+0.22*	+1.8*	+0.1	20/33
ResNet-18	Cross-entropy	−1.82*	+0.37	+2.6	+0.35	+9.0*	+2.3	0/11
	Label smoothing	−2.69*	+0.68	+15.6	−0.36	+7.1	+2.4	0/11
	SGDR	+1.37	−0.38	−3.0	+1.31	+10.6*	+3.1	7/11
	Snapshot KD	−0.90	−0.05	−0.9	+0.06	+5.5*	−0.3	1/11

reheat, and for them front-loading merely postpones relearning (−0.88 points against equal episodes).

Snapshot distillation shares the schedule of model annealing but relearns from the latest snapshot alone, a snapshot that was itself trained to imitate its predecessor. With at most a quarter of the data this chain compounds overconfidence (Fig. 5(e)). For the AdamW families the validation NLL of its student rises by 0.06, 0.18 and 0.30 nats over episodes two to four, and the accuracy gained since the first episode stalls at 1.6 points after episodes three and four. Under model annealing the NLL never moves by more than 0.02 nats on average from its first-episode value, while the accuracy keeps rising to 1.7 points. The first snapshot never learned from a teacher and keeps a share of every later memory, which may anchor the targets against this drift. Snapshot distillation also lacks the dreams, and the ablations on CIFAR-100 at 10% and 25% of the data separate the memory from the dreams. From the first episode to the last, the validation NLL of the AdamW families changes by +0.18 nats under snapshot distillation, by +0.06 when the dreams alone are added to it, by +0.03 when the memory alone replaces its teacher, and by −0.04 under model annealing. On average, the memory and the dreams each curb most of the drift, and together they remove it. With all the data each snapshot clearly improves on its predecessor, the latest one appears to be the better teacher, and model annealing may pay for down-weighting it.

The gradient probe in Fig. 5(f) shows when the memory tells the student something the labels do not. On a fixed set of training images, the label and memory gradients of the AdamW families agree at the start of an episode (cosine 0.51) and oppose each other at its end (−0.36). Once the cooling student starts to fit the labels more sharply than its memory does, the memory holds it back. For ResNet-18 the two gradients remain aligned (0.89 early in the episode), and their cosine falls below half of that value only at about 90% of

