DIVERGENCE CONTROLS ENTROPY IN DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distillation has become a core primitive of large language model training, but its properties are not yet well understood. We take an entropic perspective, studying how the entropy of the student depends on the data and the divergence that define the distillation objective. We prove that forward KL inflates the entropy of the student above that of the teacher. Since cross-entropy training is a special case, this yields an identity that we verify quantitatively in pretraining and supervised finetuning. Other divergences come with no such guarantee: reverse KL deflates entropy until the gap between student and teacher gets too large, and interpolating between the two changes entropy smoothly early in training but abruptly at convergence. The lower entropy of on-policy distillation comes from token-level reverse KL, not from on-policy sampling. The divergence therefore acts as an implicit entropy regularizer, whose role is clearest in self-distillation: as conditioning on privileged information deflates entropy, the divergence hyperparameters that work best are those that compensate for it.

1 INTRODUCTION

Distillation consists of training a student model to mimic the behavior of a teacher model and compress its knowledge (Hinton et al., 2015). It is used in increasingly diverse contexts for large language models, from pretraining lightweight, more efficient models (Muralidharan et al., 2024; Gemma Team, 2025), to solidifying reasoning abilities in posttraining (Ma et al., 2026; DeepSeek-AI, 2026). In practice, distillation amounts to minimizing a divergence between two categorical probability distributions. While divergence minimization has been extensively studied elsewhere (Minka, 2005; Bishop, 2006; Amari, 2016), its properties are not yet well understood in the context of large language models.

In this work, we study the properties of these algorithms from an entropic perspective, investigating how design choices affect the entropy of the student. Entropy is fundamentally a measure of uncertainty, whether in thermodynamics (Boltzmann, 1877) or information theory (Shannon, 1948), and it is particularly important for large language models as it relates to the diversity of the output distribution (Kirk et al., 2024; Farquhar et al., 2024) and controls exploration in reinforcement learning (Cui et al., 2025; Cheng et al., 2026; Wang et al., 2025). It is also easy to compute and logged by any training run. Here, we develop intuition for how distillation shapes entropy.

Our contributions, each developed in its own section, are as follows:

- **Decoupling sampling and divergence choice (Section 2).** We review distillation, decoupling the choice of the data generation process from that of the divergence, which enables us to study their effects on the student's entropy independently.
- **From divergence to entropy (Section 3).** In a toy model of language modeling, we show that forward KL inflates the student's entropy above the teacher's by the residual divergence, reverse KL deflates it until mimicking the teacher gets too difficult and then suddenly inflates it, and interpolating between the two varies entropy smoothly early in training but abruptly at convergence. Top-k restrictions also shift entropy, depending on how the tail is handled.
- **At scale, the divergence implicitly controls entropy (Section 4).** After pretraining and supervised finetuning, the entropy of a language model matches the cross-entropy loss it was trained to minimize; larger models therefore have lower entropy. When distilling from a larger teacher, the divergence drives entropy more than on- vs. off-policy sampling; in self-distillation, it counters the entropy drop that conditioning on privileged information induces.

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.

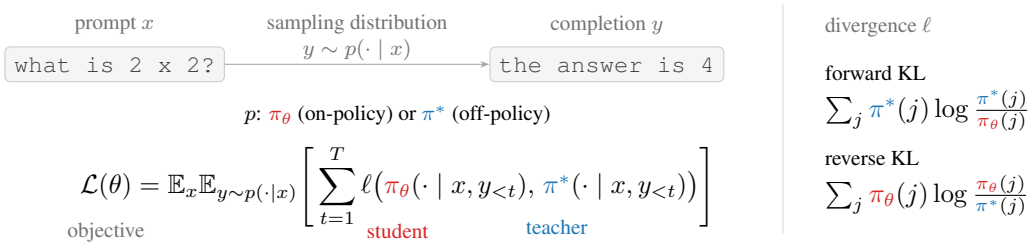

Figure 1: **Decoupling divergence from sampling in distillation.** A prompt x is completed by the sampling distribution p, and the student π_θ is then compared to the teacher π^* on each token of the completion through a divergence ℓ between their next-token distributions. Writing the objective at the token level keeps the two choices independent, whereas a sequence-level divergence ties them together. Details in Section 2.

2 Decoupling Sampling and Divergence Choice in Distillation

Distillation algorithms, in particular on-policy ones, are often derived by minimizing a divergence between the student's and the teacher's distributions over sequences of tokens. As we will argue, this couples the choice of token-level objective with the sampling distribution, which makes their respective effects on the student hard to disentangle and can yield suboptimal configurations. Following Czarnecki et al. (2019) and Agarwal et al. (2024) *inter alia*, we decouple the two, as summarized in Figure 1 and detailed in Appendix A.

Student, teacher and sampling distributions. We consider data consisting of prompts x and completions y generated by a sampling distribution p. On these completions, the student distribution $\pi_\theta(y \mid x)$ parametrized by θ is compared to that of the teacher $\pi^*(y \mid x)$. The sampling distribution p can either be the student itself, in which case learning is on-policy, or the teacher, in which case we call it off-policy distillation. The teacher can either be explicit or implicit, as when training on internet text through next-token cross-entropy minimization.

Distillation objective. The objective of distillation is to minimize the expected divergence between the student and the teacher over a dataset of prompts x and completions y,

$$\mathcal{L}(\theta) = \mathbb{E}_x \mathbb{E}_{y\sim p(\cdot|x)} \left[\sum_{t=1}^{T} \ell\left(\pi_\theta(\cdot \mid x, y_{<t}), \pi^*(\cdot \mid x, y_{<t})\right) \right], \tag{1}$$

with T the length of the completion and ℓ a divergence between the full next-token distributions of the two models, such as the forward or the reverse KL. The objective is here defined at the token level, which is what enables us to decouple the sampling distribution p from the divergence choice ℓ. Otherwise, the sampling distribution would be implicitly defined by the divergence choice—a sequence-level reverse KL forces on-policy data and a sequence-level forward KL forces off-policy data. In practice, gradients are typically not backpropagated through the sampling process, even when it is the student itself, with rare exceptions (e.g., Gu et al., 2024). The same approximation is used when improving the reasoning abilities of language models through reinforcement learning with verifiable rewards (Lambert et al., 2024; Guo et al., 2025). It amounts to ignoring the score function term, which biases the gradient but considerably reduces its variance (Williams, 1992; Roeder et al., 2017; Naesseth et al., 2018; Fu et al., 2026). This observation motivates ignoring sequence modeling aspects of distillation in our toy model of Section 3.1.

Entropy. Core to our analysis is the average entropy of the model π, which we define as

$$\bar{H}(\pi) = \mathbb{E}_x \mathbb{E}_{y\sim p(\cdot|x)} \left[\frac{1}{T} \sum_{t=1}^{T} H\left(\pi(\cdot \mid x, y_{<t})\right) \right], \tag{2}$$

with H the Shannon entropy of a distribution over the vocabulary. The distribution π need not be the one that generates the contexts $(x, y_{<t})$. When $p = \pi = \pi_\theta$, this is the student's normalized sequence-level entropy.

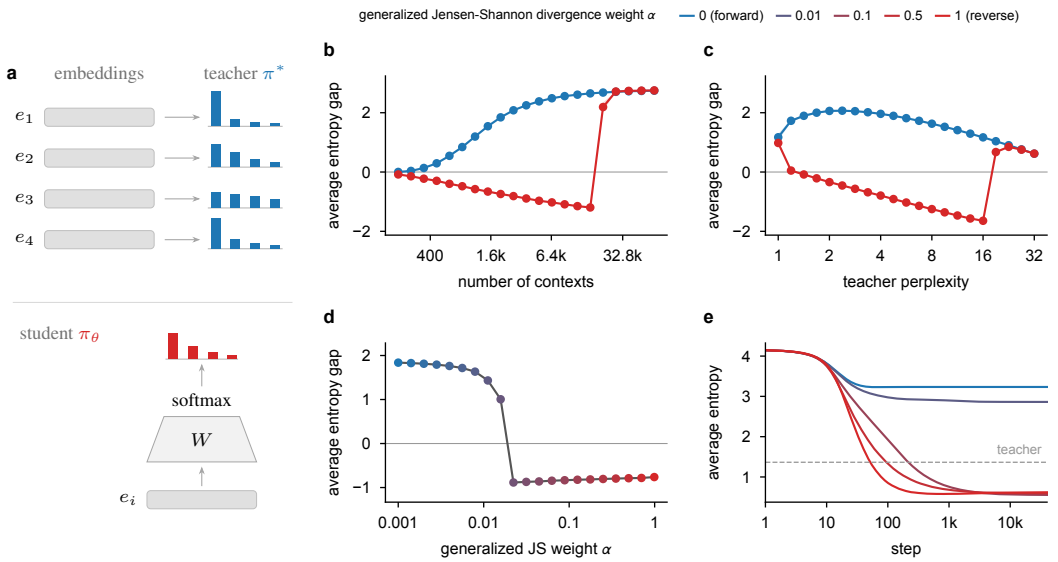

Figure 2: **A toy language model reveals phase transitions in how the divergence affects entropy. Toy model (a).** A linear softmax student maps a fixed embedding e_i per context i to next-token probabilities and is trained towards a teacher distribution per context with forward KL, reverse KL, or the generalized Jensen–Shannon (JS) divergence whose weight α interpolates between them (Section 3.1). Its shared weights W prevent it from reproducing all teacher distributions when there are too many contexts. **Simulations (b–e).** At convergence, forward KL (blue) always inflates entropy relative to the teacher and reverse KL (red) tends to deflate it, unless the teacher is too hard to model because there are too many contexts (b) or because it is too diffuse (c), in which case reverse KL inflates entropy too. Along the generalized JS family (colors in between), entropy smoothly decreases with the weight α early in training (e), whereas at convergence it drops abruptly (d).

3 Properties of Divergences and Their Effect on Entropy

The formalism of the previous section enables us to study the effect of the sampling distribution and the divergence choice independently. In this section, we restrict ourselves to that of the divergence as it is more tractable to analyze theoretically, and defer the empirical study of the sampling distribution to the next section. We first introduce a toy model that captures part of language modeling and use it as an illustrative example to highlight the properties of different divergence functions.

3.1 A Toy Model of Language Modeling

The canonical example to illustrate the properties of forward and reverse KL divergences is fitting a mixture of Gaussians with a single Gaussian distribution (Bishop, 2006). While this has been used to illustrate some of the properties of distillation or reinforcement learning in the context of large language models (Agarwal et al., 2024; Gu et al., 2024; Chen et al., 2025; Nicolicioiu et al., 2026), its continuous and non-neural-network nature makes it an imperfect proxy for what happens in practice.

Instead, we propose the following toy model, which takes as input a random embedding e_i associated with each context i in the data distribution and outputs a next-token distribution through

$$\pi_\theta(\cdot \mid i) = \mathrm{softmax}(We_i). \tag{3}$$

Such a model can be understood as the prediction head of a language model whose preceding sequence model provides random embeddings, as an associative memory model (Cabannes et al., 2024; Nichani et al., 2025; Zucchet et al., 2025), or as the simplest neural network that can produce categorical distributions over possible next tokens. This model has been shown to be able to fit a number of contexts proportional to the number of parameters of the linear layer (Cabannes et al., 2024; Nichani et al., 2025). Varying the number of contexts therefore allows us to control the difficulty of the task, or equivalently the capacity of an implicit teacher. Another knob is the entropy

of the teacher distribution, or equivalently the teacher perplexity $\exp(H(\pi^*))$, that is the effective number of tokens over which the teacher places probability mass; increasing it also makes the task harder (Zucchet et al., 2026).

Contrary to language models, this toy model has no sequence modeling component, which makes it easier to analyze theoretically. Little is lost since gradients are not propagated through the sampling process: in both cases, the distillation objective mostly aims at improving next-token prediction and not at shaping the sequence sampling distribution.

Throughout this section, we leverage this model when needed to illustrate some of the theoretical properties of the different divergences. In our simulations, as a default we set the embedding dimension to 128, the vocabulary size to 64, the number of contexts to 2048, and sample the teacher perplexity log-uniformly between 1 and 16. The model is trained with full-batch Adam (Kingma & Ba, 2015) and a learning rate of 0.1 for 40k steps; Appendix B.7 details the setup.

3.2 Divergence choice only matters when student and teacher differ

Before characterizing what each divergence does to entropy, it is worth delimiting when the choice matters at all. Indeed, a well-established property of KL divergences is that the geometries of the forward and the reverse KL are equivalent around optimality (Amari, 2016). In our case, this means that if the student is capable of reproducing the teacher and already starts close to it, the learning dynamics of distillation with forward and reverse KL are the same. Both objectives also share the same global optimum $\pi_\theta = \pi^*$. More formally, when π_θ is close to π^* on all contexts i, we have

$$\nabla_\theta \, \mathbb{E}_i \left[D_{\mathrm{KL}}(\pi^*(\cdot \mid i) \parallel \pi_\theta(\cdot \mid i)) \right] \approx \nabla_\theta \, \mathbb{E}_i \left[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid i) \parallel \pi^*(\cdot \mid i)) \right] \tag{4}$$

up to first order. The entropy of the student along these trajectories is therefore also the same. We emphasize that the statement is independent of the sampling distribution p, as long as it stays constant during training. This argument applies more broadly to any f-divergence, and in particular to the generalized Jensen–Shannon family that we use later. A proof of the general statement can be found in Appendix B.1.

A consequence of that result is that any difference in behavior induced by the choice of divergence occurs when the student and the teacher distributions differ significantly, e.g., because the student cannot represent the teacher or because the two start far from each other. The rest of this section will implicitly assume that we are in such a regime. Figure 2b illustrates part of this property in our toy model: for a small enough number of contexts, the student and the teacher have almost identical entropy at convergence regardless of the divergence choice, as the student is close to matching the teacher.

3.3 Forward KL predictably inflates entropy

The forward KL divergence is known to be mode-covering (Bishop, 2006; Murphy, 2022): as

$$D_{\mathrm{KL}}(\pi^*(\cdot \mid i) \parallel \pi_\theta(\cdot \mid i)) = \sum_j \pi^*(j \mid i) \log \frac{\pi^*(j \mid i)}{\pi_\theta(j \mid i)} \tag{5}$$

the objective diverges whenever $\pi^*(j \mid i) > 0$ and $\pi_\theta(j \mid i) \to 0$, so the student tends to cover all of the teacher's support. From this intuition, we expect the student to have higher entropy than the teacher; we now show that this does indeed hold at optimality.

Take a neural network with a linear softmax head. This includes standard language model heads as well as our toy model of Section 3.1. At optimality, if we were to multiply the weight matrix by a scalar factor, the loss would not change up to first order. The consequence of that, which we prove in Appendix B.3 following classical arguments from information geometry (Amari, 2016), is that

$$\bar{H}(\pi_\theta^{\mathrm{for}}) = \bar{H}(\pi^*) + \mathbb{E}_i \left[D_{\mathrm{KL}}(\pi^*(\cdot \mid i) \parallel \pi_\theta^{\mathrm{for}}(\cdot \mid i)) \right] \tag{6}$$

when $\pi_\theta^{\mathrm{for}}$ is a local optimum of the forward KL distillation objective. In particular, the student necessarily has higher entropy than the teacher on average, which Figure 2b and 2c confirm. This statement is only relative to the teacher: the forward KL can still deflate entropy over the course of learning if the teacher's entropy is much lower, as Figure 2e illustrates.

Given that the cross-entropy loss minimization objective can be seen as off-policy distillation with the forward KL, in which continuations of the data are samples from an implicit teacher, this result extends to pretraining and supervised finetuning as well. Expanding the KL, Equation 6 states that the average entropy of the student equals its cross-entropy with the teacher, that is its training loss, both measured on the training data. This does not require the two distributions to be close, only a linear softmax head, as in large language models, and a student at a stationary point of the forward KL. A direct consequence is that, among models trained to convergence, a higher loss comes with a higher entropy: a student that cannot match its teacher ends up more uncertain rather than overconfident. The experiments of Section 4.1 will confirm that these statements hold quantitatively in practice.

3.4 Reverse KL deflates entropy until the gap gets too large

Whereas the forward KL is mode-covering, the reverse KL is mode-seeking (MacKay, 2003; Bishop, 2006; Margossian & Saul, 2023). This mode-seeking bias of the reverse KL divergence suggests that a student trained with it would have lower entropy than the teacher. In practice, this is indeed the case in a wide range of scenarios, but breaks when the task becomes too hard, as illustrated in Figure 2b and 2c when there are too many contexts to model or when the teacher is too diffuse.

Theoretical analysis is not as easy as for the forward KL, so we focus on providing intuition for why this happens.[1] Let us start by rewriting the reverse KL divergence as

$$D_{\mathrm{KL}}(\pi_\theta(\cdot \mid i) \parallel \pi^*(\cdot \mid i)) = -\sum_j \pi_\theta(j \mid i) \log \pi^*(j \mid i) - H(\pi_\theta(\cdot \mid i)). \tag{7}$$

The first term is the mode-seeking part of the divergence: whenever $\pi^*(j \mid i) \to 0$, the student must place low probability mass on token j to avoid a large loss. As the teacher gets more diffuse, the first term flattens out, and the second one, which pushes the student's entropy up, dominates the objective. As a result, the mode-seeking bias of the reverse KL disappears and the student ends up inflating its entropy. We can formalize that intuition by analyzing the behavior of the reverse KL divergence in the limit of a flat teacher distribution. This analysis, provided in Appendix B.5, shows that the reverse KL divergence, as well as any divergence in the generalized Jensen–Shannon family, becomes equivalent to the forward KL divergence in that limit, and therefore inflates entropy. It does not predict the exact point at which the transition occurs, but it does quantify how the entropy gap between the teacher and the student evolves: the gap widens as the number of contexts grows and narrows as the teacher gets more diffuse, matching Figure 2b and 2c.

One may wonder whether the inflation part of the statement is relevant in practice: in the toy experiments it only occurs when the number of contexts is large or the teacher's output diversity is high, cases in which distillation is unlikely to bring significant performance improvements. In our large language model experiments of Section 4, we do not observe any average entropy inflation with the reverse KL. That said, the analysis here still gives us some insight into the biases of the reverse KL divergence. Even when inflation does not hold on average, it should still occur on a subset of contexts, and the intuition developed here hints at which ones are more likely to see the entropy of their next-token distribution inflated.

3.5 In between forward and reverse KL, entropy can sharply change

Forward and reverse KL are not the only divergences used in practice, and it is common to interpolate between the two through the generalized Jensen–Shannon divergence (von Werra et al., 2020; Agarwal et al., 2024; Hübotter et al., 2026; Zhao et al., 2026b; Xing et al., 2026). We wonder whether the entropy of the student smoothly and monotonically increases as we interpolate between the two divergences. The results, illustrated in Figure 2d and 2e, show that it depends on the stage of training: early in training entropy smoothly decreases as the interpolation weight α increases (Figure 2e), whereas at convergence it drops abruptly (Figure 2d). That said, the phase transition boundary does evolve smoothly as a function of α, cf. Figure 7 in the appendix, and divergences with larger α values need harder tasks to inflate entropy. As a result, one of the consequences of

[1] The stationarity argument behind Equation 6 also applies to the reverse KL, but it then constrains the spread of the log probabilities of the student rather than its entropy (Appendix B.4).

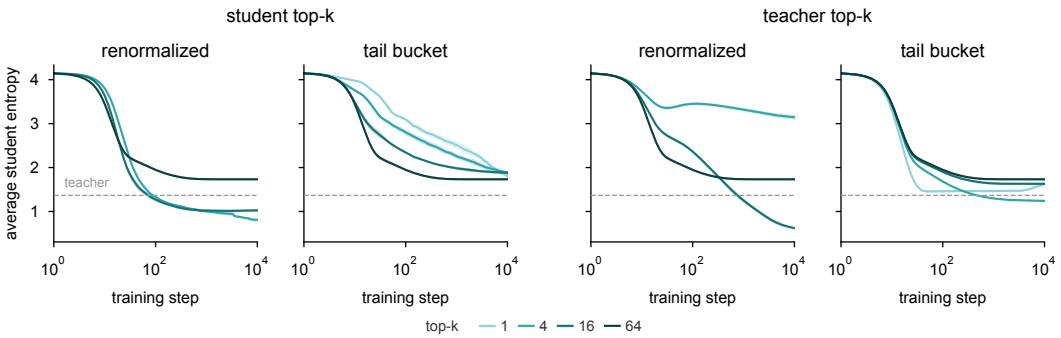

Figure 3: **Restricting the divergence to the top-k tokens shifts entropy depending on how the support and the tail are handled.** Same setup as in Figure 2 with 512 contexts, this time varying k out of a vocabulary of 64, when using the student (left) or the teacher (right) to select the top-k tokens and when using the renormalized or tail bucket cropping strategy. Both choices substantially change how the restriction affects entropy. Forward KL here; other divergences in Figure 8.

picking a divergence in the generalized Jensen–Shannon family is an implicit regularization on the entropy of the student, which can be either smooth or sharp depending on the training stage.

3.6 Top-k restrictions shift entropy depending on how the tail is handled

An important design choice that we have not yet touched upon is how much of the total probability distribution to take into account when computing the divergence. So far, we have only used full vocabulary divergences, but in practice it is common to restrict the divergence to the top-k tokens of the teacher or student distribution for compute and bandwidth efficiency (Fu et al., 2026; Hübotter et al., 2026; Xing et al., 2026). Two choices arise: which distribution selects the top-k tokens (student or teacher), and how the mass outside the selected set is handled. The *renormalized* variant renormalizes both distributions over the top-k tokens (Fu et al., 2026; Xing et al., 2026), whereas the *tail bucket* keeps the tail mass as one extra bucket (Hübotter et al., 2026; Huang & Wei, 2026); see Appendix A.2.2.

We find that top-k restrictions can have a non-negligible effect on the entropy of the student and that the effect depends strongly on the two design choices above. When the student selects the top-k tokens, renormalization keeps entropy relatively independent of k, at least early on, whereas the tail bucket yields increasingly large entropies as k decreases. When the teacher selects the top-k tokens, the early-training picture reverses. The results presented here hold for the forward KL divergence, and we additionally sweep over the generalized Jensen–Shannon divergence weight α in Figure 8 in the appendix, finding that these observations are mostly independent of the choice of divergence, although we observe some interactions.

4 Entropy in different distillation scenarios

We now turn to empirical validation of the intuition and of the theoretical properties of divergences derived in the previous section, as well as to studying the effect of the sampling distribution. We consider three settings, which are both practically relevant and will enable us to test different parts of the theory: pretraining, distillation in posttraining for reasoning, and self-distillation.