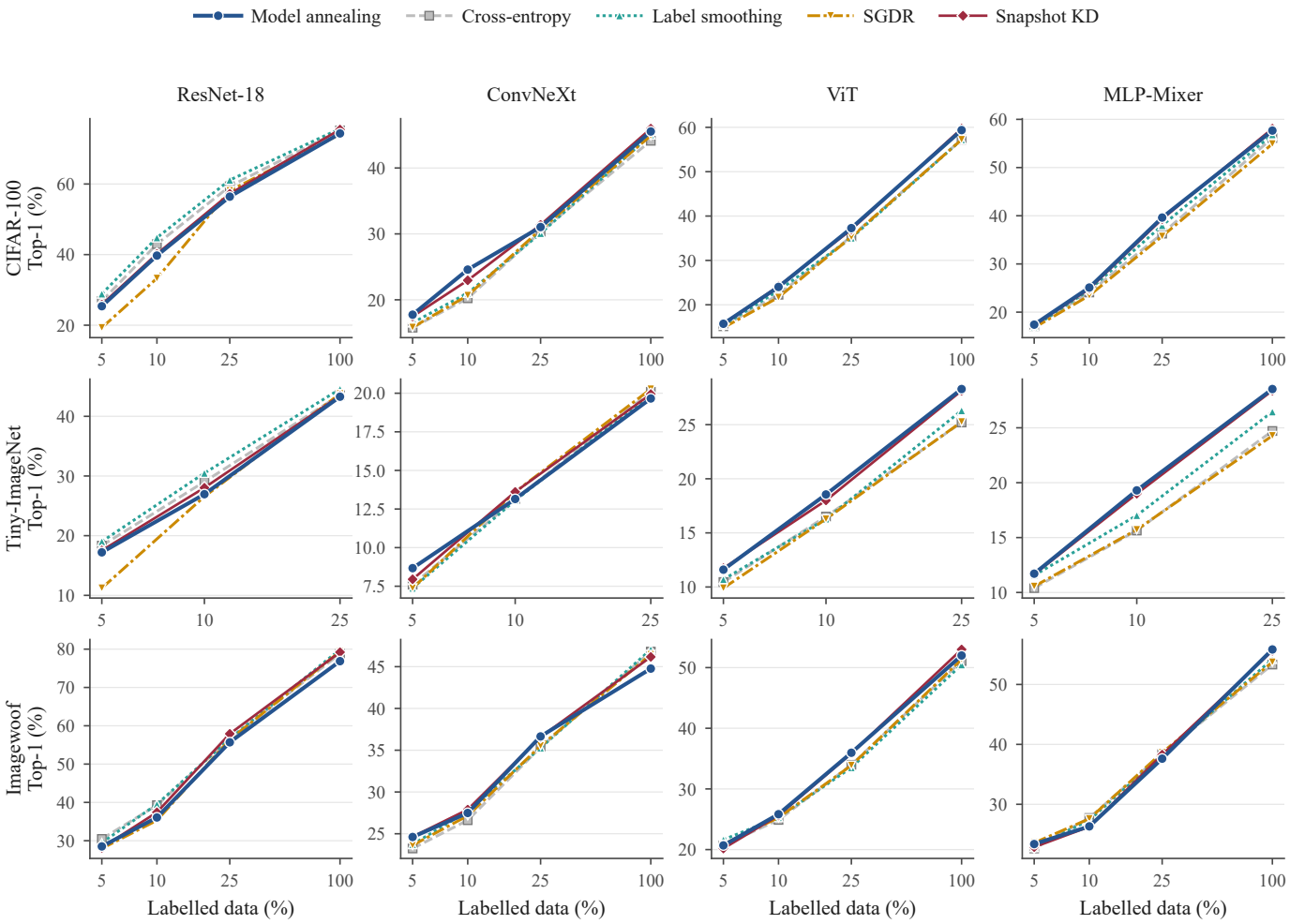


Fig. 2. Test top-1 accuracy against the amount of labelled data.

TABLE III
ABLATIONS ON CIFAR-100: VARIANT MINUS MODEL ANNEALING. UNITS AND * AS IN TABLE II.

	Top-1				
Variant	AdamW	SGD	NLL	Corr.	PGD
Latest snapshot as teacher	−0.18	−0.24	+0.082	+0.06	+0.4
Without dreams	−0.06	+0.88	+0.043	−0.53	−2.7
Weight-space memory	−0.28	−3.17	+0.033	+0.47	+0.6
With weight noise	−0.12	+0.88	−0.002	+0.06	−0.2
Halving reheats	−1.54	+0.61	+0.056	−0.61	−2.4
Growing episodes	−2.39	−2.44	+0.188	+1.02	−0.0
Front-loaded episodes	−0.88	−4.60	+0.777	−0.35	−0.4
Snapshot KD	−0.10	+0.71	+0.176*	−0.43*	−2.2*
Cross-entropy, 2× steps vs. cross-entropy	−0.36	+1.63	+0.740	−0.30	−0.2

the episode: the reheated student spends most of the episode recovering what it lost, and the memory repeats what the labels already say.

C. Ablations and Cost

Table III and Fig. 6 isolate the components on CIFAR-100. For the AdamW families no variant beats model annealing on average, and the schedule is what matters most: halved peaks, growing episodes and front-loaded episodes cost them 0.88 to 2.39 points. Swapping the memory for the latest snapshot or dropping the dreams moves their top-1 accuracy by at most 0.18 points, yet both raise the NLL, and over all four families the loss of the dreams also costs 2.7 points under PGD and 0.53 under corruptions. Weight noise at the reheats leaves them essentially unchanged. A memory averaged in weight space is the weakest teacher. It falls 1.4 points short of the student it is built from, whereas the function-space memory stays within 0.1 points of its student and is better calibrated by 0.08 nats, and it costs the AdamW families 0.28 points and ResNet-18 3.17, since the reheats of this network carry its snapshots far apart. ResNet-18 also suffers most from short episodes, and in this network the dreams trade 0.88 points of clean accuracy for 4.9 points under PGD.

Model annealing takes 2.7 times as long to train as cross-entropy (median over configurations) and snapshot distillation 1.3 times. The dreams and the forward passes through up to three memory members account for the difference, and inference costs the same for every method. Doubling the steps of cross-entropy changes accuracy by −0.4 points for the AdamW families, with gains in 3 of 6 configurations, and by +1.6 for ResNet-18, the family that saturates latest, with gains in 2 of 2. For the AdamW families more plain training

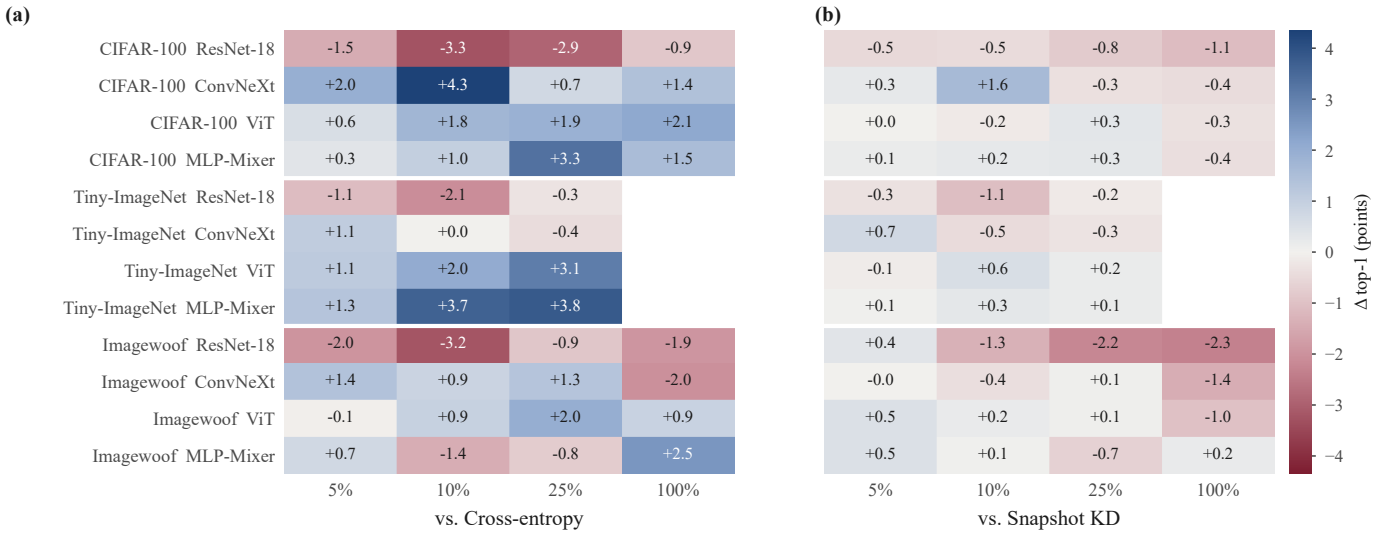


Fig. 3. Top-1 gain of model annealing over (a) cross-entropy and (b) snapshot distillation, per configuration.

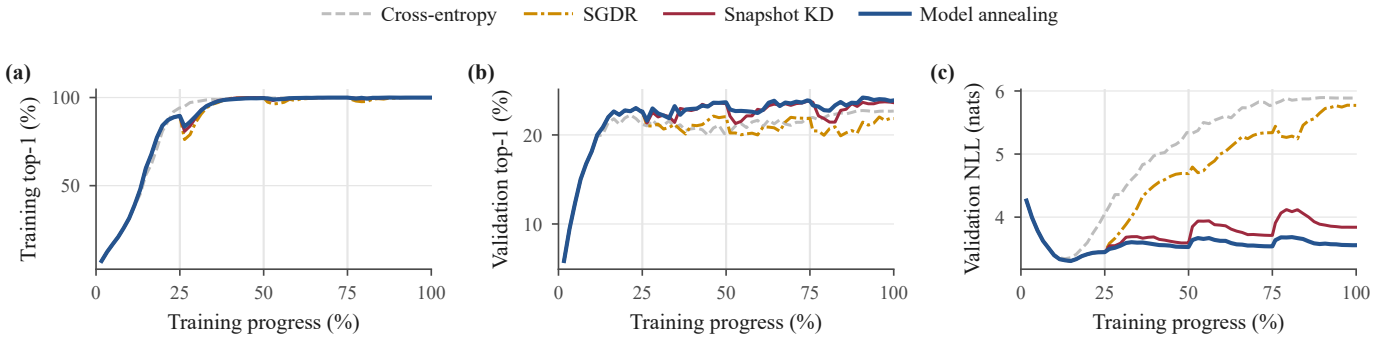


Fig. 4. Training of ViT on CIFAR-100 with 10% of the labels. Vertical lines mark the reheats.