4.1 Cross-entropy predicts entropy in pretraining and supervised finetuning

Next-token cross-entropy minimization as used during pretraining and supervised finetuning can be understood as sampled off-policy distillation with forward KL, in which the teacher is an implicit distribution over continuations of the data. In that case the divergence is fixed: we only have access to samples from the teacher distribution, so we cannot use reverse KL or any other divergence that would require direct access to the teacher's probabilities. As discussed in Section 3.3, Equation 6 predicts that the entropy of the student is equal to its cross-entropy loss at convergence, both measured on the training data. Testing it requires the cross-entropy form of the statement, which

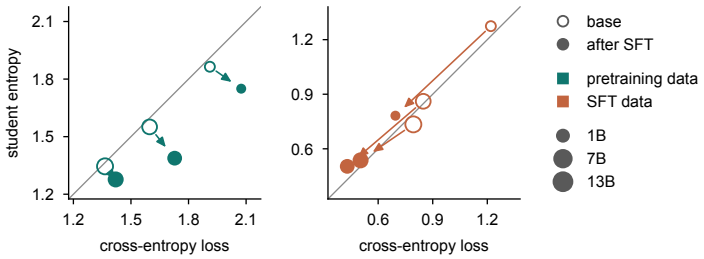

Figure 4: **Cross-entropy tracks entropy in pretraining and supervised finetuning.** Average entropy of the student $H(\pi_\theta)$ against its average cross-entropy loss on the data, both computed from the student alone, for OLMo 2 at different sizes (marker size), before (hollow) and after (filled) supervised finetuning (SFT), on the pretraining (left, green) and finetuning (right, orange) mixtures; the grey line is entropy equals cross-entropy. The identity holds on the data each model was last trained on, as predicted, and also, without guarantee, on the finetuning data after pretraining, whereas supervised finetuning breaks it on the pretraining data.

only needs samples from the teacher, rather than its form in terms of the entropy of the teacher and the KL divergence to it, which needs the probabilities of the teacher. Yet, compute-optimal large language models are typically stopped before convergence (Kaplan et al., 2020), although the trend is to longer pretraining runs (Hoffmann et al., 2022; Sardana et al., 2024). We therefore wonder whether pretraining and supervised finetuning runs are long enough for this relationship to hold; we find it to hold surprisingly precisely in practice, as illustrated in Figure 4. Despite the simplicity and robustness of that result, we are not aware of any prior work that has reported it within the language model literature.

To show this, we run experiments with the OLMo 2 suite of models (OLMo, 2025), which provides multiple pretraining and supervised finetuned checkpoints with varying model sizes, as well as the pretraining and finetuning datasets that were used to train the models. For each model size, we evaluate the different checkpoints on both pretraining and finetuning data, and report the entropy as a function of the cross-entropy loss. Overall, the identity holds, both after pretraining and supervised finetuning. Cross-entropy loss decreases as we scale models, so larger models have lower entropy. Figure 9 in the appendix confirms that the same holds for the Pythia suite of models (Biderman et al., 2023). Entropy decreases everywhere during finetuning even though we are using a forward KL objective, which inflates entropy. We reiterate that the inflation is relative to the teacher, and that lower entropy overall reflects the low entropy of the finetuning data, which is for example a well-known property of alignment data (West & Potts, 2025; Lanchantin et al., 2025; Zhang et al., 2025; Springer et al., 2026).

Zooming out, this result informs us about how pretrained models fail to capture the data distribution: they tend to leak probability mass outside the support of the data distribution rather than overconfidently predicting high probability tokens.

4.2 On-policy distillation's lower entropy is due to token-level reverse KL

So far, we have focused on the divergence and kept the sampling distribution fixed, which is the setting in which the results of Section 3 hold. On-policy distillation is increasingly used in the post-training of large language models (Agarwal et al., 2024; Gu et al., 2024; Yang et al., 2025; Thinking Machines Lab, 2025; Xiao et al., 2026; Yang et al., 2026), as training on its own completions removes the exposure bias (Bengio et al., 2015) that models trained off-policy face. It is usually paired with reverse KL, as both follow from the same sequence-level derivation (Gu et al., 2024; Thinking Machines Lab, 2025), so that a loss of diversity could a priori come from either choice. In this set of experiments, we ablate the effects of the divergence (forward vs. reverse KL) and the sampling distribution (on-policy vs. off-policy) on entropy. We distill the Qwen3-Instruct models (Yang et al., 2025) in non-thinking mode into a Qwen3-0.6B-Base student on the math problems of DeepScaleR (Luo et al., 2025). We measure entropy on MATH500 (Lightman et al., 2024) on student completions in Figure 5, and on frozen teacher completions in Figure 10 in the appendix.

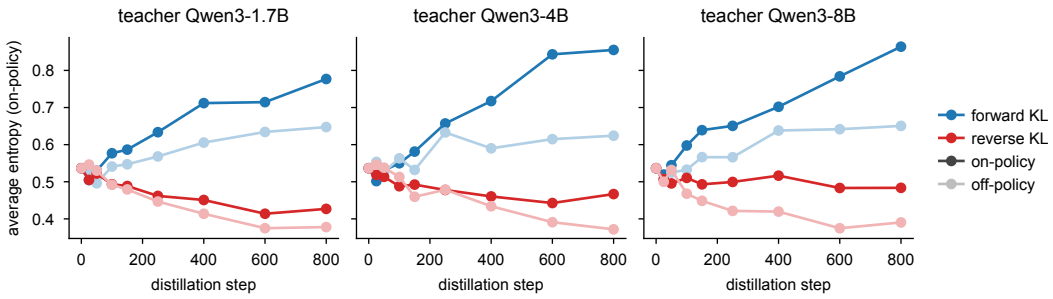

Figure 5: **The lower entropy of on-policy distillation is due to token-level reverse KL, on-policy sampling nudges it up, and teacher size barely matters.** Average entropy of the Qwen3-0.6B-Base student on its own MATH500 completions over distillation from Qwen3-Instruct teachers of increasing size in non-thinking mode (left to right). Color is the divergence, with the full color for on-policy and a lighter shade for off-policy, and each curve is a single seed.

The divergence has the largest effect on entropy: forward KL raises the entropy of the student and reverse KL lowers it, for every teacher and both sampling distributions, consistently with the ordering of Section 3. On-policy distillation consistently yields higher entropy than off-policy distillation, but this effect is two to three times smaller than that of the divergence. Zhao et al. (2026a) similarly report that the prefix source has a much smaller effect on entropy than the direction of the KL, particularly on short completions. On-policy sampling therefore modulates the effect of the divergence, and the lower entropy of on-policy distillation is mainly due to reverse KL. Finally, teacher size barely matters: the three teachers have almost the same entropy on their own completions and are all obtained by distillation from the same larger models (Yang et al., 2025), and Li et al. (2026) have shown that teachers trained with the same pipeline induce nearly identical next-token distributions on the contexts the student visits despite their difference in size.

4.3 Self-distillation deflates entropy, divergence can compensate for it

In self-distillation, the teacher is the student itself, given additional feedback on how to solve the task. This sidesteps the need for a more capable model to distill from, and improves performance on a variety of tasks. Nicolicioiu et al. (2026) have shown that conditioning on task feedback decreases the output diversity of the teacher, and thus that of the student. Here, we focus on how the choice of divergence interacts with this effect. As the teacher is derived from the student, any entropy the student loses is also lost by the teacher, which can lead to entropy collapse.

We follow the experimental setup of Hübotter et al. (2026) on the chemistry subset of SciKnowEval (Feng et al., 2024), which consists of multiple-choice questions testing scientific knowledge and reasoning in chemistry. We finetune Olmo 3 7B Instruct (Team Olmo, 2025), starting from the default hyperparameters of Hübotter et al. (2026), and varying three knobs that are important in practice: the divergence within the generalized Jensen–Shannon family, the frequency at which the teacher is updated, and the number of tokens taken into account when computing the divergence. As in the implementation of Hübotter et al. (2026), these tokens are the top-k tokens of the student. The main results are reported in Figure 6, and we run the same experiments on GSM8K (Cobbe et al., 2021) in Appendix C.5, with similar conclusions.

Overall, we find that the default hyperparameters are ones that stabilize entropy, and that moving away from them can lead to entropy collapse. When varying the weight of the generalized Jensen–Shannon divergence (Figure 6 left), entropy smoothly decreases with α, consistently with the early learning dynamics of the toy model (Figure 2e). In the longer runs of Appendix C.5, entropy sharply increases between $\alpha = 0.25$ and $\alpha = 0$, although not as abruptly as in the toy model at convergence (Figure 2d). This could be due to the almost on-policy nature of the data, which makes different runs see different contexts. With reverse KL, entropy collapses (Figure 6 left): the entropy deflation of reverse KL compounds with the reduced diversity of the teacher, which in turn lowers the entropy of the next teacher, and so on. Entropy also collapses when the teacher is the live student (Figure 6 center), whereas an exponential moving average, even with a decay of 0.95, behaves like a frozen teacher and prevents the collapse. Restricting the divergence to fewer tokens (Figure 6 right) affects

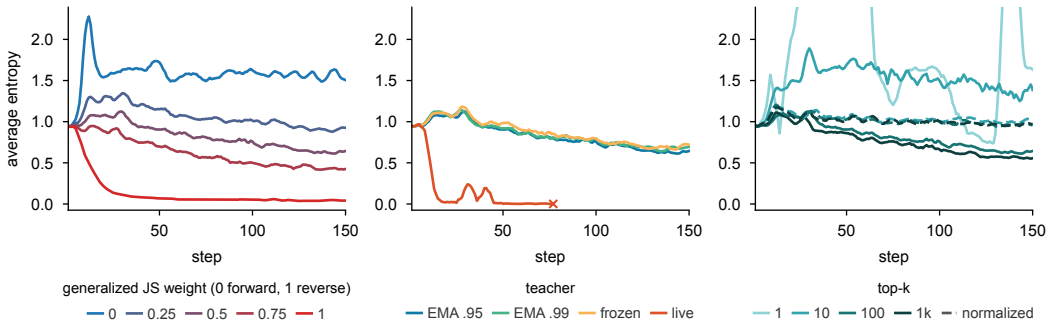

Figure 6: **Self-distillation tends to deflate entropy, the choice of divergence must compensate for it to avoid entropy collapse.** Self-distillation experiments on the Chemistry dataset following Hübotter et al. (2026), varying the weight of the generalized Jensen–Shannon (JS) divergence (left), the frequency at which the teacher is updated (center), and how many tokens are taken into account (right); defaults are $\alpha = 0.5$, a teacher that is an exponential moving average (EMA) of the student with decay 0.95, and the top-100 tokens of the student with tail bucket. Each curve is a single seed. Conditioning on privileged information lowers the entropy of the teacher (Nicolicioiu et al., 2026), and as the teacher derives from the student, this can compound into entropy collapse, e.g., with reverse KL (left) or a live teacher (center). Divergences that inflate entropy, a slowly moving teacher, or a tail bucket (right) counterbalance it.

entropy as the toy model of Section 3.6 predicts early in training when the student selects the top-k tokens (Figure 3 left): with the tail bucket, entropy increases as k decreases, whereas renormalizing leaves it roughly independent of k. Performance-wise, these different hyperparameters account for differences of a few percentage points as long as entropy does not collapse, cf. Figure 11.

Our results emphasize the importance of separating the choice of divergence from the sampling distribution. Indeed, deriving self-distillation from a sequence-level reverse KL to justify on-policy sampling forces the choice of reverse KL, which collapses entropy and damages performance. This is consistent with self-distillation papers (e.g., Shenfeld et al., 2026; Hübotter et al., 2026; Zhao et al., 2026b) not finding reverse KL to work best in practice.[2]

5 Conclusion

The divergence used in distillation is often inherited rather than chosen: a sequence-level derivation ties reverse KL to on-policy data and forward KL to off-policy data. We have shown that this choice is not innocuous, as the divergence acts as an implicit entropy regularizer on the student. Forward KL inflates the entropy of the student above that of the teacher by exactly the residual divergence on average, so the entropy of a converged pretraining run can be read from its loss. Reverse KL deflates entropy, but only as long as the student can mimic the teacher; whether it does is a property of the divergence and the task together, not of the divergence alone. In between, and depending on how the tail of the distribution is handled, entropy can be tuned, smoothly early in training and abruptly at convergence. These effects compound in self-distillation, where the teacher is derived from the student: any entropy drop feeds back into the next teacher, and the hyperparameters that work best are those whose inflation counteracts the drop that conditioning on privileged information induces.

Practically, our results suggest choosing the sampling distribution and the divergence separately, and treating the latter as an entropy knob rather than as a consequence of the former.

We obtained these results with a toy model of language modeling, simple enough to admit precise mathematical statements yet rich enough to reproduce key phenomena at scale. We hope that it will replace the Gaussian mixtures inherited from variational inference as the go-to example to reason about distillation in large language models, and that it will prove useful beyond entropy.

[2] Shenfeld et al. (2026) report using reverse KL in the paper, but the code uses forward KL.

AI USE STATEMENT

In this work, we used generative AI tools for

- helping with the literature review.
- assisting in the development of the theory by formulating mathematical claims and providing insights into how to prove them. We used these tools to draft proofs and carefully rewrote them ourselves to ensure correctness and improve clarity.
- providing feedback on experiments, writing code, running experiments, analyzing the results and producing the figures.
- polishing the writing of the paper and drafting some sections of the appendix. The first draft of the main text was written exclusively by the authors.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International conference on learning representations*, 2024.

Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

Lukas Balles and Philipp Hennig. Dissecting Adam: the sign, magnitude and variance of stochastic gradients. In *International conference on machine learning*, 2018.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in neural information processing systems*, 2015.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and others. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, 2023.

Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. When does optimizing a proper loss yield calibration? In *Advances in neural information processing systems*, 2023.

Ludwig Boltzmann. Über die beziehung zwischen dem zweiten hauptsatze der mechanischen wärmetheorie und der wahrscheinlichkeitsrechnung respektive den sätzen über das wärmegleichgewicht. *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften. Mathematisch-Naturwissenschaftliche Classe*, 76, 1877.

Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories. *International conference on learning representations*, 2024.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 40, 2026.

Lucas Clarté, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Expectation consistency for calibration of neural networks. In *Uncertainty in artificial intelligence*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Wojciech M. Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant M. Jayakumar, Grzegorz Świrszcz, and Max Jaderberg. Distilling policy distillation. In *International conference on artificial intelligence and statistics*, pp. 1331–1340. PMLR, 2019.

DeepSeek-AI. DeepSeek-V4: towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017), 2024.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. SciKnowEval: evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: an 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: knowledge distillation of large language models. In *International conference on learning representations*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081), 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and others. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Huipeng Huang and Hongxin Wei. Tail-aware top-k on-policy distillation. *arXiv preprint arXiv:2608.14728*, 2026.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International conference on learning representations*, 2015.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International conference on learning representations*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th symposium on operating systems principles*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International conference on learning representations*, 2024.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-preview with a 1.5B model by scaling RL. Blog post, 2025.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: multi-teacher on-policy distillation for capability integration in LLM post-training. *arXiv preprint arXiv:2606.30406*, 2026.

David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

Charles C. Margossian and Lawrence K. Saul. The shrinkage-delinkage trade-off: an analysis of factorized Gaussian approximations for variational inference. In *Uncertainty in artificial intelligence*, pp. 1358–1367. PMLR, 2023.

Tom Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, 2005.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. In *Advances in neural information processing systems*, 2024.

Kevin P. Murphy. *Probabilistic machine learning: an introduction*. MIT Press, 2022.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International conference on artificial intelligence and statistics*, 2018.

Eshaan Nichani, Jason D. Lee, and Alberto Bietti. Understanding factual recall in transformers via associative memories. In *International conference on learning representations*, 2025.

Andrei Liviu Nicolicioiu, Mohammad Pezeshki, and Aaron Courville. On-policy self-distillation with sampled demonstrations reduces output diversity. *arXiv preprint arXiv:2606.26091*, 2026.

Team OLMo. 2 OLMo 2 Furious. *Conference on language modeling*, 2025.

Yury Polyanskiy and Yihong Wu. *Information theory: From coding to learning*. Cambridge University Press, 2025.

Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in neural information processing systems*, 2017.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. In *International conference on machine learning*, 2024.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International conference on learning representations*, 2016.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3), 1948.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Jacob Mitchell Springer, Madhu Advani, Lukas Aichberger, Arwen Bradley, Eran Malach, Omid Saremi, Sinead Williamson, Preetum Nakkiran, Etai Littwin, and Aditi Raghunathan. Annotations mitigate post-training mode collapse. In *International conference on machine learning*, 2026.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Thinking Machines Lab. On-policy distillation. `https://thinkingmachines.ai/blog/on-policy-distillation/`, 2025.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. `https://github.com/huggingface/trl`, 2020.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and trends in machine learning*, 1(1–2):1–305, 2008.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: high-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in neural information processing systems*, 2025.

Peter West and Christopher Potts. Base models beat aligned models at randomness and creativity. In *Conference on language modeling*, 2025.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3–4):229–256, 1992.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. MiMo-V2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized sampling: how to mitigate mode collapse and unlock LLM diversity. *arXiv preprint arXiv:2510.01171*, 2025.

Anhao Zhao, Haoran Xin, Yingqi Fan, Junlong Tong, Wenjie Li, and Xiaoyu Shen. Decoupling KL and trajectories: a unified perspective for SFT, DAgger, offline RL, and OPD in LLM distillation. *arXiv preprint arXiv:2605.16826*, 2026a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: on-policy self-distillation for large language models. In *International conference on machine learning*, 2026b.

Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? Dynamics, curricula and hallucinations. In *Proceedings of the 2nd conference on language modeling*, 2025.

Nicolas Zucchet, Hyun Dong Lee, and Scott Linderman. Language models suffer from a curse of ambiguity. *arXiv preprint arXiv:2608.15448*, 2026.

Appendix

Table of contents

Notation **15**

A The distillation framework **16**
A.1 Token- and sequence-level objectives and where gradients are stopped 16
A.1.1 The different objectives 16
A.1.2 How the different objectives and their gradients compare 16
A.1.3 What the stop-gradients used in practice ignore 17
A.2 Practical evaluation of the token-level divergence 18
A.2.1 Sampled-token evaluation 18
A.2.2 Top-k restrictions: renormalization and tail bucket 19
A.3 The divergence families we consider 19

B Theory **21**
B.1 Divergence choice does not matter around optimality 21
B.2 Where the entropy gap changes sign 21
B.3 The forward KL inflates entropy 21
B.4 Reverse KL: a flatness statement 23
B.5 Reverse KL inflates entropy in the diffuse high-load limit 24
B.6 Additional results on top-k 25
B.7 Setup of the toy model experiments 27

C Additional experimental results **28**
C.1 The identity across model families and pretraining stages 28
C.2 Additional on- vs. off-policy distillation results 28
C.3 Setup of the on- vs. off-policy distillation experiments 29
C.4 Accuracy vs. entropy for self-distillation on the Chemistry dataset 31
C.5 Additional self-distillation experiments on GSM8K 31
C.6 Hyperparameters of the self-distillation experiments 34

NOTATION

Symbol	Meaning
Distillation framework (Section 2)	
x	prompt, sampled from a dataset
y, y_t, T	completion of the prompt, its t-th token and its length
$y_{<t}$	prefix of the completion before position t
$i = (x, y_{<t})$	context used to compare the student and the teacher
d	vocabulary size
$\pi_\theta(\cdot \mid i)$	student distribution over the next token
$\pi^*(\cdot \mid i)$	teacher distribution over the next token
$p(\cdot \mid x)$	data-generating process the completions are sampled from
ℓ	token-level objective, a divergence between next-token distributions
$\mathcal{L}(\theta)$	total loss, ℓ summed over tokens and averaged over completions
$z(i) \in \mathbb{R}^d$	logits of the student in context i
$\mathbb{E}_i$	average over contexts, one per token
$H(\pi)$	Shannon entropy of a next-token distribution
$\bar{H}(\pi)$	average entropy over contexts, per token
$D_{\mathrm{KL}}(a \parallel b)$	Kullback–Leibler divergence, forward in a and reverse in b
$D_f(a \parallel b)$	f-divergence of a from the reference distribution b
g_{seq}	prefix-visitation gradient the stop-gradient drops
Divergences and entropy (Section 3)	
α	interpolation weight of the generalized Jensen–Shannon family
JS_α	generalized Jensen–Shannon divergence
$m_\alpha = \alpha\pi^* + (1-\alpha)\pi_\theta$	mixture that the family compares both models to
$\pi_\theta^{\mathrm{for}}$, $\pi_\theta^{\mathrm{rev}}$	student at a local optimum of the forward and reverse KL objectives
k	number of tokens kept by a top-k restriction
$g(i)$	mean-logit mismatch between the student and the teacher
$\Delta(i)$	excess surprisal of the student's samples under the teacher
Toy model (Section 3.1)	
h	embedding dimension
n	number of contexts
$e_i \in \mathbb{R}^h$	random embedding of context i
$W \in \mathbb{R}^{d \times h}$	weight matrix of the student's linear head
$\exp(H(\pi^*))$	perplexity of the teacher, its effective support size
Diffuse high-load limit (Appendix B.5)	
$E \in \mathbb{R}^{n \times h}$	embedding matrix, stacking the $e_i^\top$
P_E	orthogonal projector onto the column space of E
u	uniform distribution over the vocabulary
β	inverse teacher temperature, the teacher being diffuse as $\beta \to 0$
$z^*(i)$, $z_\theta(i)$	centered teacher and student logits in context i, stacked into Z^*, Z_θ

A THE DISTILLATION FRAMEWORK

Section 2 introduces the token-level distillation objective of Equation 1 and states that writing the objective at the token level is what decouples the data-generating process p from the divergence ℓ. This appendix makes that claim precise. We first define the sequence-level objective, the token-level one and the surrogate that implementations actually differentiate, and compare them (Appendix A.1). We then discuss how the token-level objective is evaluated in practice (Appendix A.2) and define the divergence families we sweep over (Appendix A.3).

A.1 TOKEN- AND SEQUENCE-LEVEL OBJECTIVES AND WHERE GRADIENTS ARE STOPPED

There are a few technical details in how distillation methods are defined, mostly when the completions are sampled from the student, that have important consequences. Here, we emphasize the difference between token- and sequence-level objectives, and review how an implementation detail, the stop-gradient, alters the objective that is actually optimized.

A.1.1 THE DIFFERENT OBJECTIVES

Definition 1 (Sequence-level objective)**.** The sequence-level objective compares the student and teacher distributions over entire sequences,

$$\mathcal{L}_{\text{seq}}(\theta) = \mathbb{E}_x \left[\ell_{\text{seq}} \left(\pi_\theta(\cdot \mid x), \pi^*(\cdot \mid x) \right) \right], \tag{8}$$

where ℓ_{seq} is a divergence between the two probability distributions over the full sequence space. For the reverse KL divergence, $\ell_{\text{seq}} = D_{\text{KL}}(\pi_\theta(\cdot \mid x) \parallel \pi^*(\cdot \mid x))$, which hides an expectation over student-generated sequences. The forward KL divergence instead hides an expectation over teacher-generated sequences.

Definition 2 (Token-level objective)**.** The token-level objective is that of Equation 1,

$$\mathcal{L}_{\text{tok}}(\theta) = \mathbb{E}_x \mathbb{E}_{y \sim p(\cdot|x)} \left[\sum_{t=1}^{T} \ell_{\text{tok}} \left(\pi_\theta(\cdot \mid x, y_{<t}), \pi^*(\cdot \mid x, y_{<t}) \right) \right], \tag{9}$$

with ℓ_{tok} a divergence between the token-level distributions in context $(x, y_{<t})$ and p the data-generating process from which the sequences are sampled.