thus buys no accuracy, and over all four families it raises the NLL (Table III). Distilling from a cross-entropy teacher trained on the entire training pool raises accuracy at 10% of the CIFAR-100 data by 7.8 points over cross-entropy for the AdamW families and by 17.6 for ResNet-18, and by 5.4 and 20.9 points over model annealing. This teacher has learned from labels the student never sees, so it marks what a better teacher is worth rather than competing with model annealing, and the large gain of ResNet-18 shows that learning from soft targets does not by itself harm this network.

VI. Discussion

Relearning helps every family although the memory is no more accurate than the student it is built from, which suggests that review passes on the relations among classes and a per-image confidence [7], [11] rather than better answers, and, for the AdamW families, a restraint on fitting the labels more sharply than the memory does late in each episode. For ResNet-18, whose student falls far below its memory after every reheat, review may instead repair what the reheat erased. Review and practice also divide the work. Both keep the likelihood from drifting across episodes, while practice on dreams supplies most of the robustness that model annealing adds over snapshot distillation, beyond the share that soft targets alone already bring.

For ResNet-18 a reheat does not so much open a new study session as undo much of the last one. Several properties set this network apart: it alone is trained with SGD, without gradient clipping or stochastic depth, it alone uses batch normalization, and its learning rate is the largest value of its sweep. ConvNeXt, convolutional but trained with AdamW, keeps its fit through reheats, so convolution is not what makes the difference. The sweep chose each peak for a network at the start of training, yet a reheat applies it to a network that has just cooled, and warm restarts and snapshot ensembles that reheated SGD-trained residual networks with batch normalization to lower peaks reported gains [12], [20]. This points to the peak rate, and to peaks matched to the cooled network as the remedy.

Nothing was tuned on Tiny-ImageNet or Imagewoof, and the gain of the AdamW families carries over to both: +1.7 points over cross-entropy on Tiny-ImageNet (8 of 9 configurations) and +0.5 on Imagewoof (8 of 12), against +1.8 on CIFAR-100, where the form of the memory was chosen. Because the memory speaks only through predictions, review could also draw on models whose weights cannot be shared, as in query-based collaborative learning [6]. Fine-tuning pre-

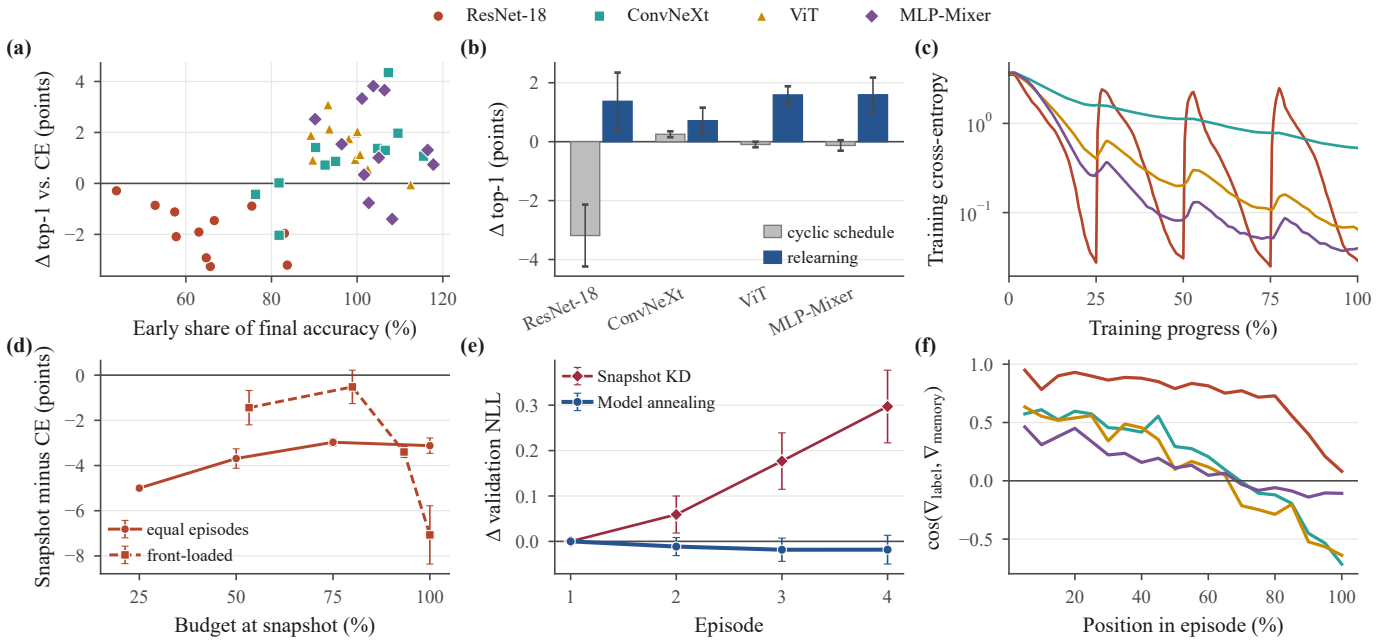


Fig. 5. (a) Gain over cross-entropy against its early share of final accuracy. (b) Schedule and relearning effects. (c) Training cross-entropy under model annealing. (d) ResNet-18 snapshots relative to cross-entropy. (e) Validation NLL relative to the first episode. (f) Cosine between label and memory gradients. Error bars: standard errors over configurations.

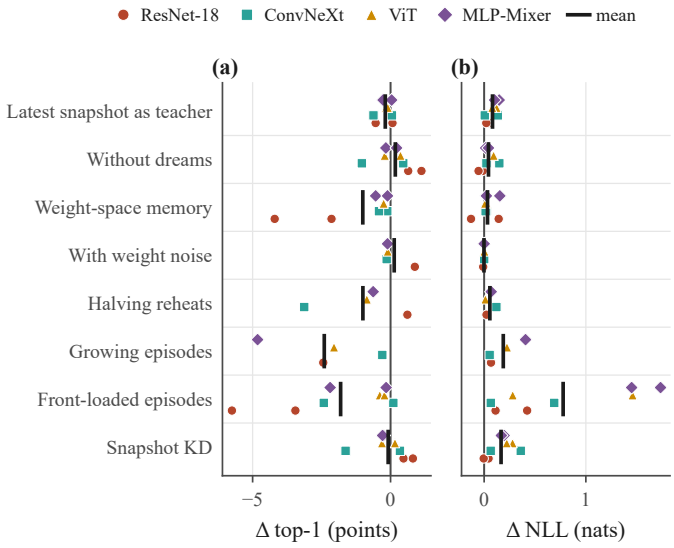


Fig. 6. Ablations on CIFAR-100, variant minus model annealing per configuration. Bars mark means.

trained networks, a common and strong choice when labels are few [36], is a natural next setting for review and consolidation.

VII. Conclusion

Model annealing turns the memorization phase of small-data training into a cycle of review, practice and consolidation. For the families trained with AdamW it raises accuracy, likelihood and robustness together, and it matches the strongest self-distillation baseline in accuracy while surpassing it in calibration and robustness. Review needs something to build on. Where a reheat erases most of what a network had learned, as it does for ResNet-18 under SGD, the evidence favors longer episodes or lower peaks. With little data, reviewing the whole notebook and practicing on dreams keep the likelihood of the student stable from one session to the next. Episode lengths and reheat peaks matched to each network, and memories that weight each snapshot by its validation likelihood, are natural next steps.

Acknowledgment

This research was funded by MetaKet Inc. AI tools were used to polish the text, correct grammar, improve readability, and debug code.

References

[1] J. Dunlosky, K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. T. Willingham, "Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology," *Psychol. Sci. Public Interest*, vol. 14, no. 1, pp. 4–58, 2013, doi: 10.1177/1529100612453266.