Definition 3 (Surrogate token-level objective)**.** The surrogate token-level objective ignores the potential dependency of the data-generating process on the parameters:

$$\mathcal{L}_{\text{sur-tok}}(\theta) = \mathbb{E}_x \mathbb{E}_{y \sim \text{stop-gradient}[p](\cdot|x)} \left[\sum_{t=1}^{T} \ell_{\text{tok}} \left(\pi_\theta(\cdot \mid x, y_{<t}), \pi^*(\cdot \mid x, y_{<t}) \right) \right]. \tag{10}$$

Its gradient is

$$\nabla_\theta \mathcal{L}_{\text{sur-tok}} = \mathbb{E}_x \mathbb{E}_{y \sim p} \left[\sum_{t} \nabla_\theta \ell_{\text{tok}}(\pi_\theta(\cdot \mid x, y_{<t}), \pi^*(\cdot \mid x, y_{<t})) \right], \tag{11}$$

and this is what most papers use in practice.

Remark 1. When $p = \pi^*$, the data-generating process does not depend on θ so the stop-gradient has no effect and $\mathcal{L}_{\text{sur-tok}}$ and $\mathcal{L}_{\text{tok}}$ have the same gradients. The surrogate differs from the token-level objective only when sampling on-policy, i.e. $p = \pi_\theta$, or more generally when the sampling distribution depends on the student.

A.1.2 HOW THE DIFFERENT OBJECTIVES AND THEIR GRADIENTS COMPARE

Fact 1 (The token- and sequence-level objectives can be equal)**.** Take $p = \pi_\theta$ with ℓ_{tok} and ℓ_{seq} the reverse KL divergence, or $p = \pi^*$ with ℓ_{tok} and ℓ_{seq} the forward KL divergence. Then $\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{seq}}$, and in particular $\nabla_\theta \mathcal{L}_{\text{tok}} = \nabla_\theta \mathcal{L}_{\text{seq}}$.

Proof. The proof relies on the chain rule of the KL divergence, which we state for general autoregressive distributions before applying it to the setting of interest. Let $a(\cdot \mid x)$ and $b(\cdot \mid x)$ be two distributions. Their sequence-level log-ratio decomposes token by token,

$$\log \frac{a(y \mid x)}{b(y \mid x)} = \sum_{t=1}^{T} \log \frac{a(y_t \mid x, y_{<t})}{b(y_t \mid x, y_{<t})}. \tag{12}$$

Taking the expectation under $y \sim a(\cdot \mid x)$ and exchanging it with the finite sum,

$$D_{\mathrm{KL}}(a(\cdot \mid x) \parallel b(\cdot \mid x)) = \sum_{t=1}^{T} \mathbb{E}_{y \sim a(\cdot|x)} \left[\log \frac{a(y_t \mid x, y_{<t})}{b(y_t \mid x, y_{<t})} \right]. \tag{13}$$

The t-th summand depends on y only through the prefix $y_{<t}$ and the current token y_t, so the future tokens $y_{>t}$ marginalize out (the tower property) and the expectation becomes

$$\mathbb{E}_{y \sim a(\cdot|x)} \left[\log \frac{a(y_t \mid x, y_{<t})}{b(y_t \mid x, y_{<t})} \right] = \mathbb{E}_{y_{<t} \sim a(\cdot|x)} \, \mathbb{E}_{y_t \sim a(\cdot|x, y_{<t})} \left[\log \frac{a(y_t \mid x, y_{<t})}{b(y_t \mid x, y_{<t})} \right] \tag{14}$$

$$= \mathbb{E}_{y_{<t} \sim a(\cdot|x)} \left[D_{\mathrm{KL}} \left(a(\cdot \mid x, y_{<t}) \parallel b(\cdot \mid x, y_{<t}) \right) \right], \tag{15}$$

the inner expectation being the KL divergence between the token-level distributions in context $(x, y_{<t})$. Summing over t and collecting the per-step expectations back into a single expectation over $y \sim a(\cdot \mid x)$,

$$D_{\mathrm{KL}}(a(\cdot \mid x) \parallel b(\cdot \mid x)) = \mathbb{E}_{y \sim a(\cdot|x)} \left[\sum_{t=1}^{T} D_{\mathrm{KL}} \left(a(\cdot \mid x, y_{<t}) \parallel b(\cdot \mid x, y_{<t}) \right) \right]. \tag{16}$$

This has exactly the same form as the token-level objective of Definition 2 with the data sampled from a and ℓ_{tok} the per-token KL divergence between a and b. The two subcases follow by choosing (a, b).

- Taking $a = \pi_\theta$ and $b = \pi^*$ gives the reverse KL divergence with $p = a = \pi_\theta$.
- Taking $a = \pi^*$ and $b = \pi_\theta$ gives the forward KL divergence with $p = a = \pi^*$.

In both, the outer expectation samples the prefixes from the first argument of the divergence, which is precisely p. Equality of the objectives implies equality of their gradients, as there are no stop-gradients or additional implicit parameters held constant here. □

Remark 2. Because the chain rule samples the prefixes from the first argument of the divergence, deriving an objective at the sequence level fixes the data-generating process at the token level: the reverse KL divergence forces $p = \pi_\theta$ and the forward KL divergence forces $p = \pi^*$. This is why we argued that sequence-level objectives couple ℓ_{tok} to p in the main text. Conversely, nothing couples the two once the objective is written at the token level from the start, which is what lets us sweep the divergence at a fixed data-generating process in Section 3.

A.1.3 What the stop-gradients used in practice ignore

In practice, the surrogate objective is typically used. It follows that the resulting gradient is a biased estimate of that of the actual objective.

Fact 2 (The token-level gradient and its surrogate)**.** Write $\ell_t(y, \theta) = \ell_{\mathrm{tok}}(\pi_\theta(\cdot \mid x, y_{<t}), \pi^*(\cdot \mid x, y_{<t}))$ for the per-token loss along a sequence. The token-level gradient decomposes as

$$\nabla_\theta \mathcal{L}_{\mathrm{tok}} = \nabla_\theta \mathcal{L}_{\text{sur-tok}} + g_{\mathrm{seq}}, \tag{17}$$

with the visitation gradient in its causal form

$$g_{\mathrm{seq}} = \mathbb{E}_x \mathbb{E}_{y \sim p} \left[\sum_{s=1}^{T} \nabla_\theta \log p(y_s \mid x, y_{<s}) \sum_{t=s}^{T} \ell_t(y, \theta) \right], \tag{18}$$

each token weighted by its cost-to-go $\sum_{t=s}^{T} \ell_t$. When $p = \pi^*$, $g_{\mathrm{seq}} = 0$ and the three gradients coincide. When $p = \pi_\theta$ it is generally nonzero, and only then does the surrogate the algorithm optimizes depart from the true gradient of the objective.

Proof. Write $L(y,\theta) = \sum_t \ell_t(y,\theta)$ so that $\mathcal{L}_{\text{tok}} = \mathbb{E}_x\mathbb{E}_{y\sim p}[L]$, and differentiate the sampled expectation with the score-function identity,

$$\nabla_\theta \mathbb{E}_{y\sim p}[L] = \mathbb{E}_{y\sim p}\left[\nabla_\theta L\right] + \mathbb{E}_{y\sim p}\left[L\nabla_\theta \log p(y \mid x)\right], \tag{19}$$

using $\nabla_\theta p = p\,\nabla_\theta \log p$. The first term differentiates only the losses, at fixed sampling distribution, and is $\nabla_\theta \mathcal{L}_{\text{sur-tok}}$ of Definition 3; the second is the visitation gradient. When $p = \pi^*$ does not depend on θ, this score term is absent and $g_{\text{seq}} = 0$. It remains to bring the visitation gradient to the causal form of the statement. The score factors over tokens,

$$\nabla_\theta \log p(y \mid x) = \sum_{s=1}^{T} \nabla_\theta \log p(y_s \mid x, y_{<s}), \tag{20}$$

and the losses ℓ_t with $t < s$ do not depend on the token drawn at step s given its prefix, so, conditioning on $y_{<s}$ and using $\mathbb{E}_{y_s\sim p(\cdot|x,y_{<s})}[\nabla_\theta \log p(y_s \mid x, y_{<s})] = 0$, their cross terms vanish in expectation. Only the cost-to-go $\sum_{t=s}^{T} \ell_t$ survives against the step-s score, giving the causal form

$$g_{\text{seq}} = \mathbb{E}_x\mathbb{E}_{y\sim p}\left[\sum_{s=1}^{T} \nabla_\theta \log p(y_s \mid x, y_{<s}) \sum_{t=s}^{T} \ell_t(y,\theta)\right]. \tag{21}$$

□

Remark 3. The per-token losses ℓ_t are per-step costs, and g_{seq} is the credit assigned to each token for its effect on the future costs it steers the sequence towards. Keeping it in full is the undiscounted, full-horizon policy gradient, which corresponds to a discount factor of $\gamma = 1$; the surrogate gradient drops it entirely, crediting each token only for its immediate cost, that is the discount factor is taken to be $\gamma = 0$ (Thinking Machines Lab, 2025). The myopic token-level algorithm used in practice is therefore the surrogate, not the sequence-level objective: the two objectives are equal (Fact 1), and it is only the gradient that is truncated. This is why the surrogate dynamics do not in general follow the gradient of any well-defined objective (Czarnecki et al., 2019), the field $\nabla_\theta \mathcal{L}_{\text{sur-tok}}$ being non-conservative once $p = \pi_\theta$ moves with θ.

Remark 4. Dropping g_{seq} is not only a modeling choice and can be a practical necessity. As a score-function estimator over full rollouts, the gradient g_{seq} is high-variance and expensive to estimate, a difficulty at the heart of reinforcement learning with policy gradients (Williams, 1992; Schulman et al., 2016). It is also the term that makes the student account for how its own choices reshape the sequence distribution on which it is later scored. This is consistent with the reward hacking observed when solely optimizing the sequence-level reverse KL divergence (Gu et al., 2024).

Remark 5. This is the observation that lets the rest of the paper ignore the sequence-modeling aspects of distillation in our toy model of Section 3.1. Once g_{seq} is dropped, every gradient the algorithm takes is an average of per-context gradients of ℓ_{tok}, and the contexts only enter through the distribution p that generates them. The toy model of Section 3.1 keeps exactly this structure, a set of contexts and a per-context divergence, and discards the autoregressive coupling between them.

A.2 Practical evaluation of the token-level divergence

Evaluating the token-level divergence ℓ in Equation 1 over the full vocabulary can be expensive. Two strategies reduce this cost: estimating the divergence from a single sampled token, or restricting the sum to a top-k subset of the vocabulary.

A.2.1 Sampled-token evaluation

Instead of summing over the vocabulary, one can estimate ℓ_t from a single sampled token, reducing the cost to a single teacher probability per context. This is the only option when the teacher is implicit, as in next-token cross-entropy during pretraining, but is also used with explicit teachers for efficiency. The gradient of this estimate needs care when the sample comes from the student. We illustrate with the reverse KL, where $y_t \sim \pi_\theta$; the same applies to the student branch of the generalized Jensen–Shannon divergence but not to the forward KL, which samples from the teacher.

Naively differentiating the sampled log-ratio at fixed y_t gives

$$\nabla_\theta \log \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi^*(y_t \mid x, y_{<t})} = \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}), \tag{22}$$

since the teacher does not depend on θ, and its expected value vanishes as

$$\mathbb{E}_{y_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})] = 0, \tag{23}$$

whereas $\nabla_\theta \ell_t$ is generally nonzero, because this estimate ignores that the distribution from which y_t is drawn itself depends on θ. Accounting for it through the score-function identity

$$\nabla_\theta \mathbb{E}_{y_t \sim \pi_\theta}[f] = \mathbb{E}_{y_t \sim \pi_\theta}[f \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) + \nabla_\theta f] \tag{24}$$

with f the log-ratio, the integrand term $\mathbb{E}_{y_t \sim \pi_\theta}[\nabla_\theta f]$ is exactly the vanishing naive estimate above, so only the score-function term remains,

$$\nabla_\theta \ell_t = \mathbb{E}_{y_t \sim \pi_\theta}\left[\log \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi^*(y_t \mid x, y_{<t})} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})\right]. \tag{25}$$

This is a second score function, taken at the output token level instead of the sequence level as in the prefix-visitation gradient g_{seq} of Fact 2.

Practical implementations. The on-policy distillation recipe of Thinking Machines Lab (2025) combines this approximation with the one of Remark 3. It estimates ℓ_t from the sampled token, using $-\text{stop-grad}[\log(\pi_\theta(y_t \mid x, y_{<t})/\pi^*(y_t \mid x, y_{<t}))]$ as a per-token advantage inside a standard policy-gradient pipeline, and it truncates the cost-to-go to the immediate term ($\gamma = 0$). The first is the Monte Carlo approximation of the local loss gradient over the action space discussed above. The second is the truncation of the sequence-level gradient over time that Appendix A.1 analyzes. The two are independent of each other.

A.2.2 Top-k restrictions: renormalization and tail bucket

When the teacher's full probability vector is available, it is common to restrict the divergence to a subset $S \subseteq \{1, \ldots, d\}$ of the vocabulary, typically the top-k tokens of the teacher or the student, for compute and bandwidth efficiency. Given such a subset, there are two standard ways to handle the probability mass outside S.

Renormalization. Both distributions are renormalized over S and the divergence is computed between the restricted distributions:

$$\ell_{\text{renorm}}(\pi_\theta, \pi^*; S) = \ell\left(\frac{\pi_\theta|_S}{M_\theta}, \frac{\pi^*|_S}{M^*}\right), \tag{26}$$

where $M_\theta = \sum_{j \in S} \pi_\theta(j)$ and $M^* = \sum_{j \in S} \pi^*(j)$ are the total masses on S, and $\pi|_S$ denotes the restriction of π to S. This discards the tail entirely: the gradient with respect to logits outside S is zero, and the student receives no signal about how much total mass to place on the top-k set.

Tail bucket. All mass outside S is lumped into a single extra category and the divergence is computed over $k+1$ tokens:

$$\ell_{\text{tail}}(\pi_\theta, \pi^*; S) = \ell(\tilde{\pi}_\theta, \tilde{\pi}^*), \quad \text{with} \quad \tilde{\pi}(j) = \begin{cases} \pi(j) & j \in S, \\ 1 - \sum_{j' \in S} \pi(j') & j = \text{tail}. \end{cases} \tag{27}$$

This preserves the total-mass constraint: the student is trained to match the teacher's tail mass as a lump sum, which provides one additional degree of freedom compared to renormalization.

As Section 3.6 and Appendix B.6 show, the choice between the two has a direct effect on the entropy of the student.

A.3 The divergence families we consider

Forward and reverse KL are the two main focuses of our analysis, but the transition between them is itself informative, so we also train on a family that interpolates between the two. We use the generalized Jensen–Shannon divergence, the objective family of Agarwal et al. (2024) also used in other works (e.g., Hübotter et al., 2026), which compares both models to their mixture

$$m_\alpha = \alpha \pi^* + (1 - \alpha)\pi_\theta \tag{28}$$

through

$$\mathrm{JS}_\alpha(\pi^*, \pi_\theta) = \alpha D_{\mathrm{KL}}(\pi^* \parallel m_\alpha) + (1-\alpha) D_{\mathrm{KL}}(\pi_\theta \parallel m_\alpha), \tag{29}$$

with $\alpha \in [0, 1]$. The two KL divergences are recovered in the limits, after dividing by the prefactor that vanishes there.

As $\alpha \to 0$ the mixture tends to π_θ, so

$$D_{\mathrm{KL}}(\pi_\theta \parallel m_\alpha) = O(\alpha^2) \tag{30}$$

and

$$\frac{\mathrm{JS}_\alpha}{\alpha} \to D_{\mathrm{KL}}(\pi^* \parallel \pi_\theta), \tag{31}$$

which is the forward KL.

As $\alpha \to 1$ the mixture tends to π^* and symmetrically

$$\frac{\mathrm{JS}_\alpha}{1-\alpha} \to D_{\mathrm{KL}}(\pi_\theta \parallel \pi^*), \tag{32}$$

the reverse KL.

Both endpoints of Equation 29 are identically zero, so only the interior of the family has a gradient; we take the divergence at $\alpha = 0$ and $\alpha = 1$ to be its normalized limits, that is the exact forward and reverse KL, which makes the whole interval trainable, matching the implementations of Agarwal et al. (2024) and von Werra et al. (2020). This scale mismatch is not a problem to compare learning dynamics and their observables (like entropy), as Adam (Kingma & Ba, 2015) is almost invariant to the overall scale of the loss (Balles & Hennig, 2018).

f-divergences. Throughout the theory, we will sometimes use results that hold for a broad class of divergences, f-divergences, which includes the generalized Jensen–Shannon family and its two endpoints. We introduce them here for completeness and make the connection to the generalized Jensen–Shannon family explicit. Given a convex function f with $f(1) = 0$, the f-divergence of a distribution a from a reference distribution b is

$$D_f(a \parallel b) = \sum_j b(j) f\left(\frac{a(j)}{b(j)}\right). \tag{33}$$

The reverse KL is recovered with $f(t) = t \log t$, the forward KL with $f(t) = -\log t$, and the generalized Jensen–Shannon divergence with

$$f_\alpha(t) = -\alpha \log\big(\alpha + (1-\alpha)t\big) + (1-\alpha) t \log \frac{t}{\alpha + (1-\alpha)t}. \tag{34}$$

What will matter to us is that all members of this class that are twice differentiable at 1 share the same local geometry around $a = b$, namely the Fisher metric scaled by the curvature $f''(1)$, which is the statement of Appendix B.1 and what makes the diffuse limit of Appendix B.5 divergence-independent. That curvature is $f''(1) = 1$ for both KL directions and $f''_\alpha(1) = \alpha(1-\alpha)$ for the generalized Jensen–Shannon divergence, which vanishes at the two endpoints and is brought back to 1 by the $1/\alpha$ and $1/(1-\alpha)$ normalizations above.

B THEORY

B.1 DIVERGENCE CHOICE DOES NOT MATTER AROUND OPTIMALITY

We now show the claim of Section 3.2: at a fixed data-generating process, the forward and reverse KL objectives have the same gradient to first order around $\pi_\theta = \pi^*$. The statement is per context, so fix a context i and write π_θ and π^* for the two next-token distributions there. The forward KL gradient is

$$\nabla_\theta D_{\mathrm{KL}}(\pi^* \parallel \pi_\theta) = -\sum_j \pi^*(j) \nabla_\theta \log \pi_\theta(j), \tag{35}$$

the teacher being independent of θ. For the reverse KL, differentiating the product and using $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$ gives

$$\nabla_\theta D_{\mathrm{KL}}(\pi_\theta \parallel \pi^*) = \sum_j \nabla_\theta \pi_\theta(j) \log\left(\frac{\pi_\theta(j)}{\pi^*(j)}\right) + \sum_j \pi_\theta(j) \nabla_\theta \log \pi_\theta(j) \tag{36}$$

$$= \sum_j \pi_\theta(j) \nabla_\theta \log \pi_\theta(j) \log\left(\frac{\pi_\theta(j)}{\pi^*(j)}\right), \tag{37}$$

the second term vanishing by the vocabulary-level score identity of Equation 23. Around optimality the log ratio linearizes as

$$\log\left(\frac{\pi_\theta(j)}{\pi^*(j)}\right) = -\log\left(\frac{\pi^*(j)}{\pi_\theta(j)}\right) \approx 1 - \frac{\pi^*(j)}{\pi_\theta(j)}, \tag{38}$$

and substituting it, the constant term vanishes by the score identity again, leaving

$$\nabla_\theta D_{\mathrm{KL}}(\pi_\theta \parallel \pi^*) \approx -\sum_j \pi^*(j) \nabla_\theta \log \pi_\theta(j) = \nabla_\theta D_{\mathrm{KL}}(\pi^* \parallel \pi_\theta). \tag{39}$$

Averaging the two sides over contexts sampled from any data-generating process p preserves the equality, as p only weights the contexts and weights them identically on both sides. This is why the statement is independent of p: the surrogate gradients of Definition 3 for the forward and the reverse KL agree to first order whatever the data-generating process is, as long as it is held fixed. Near optimality, the choice between the two divergences is therefore immaterial, and whatever difference remains between two distillation runs has to come from the data they are trained on.

This equivalence is well known: it is a classical result from information geometry that all suitably standardized f-divergences, of which forward and reverse KL are special cases, agree up to second order around $\pi_\theta = \pi^*$, with the shared local quadratic form given by the Fisher information metric (Amari, 2016; Polyanskiy & Wu, 2025). The whole generalized Jensen–Shannon family of Appendix A.3 is therefore also equivalent to its two endpoints in this regime.

Importantly, the expansion above requires the perturbation $\pi_\theta - \pi^*$ to be small relative to π^* on every token, that is $|\log(\pi_\theta(j)/\pi^*(j))|$ uniformly small. This is a much stronger requirement than a small average KL divergence, since the KL weights log ratios by probability mass and is therefore blind to large relative perturbations on low-probability tokens, which is precisely where the gradients of the two objectives differ.

B.2 WHERE THE ENTROPY GAP CHANGES SIGN

Figure 2 varies the number of contexts, the teacher perplexity and the interpolation weight one at a time. Figure 7 repeats this experiment in the plane of number of contexts and teacher perplexity, over 5 different generalized Jensen–Shannon divergence weights α. The results highlight that even though sweeping individual parameters yields sharp phase transitions, the boundary of the phase transition evolves much more smoothly.

B.3 THE FORWARD KL INFLATES ENTROPY

This appendix gives the formal backing for the claim in Section 3.3 that the forward KL inflates entropy. Assume the student family is closed under temperature scaling of the logits, which holds

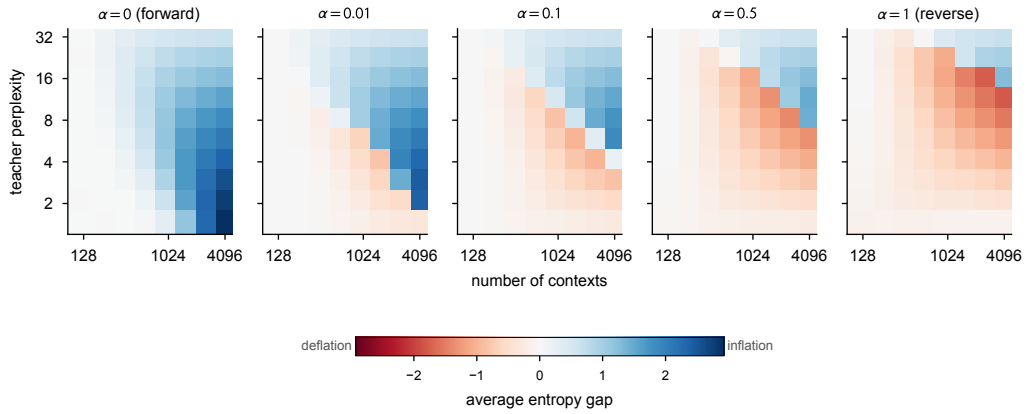

Figure 7: **The entropy gap phase transition moves with the divergence.** Average entropy gap $\bar{H}(\pi_\theta) - \bar{H}(\pi^*)$ at convergence over the plane of teacher perplexity and number of contexts in the setup of Section 3, at five weights α of the generalized Jensen–Shannon family, with the forward KL on the left and the reverse KL on the right. Each cell is its own configuration, averaged over 5 seeds, and color indicates the entropy gap. The forward KL always inflates entropy and increasing α progressively moves the diagonal boundary, that is the task needs to be harder for the student to inflate entropy.

for any softmax model whose final linear layer can be rescaled, including standard language model heads. As above we write i for a context, $\pi_\theta^{\text{for}}(\cdot \mid i)$ for a student trained with the forward KL and $z(i)$ for its logits there, and take the contexts to be drawn from a data-generating process p that is held fixed, as the surrogate objective of Definition 3 does.