[2] J. D. Karpicke and H. L. Roediger, III, "The critical importance of retrieval for learning," *Science*, vol. 319, no. 5865, pp. 966–968, 2008, doi: 10.1126/science.1152408.

[3] H. L. Roediger, III and J. D. Karpicke, "Test-enhanced learning: Taking memory tests improves long-term retention," *Psychol. Sci.*, vol. 17, no. 3, pp. 249–255, 2006, doi: 10.1111/j.1467-9280.2006.01693.x.

[4] J. L. McGaugh, "Memory—a century of consolidation," *Science*, vol. 287, no. 5451, pp. 248–251, 2000, doi: 10.1126/science.287.5451.248.

[5] A. Mohammadi Ruzbahani, A. Yazdinejad, and H. Karimipour, "PIA: Agentic physics-informed runtime anomaly detection," *arXiv preprint arXiv:2611.11009*, 2026, doi: 10.48550/arXiv.2611.11009.

[6] A. Mohammadi Ruzbahani, A. Yazdinejad, and H. Karimipour, "Poster: Decentralized trustless query-based collaborative learning under adversarial heterogeneity without model sharing," in *Proc. 47th IEEE Symp. Security and Privacy (SP), Poster Session*, San Francisco, CA, USA, May 2026. [Online]. Available: https://www.ieee-security.org/TC/SP2026/downloads/posters/sp2026posters-final24.pdf

[7] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, doi: 10.48550/arXiv.1805.04770.

[8] C. Yang, L. Xie, C. Su, and A. L. Yuille, "Snapshot distillation: Teacher-student optimization in one generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2854–2863, doi: 10.1109/CVPR.2019.00297.

[9] K. Kim, B. Ji, D. Yoon, and S. Hwang, "Self-knowledge distillation with progressive refinement of targets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 6547–6556, doi: 10.1109/ICCV48922.2021.00650.

[10] A. Taha, A. Shrivastava, and L. S. Davis, "Knowledge evolution in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 12 838–12 847, doi: 10.1109/CVPR46437.2021.01265.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015, doi: 10.48550/arXiv.1503.02531.

[12] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, doi: 10.48550/arXiv.1608.03983.

[13] J. T. Ash and R. P. Adams, "On warm-starting neural network training," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, doi: 10.48550/arXiv.1910.08475.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983, doi: 10.1126/science.220.4598.671.

[15] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018, doi: 10.48550/arXiv.1803.02999.

[16] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, 2018, doi: 10.48550/arXiv.1803.05407.

[17] A. Mohammadi Ruzbahani, A. Yazdinejad, and H. Karimipour, "GradReason: LLMs as meta-learners that replace training epochs with training-data-free weight updates," *arXiv preprint arXiv:2611.11005*, 2026, doi: 10.48550/arXiv.2611.11005.

[18] A. Mohammadi Ruzbahani, A. Yazdinejad, and H. Karimipour, "FedReason: Large language models as weight-space meta-aggregators for heterogeneous federated learning," *arXiv preprint arXiv:2611.08383*, 2026, doi: 10.48550/arXiv.2611.08383.

[19] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 3259–3269, doi: 10.48550/arXiv.1912.05671.

[20] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, doi: 10.48550/arXiv.1704.00109.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, doi: 10.48550/arXiv.1406.2661.

[22] P. Micaelli and A. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, doi: 10.48550/arXiv.1905.09768.

[23] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, 2019, doi: 10.1109/TPAMI.2018.2858821.

[24] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, doi: 10.48550/arXiv.1906.02629.

[25] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011. [Online]. Available: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[29] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 11 966–11 976, doi: 10.1109/CVPR52688.2022.01167.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, doi: 10.48550/arXiv.2010.11929.

[31] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-Mixer: An all-MLP architecture for vision," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, doi: 10.48550/arXiv.2105.01601.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, doi: 10.48550/arXiv.1711.05101.

[33] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, doi: 10.48550/arXiv.1706.04599.

[34] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, doi: 10.48550/arXiv.1610.02136.

[35] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, doi: 10.48550/arXiv.1903.12261.

[36] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (BiT): General visual representation learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 491–507, doi: 10.1007/978-3-030-58558-7_29.