Define $g(i) = \mathbb{E}_{\pi_\theta^{\text{for}}}[z(i)] - \mathbb{E}_{\pi^*}[z(i)]$, the mean-logit mismatch in context i, which is also the derivative of the loss in i when sharpening the logits along the temperature direction. To see this, perturb the logits along the temperature direction, $z(i, \varepsilon) = (1+\varepsilon)z(i)$, so that the loss in i becomes

$$\ell^{\text{for}}(i, \varepsilon) = -(1 + \varepsilon)\mathbb{E}_{\pi^*}[z(i)] + \log Z(i, \varepsilon) \tag{40}$$

up to a term that does not depend on the student, with $Z(i, \varepsilon) = \sum_j \exp(z(i, \varepsilon)_j)$ the partition function of the perturbed student. Using $\nabla_z \log Z = \pi_\theta^{(\varepsilon)}$, the chain rule gives

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\ell^{\text{for}}(i, \varepsilon) = -\mathbb{E}_{\pi^*}[z(i)] + \mathbb{E}_{\pi_\theta^{(\varepsilon)}}[z(i)], \tag{41}$$

which at $\varepsilon = 0$ reduces to

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\ell^{\text{for}}(i, 0) = \mathbb{E}_{\pi_\theta^{\text{for}}}[z(i)] - \mathbb{E}_{\pi^*}[z(i)] = g(i). \tag{42}$$

At any interior stationary point of the objective, the derivative of the total loss along the temperature direction must vanish, which is exactly

$$\mathbb{E}_i[g(i)] = 0, \tag{43}$$

the expectation being over contexts drawn from p, each token counting once as in Equation 1.

The difference in entropy between the student and the teacher can be rewritten to make its dependence on g explicit:

$$\begin{aligned} H(\pi_\theta^{\text{for}}) - H(\pi^*) &= -\mathbb{E}_{\pi_\theta^{\text{for}}}\left[\log \pi_\theta^{\text{for}}\right] + \mathbb{E}_{\pi^*}\left[\log \pi^*\right] && (44)\\ &= \mathbb{E}_{\pi^*}\left[\log \pi^* - \log \pi_\theta^{\text{for}}\right] + \mathbb{E}_{\pi^*}\left[\log \pi_\theta^{\text{for}}\right] - \mathbb{E}_{\pi_\theta^{\text{for}}}\left[\log \pi_\theta^{\text{for}}\right] && (45)\\ &= D_{\text{KL}}(\pi^* \,\|\, \pi_\theta^{\text{for}}) + \mathbb{E}_{\pi^*}\left[z(i) - \log Z(i)\right] - \mathbb{E}_{\pi_\theta^{\text{for}}}\left[z(i) - \log Z(i)\right] && (46)\\ &= D_{\text{KL}}(\pi^* \,\|\, \pi_\theta^{\text{for}}) - g(i), && (47) \end{aligned}$$

where the last equality comes from the fact that $\log Z$ is constant in the token and therefore cancels. Leveraging the stationarity condition on g (Equation 43) yields

$$\mathbb{E}_i\left[H(\pi_\theta^{\text{for}})\right] = \mathbb{E}_i\left[H(\pi^*)\right] + \mathbb{E}_i\left[D_{\text{KL}}(\pi^* \parallel \pi_\theta^{\text{for}})\right]. \tag{48}$$

Note that nothing in the argument constrains p: the identity is a property of the forward KL alone, the data-generating process only setting which contexts the two averages run over. When the teacher is implicit and only accessible through its samples, as in pretraining, the right-hand side is the cross-entropy loss, which gives the form of Equation 6 that Section 4.1 tests.

This is an instance of a classical fact from information geometry: along its temperature direction, the student is a one-parameter exponential family with its own logits as sufficient statistic. Stationarity of the forward KL divergence is moment matching of that statistic (Wainwright & Jordan, 2008), and the entropy of a moment-matched projection exceeds the entropy of its target by exactly their divergence (Amari, 2016).

From average to pointwise inflation. So far, we have shown that entropy increases when averaging over contexts. A natural follow-up question is whether the same holds in each context individually, that is, whether the entropy gap is always positive, or equivalently whether $g(i) < D_{\text{KL}}(\pi^* \parallel \pi_\theta^{\text{for}})$ everywhere. The statement becomes pointwise as soon as the family contains per-context temperature directions, meaning that for every context the perturbation rescaling $z(i)$ while leaving the logits in other contexts unchanged is tangent to the family. This is the case when the model has a learned input-dependent temperature on its logits, and approximately when the body is expressive enough to rescale its last-layer representation in one context without affecting the others. Stationarity along these directions forces $g(i) = 0$ separately in every context, and the identity becomes

$$H(\pi_\theta^{\text{for}}) = H(\pi^*) + D_{\text{KL}}(\pi^* \parallel \pi_\theta^{\text{for}}) \tag{49}$$

in every context: the student entropy is strictly inflated wherever the student fails to match the teacher, by exactly the misfit there. Note that per-context temperature scaling is far weaker than realizability, as a single scalar per context cannot match a full distribution, so the pointwise statement is not vacuous. This average-versus-pointwise structure echoes the calibration literature, where stationarity along a single shared rescaling direction yields guarantees that hold on average over inputs (Clarté et al., 2023), while near-optimality against richer families of input-dependent post-processings yields correspondingly stronger pointwise ones (Błasiok et al., 2023).

B.4 Reverse KL: a flatness statement

A general statement on entropy similar to the one we obtained for the forward KL does not hold for the reverse one. We apply the same recipe, decomposing the misfit and then exploiting stationarity along the temperature direction, but it yields only a flatness statement. Writing π_θ^{rev} for a student trained with the reverse KL and expanding the divergence as

$$D_{\text{KL}}(\pi_\theta^{\text{rev}} \parallel \pi^*) = \mathbb{E}_{\pi_\theta^{\text{rev}}}[\log \pi_\theta^{\text{rev}}] - \mathbb{E}_{\pi_\theta^{\text{rev}}}[\log \pi^*] = \mathbb{E}_{\pi_\theta^{\text{rev}}}[-\log \pi^*] - H(\pi_\theta^{\text{rev}}), \tag{50}$$

adding $H(\pi_\theta^{\text{rev}})$ and subtracting $H(\pi^*)$ and rearranging gives, in every context,

$$H(\pi_\theta^{\text{rev}}) = H(\pi^*) + \Delta(i) - D_{\text{KL}}(\pi_\theta^{\text{rev}} \parallel \pi^*) \quad \text{with} \quad \Delta(i) = \mathbb{E}_{\pi_\theta^{\text{rev}}}[-\log \pi^*] - H(\pi^*). \tag{51}$$

The excess surprisal $\Delta(i)$ measures how much more surprising the student's samples are to the teacher than the teacher's own samples are.

The student is therefore less confident than the teacher exactly when $\Delta(i)$ exceeds the misfit $D_{\text{KL}}(\pi_\theta^{\text{rev}} \parallel \pi^*)$. The temperature-direction argument no longer pins this sign down; it instead yields a flatness statement. Perturbing the logits along the temperature direction, the derivative of the reverse KL loss in context i is

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\ell^{\text{rev}}(i, 0) = \mathrm{Var}_{\pi_\theta^{\text{rev}}}(\log \pi_\theta^{\text{rev}}) - \mathrm{Cov}_{\pi_\theta^{\text{rev}}}(\log \pi_\theta^{\text{rev}}, \log \pi^*). \tag{52}$$

Requiring the total derivative to vanish gives the stationarity condition on average over contexts,

$$\mathbb{E}_i\left[\mathrm{Var}_{\pi_\theta^{\text{rev}}}(\log \pi_\theta^{\text{rev}})\right] = \mathbb{E}_i\left[\mathrm{Cov}_{\pi_\theta^{\text{rev}}}(\log \pi_\theta^{\text{rev}}, \log \pi^*)\right]. \tag{53}$$

Bounding each covariance by Cauchy–Schwarz,

$$\mathrm{Cov}_{\pi_\theta^{\mathrm{rev}}}(\log \pi_\theta^{\mathrm{rev}}, \log \pi^*) \le \sqrt{\mathrm{Var}_{\pi_\theta^{\mathrm{rev}}}(\log \pi_\theta^{\mathrm{rev}})\mathrm{Var}_{\pi_\theta^{\mathrm{rev}}}(\log \pi^*)}, \tag{54}$$

applying Cauchy–Schwarz a second time to the average over contexts, dividing by $\sqrt{\mathbb{E}_i[\mathrm{Var}_{\pi_\theta^{\mathrm{rev}}}(\log \pi_\theta^{\mathrm{rev}})]}$ and squaring gives

$$\mathbb{E}_i\left[\mathrm{Var}_{\pi_\theta^{\mathrm{rev}}}(\log \pi_\theta^{\mathrm{rev}})\right] \le \mathbb{E}_i\left[\mathrm{Var}_{\pi_\theta^{\mathrm{rev}}}(\log \pi^*)\right]. \tag{55}$$

As a result, the same techniques that provided an entropy statement for the forward KL only provide a flatness statement here.

B.5 Reverse KL inflates entropy in the diffuse high-load limit

Empirically (Figure 2), we find that when the number of contexts is large (high load) or the teacher is diffuse (high perplexity), the reverse KL stops being mode-seeking and inflates entropy, as the forward KL does. Here, we expand the distillation objective around a diffuse teacher and show that to leading order, forward KL, reverse KL and the whole generalized Jensen–Shannon family reduce to the same quadratic objective and therefore share the same solution. This is why the top right corner of Figure 7 in which both the teacher perplexity and the number of contexts are largest looks the same for every divergence.

The setup. We work in the toy model of Section 3.1, in which the teacher $\pi^*(\cdot \mid i)$ is drawn independently of the embedding e_i, from a law that is invariant under permutations of the vocabulary. Stack the embeddings into $E \in \mathbb{R}^{n\times h}$, whose i-th row is $e_i^\top$, and write P_E for the orthogonal projector onto its column space in $\mathbb{R}^n$. For simplicity, we assume that $n > h$. Random embeddings are in general position, so $\mathrm{rank}(E) = h$. The student head produces the logit matrix $Z_\theta = EW^\top \in \mathbb{R}^{n\times d}$, so each of its columns is constrained to the h-dimensional column space of E. The ratio h/n is therefore the fraction of the n context directions the student can exactly represent.

The diffuse limit. We make the teacher diffuse, keeping h/n arbitrary. We model it by fixing the teacher logits $z^*(i)$ and scaling them down by a factor $\beta \to 0$, so that the teacher distribution becomes uniform in the limit. Write the teacher and the student in centered logit coordinates,

$$\pi^*(\cdot \mid i) = \mathrm{softmax}\big(\beta z^*(i)\big), \tag{56}$$

$$\pi_\theta(\cdot \mid i) = \mathrm{softmax}\big(z_\theta(i)\big), \tag{57}$$

$$\mathbf{1}^\top z^*(i) = \mathbf{1}^\top z_\theta(i) = 0, \tag{58}$$

with the logits $z^*(i)$ fixed and drawn independently across contexts.

All f-divergences that are twice differentiable at 1 agree to second order around coinciding distributions, and their common quadratic form is the Fisher metric scaled by $f''(1)$ (Amari, 2016); this is the argument we used in Appendix B.1 for the two KL directions. In logit coordinates it reads

$$D_f\big(\mathrm{softmax}(z+\delta) \,\|\, \mathrm{softmax}(z)\big) = \frac{f''(1)}{2}\mathrm{Var}_{\mathrm{softmax}(z)}(\delta) + O(\|\delta\|^3), \tag{59}$$

the variance being over tokens drawn from the reference distribution, the second argument of D_f. The expansion holds as long as the offset δ is small, which here means that all the logits involved are $O(\beta)$: the teacher's by construction, and the student's because we expand around the branch on which it stays that close to uniform. We apply it with the teacher as the reference and $\delta = z_\theta(i) - \beta z^*(i)$. The teacher is itself $O(\beta)$ away from the uniform distribution, so Var_{π^*} can in turn be replaced by Var_u at the price of a further $O(\beta^3)$ term, and for centered logits $\mathrm{Var}_u(\delta) = \|\delta\|^2/d$. Stacking the logits into $Z^*, Z_\theta \in \mathbb{R}^{n\times d}$, this gives

$$\mathbb{E}_i\left[D_f\big(\pi_\theta(\cdot \mid i) \,\|\, \pi^*(\cdot \mid i)\big)\right] = \frac{f''(1)}{2nd}\|Z_\theta - \beta Z^*\|_F^2 + O(\beta^3). \tag{60}$$

Both KL directions are of this form, with π^* as the reference distribution and $f''(1) = 1$, so they have the same minimizer. The same argument applies to the generalized Jensen–Shannon family of Appendix A.3.

Minimizing Equation 60 over the achievable logits $Z_\theta = EW^\top$ is a least-squares projection onto the column space of E, so the optimal student logits are

$$Z_\theta^\star = \beta P_E Z^* + O(\beta^2). \tag{61}$$

The teacher's logits have scale $O(\beta)$, the fitted ones only $O(\beta\sqrt{h/n})$: the entries of Z^* are independent and P_E is a projector of rank h that does not depend on them, so

$$\mathbb{E}_{Z^*}\|P_E Z^*\|_F^2 = \frac{h}{n}\mathbb{E}_{Z^*}\|Z^*\|_F^2. \tag{62}$$

As a result, increasing the number of contexts while keeping the model size fixed, which is what we do in our experiments, makes the student logits smaller relative to the teacher's, and the student distribution closer to uniform.

Entropy inflation. Applying Equation 59 again, this time with the uniform distribution as the reference and combining it with $D_{\mathrm{KL}}(\pi \parallel u) = \log d - H(\pi)$ provides an entropy expansion: a distribution $\mathrm{softmax}(\delta)$ with small centered logits has entropy

$$H(\mathrm{softmax}(\delta)) = \log d - \|\delta\|^2/(2d) + O(\|\delta\|^3). \tag{63}$$

Applying this to the teacher and to the student at the optimum of Equation 61,

$$\bar{H}(\pi^*) = \log d - \frac{\beta^2}{2nd}\|Z^*\|_F^2 + O(\beta^3), \qquad \bar{H}(\pi_\theta) = \log d - \frac{\beta^2}{2nd}\|P_E Z^*\|_F^2 + O(\beta^3), \tag{64}$$

so that

$$\bar{H}(\pi_\theta) - \bar{H}(\pi^*) = \frac{\beta^2}{2nd}\|(I - P_E)Z^*\|_F^2 + O(\beta^3). \tag{65}$$

That is, for small enough β values, the student inflates entropy relative to the teacher.

If we further assume that the numbers of contexts and tokens are large enough that we can replace the teacher-related quantities by their expected values, the entropy gap only depends on the student through the single ratio h/n. Indeed

$$\mathbb{E}_{Z^*}\|(I - P_E)Z^*\|_F^2 = \left(1 - \frac{h}{n}\right)\mathbb{E}_{Z^*}\|Z^*\|_F^2, \tag{66}$$

so that Equation 65 becomes

$$\begin{aligned}\bar{H}(\pi_\theta) - \bar{H}(\pi^*) &= \frac{\beta^2}{2nd}\left(1 - \frac{h}{n}\right)\mathbb{E}_{Z^*}\|Z^*\|_F^2 + O(\beta^3) \\ &= \left(1 - \frac{h}{n}\right)\left(\log d - \bar{H}(\pi^*)\right) + O(\beta^3),\end{aligned} \tag{67}$$

the second line recognizing in the first the teacher's entropy deficit below uniform of Equation 64. It follows that, in the diffuse high-load limit, the gap increases as the number of contexts n increases, and decreases as the perplexity of the teacher increases, consistently with Figure 2b and 2c.

Evaluating Equation 60 at the optimum and comparing it to Equation 65 gives one last identity,

$$\mathbb{E}_i\left[D_f\big(\pi_\theta(\cdot \mid i) \parallel \pi^*(\cdot \mid i)\big)\right] = f''(1)\left(\bar{H}(\pi_\theta) - \bar{H}(\pi^*)\right) + O(\beta^3), \tag{68}$$

so for either KL direction the residual divergence equals the entropy inflation. For the forward KL this is Equation 6, which holds for any number of contexts and any teacher; reverse KL obeys it too in the diffuse limit.

B.6 Additional results on top-k

Figure 3 studies the effect of the different top-k approximations for the forward KL only. The forward KL is the $\alpha = 0$ endpoint of the generalized Jensen–Shannon family. Figure 8 repeats the experiment at five different generalized Jensen–Shannon weights of Figure 7. The vocabulary size is 64, so $k = 64$ is the untruncated objective we use in the rest of the toy model experiments. Renormalizing a top-1 distribution gives a Dirac on one token for both student and teacher, so the divergence is identically zero and no learning occurs; we therefore remove this configuration from all our experiments.

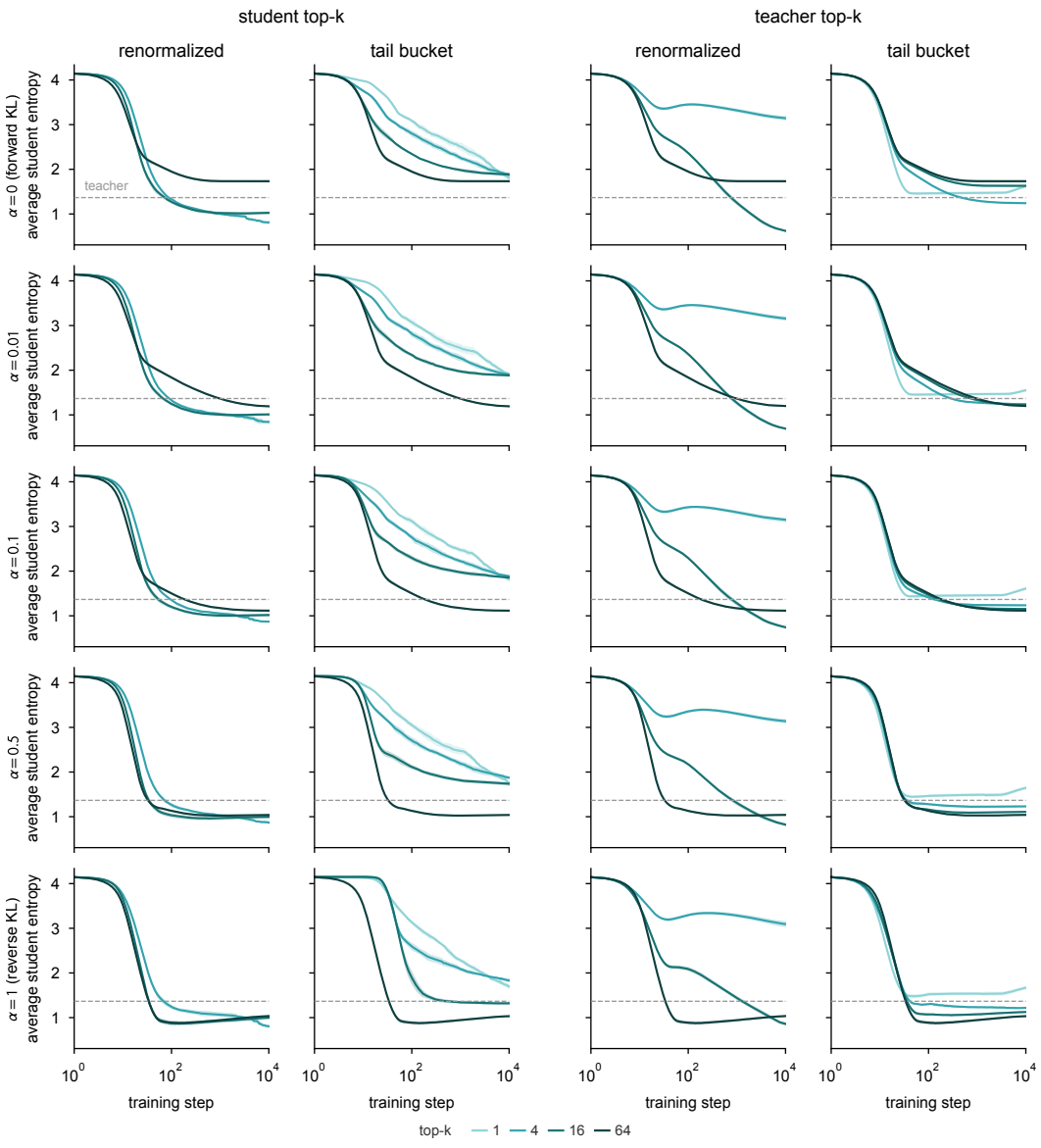

Figure 8: **The top-k approximation across the divergence family.** Average student entropy over training, for the two choices the approximation makes: which model defines the support, i.e. whether we take the student's own top-k or the teacher's, and what happens to the mass left outside the support, either renormalizing both sides over the window or keeping it in one extra tail bucket. Rows are the generalized Jensen–Shannon weight, from the forward KL at $\alpha = 0$ to the reverse KL at $\alpha = 1$. Curves are averaged over 5 seeds $\pm$std, and the dashed line is the teacher entropy.

B.7 SETUP OF THE TOY MODEL EXPERIMENTS

This section details the toy model of Section 3.1 and the simulations of Figures 2, 3, 7 and 8.

Model. The vocabulary has $d = 64$ tokens and the embeddings have dimension $h = 128$. Each of the n contexts is assigned an embedding e_i drawn from a standard Gaussian distribution in $\mathbb{R}^h$ and normalized to unit norm, which stays fixed during training. The student is a linear layer followed by a softmax, $\pi_\theta(\cdot \mid i) = \mathrm{softmax}(We_i + b)$, with $W \in \mathbb{R}^{d \times h}$ initialized with LeCun normal initialization and b initialized at zero. The main text omits the bias b for simplicity.

Teachers. For each context i, we draw a target perplexity P_i and Gaussian logits $\zeta_i \sim \mathcal{N}(0, I_d)$, and set $\pi^*(\cdot \mid i) = \mathrm{softmax}(\zeta_i/\tau_i)$, with the temperature τ_i chosen so that $H(\pi^*(\cdot \mid i)) = \log P_i$. By default, the perplexities are drawn independently and log-uniformly in $[1, 16]$, so that contexts differ both in which tokens the teacher favors and in how diffuse it is. The teacher perplexity sweep of Figure 2c and the heatmaps of Figure 7 instead give the same perplexity to all contexts. All contexts have the same weight in the objective.

Training. Every step uses all contexts, and the student is trained with Adam (Kingma & Ba, 2015) and a constant learning rate of 0.1 for $40{,}000$ steps. All results are averaged over 5 seeds, each of which redraws the embeddings, the teachers and the initialization.

Sweeps. The default number of contexts is $n = 2048$. Figure 2b uses 21 values of n log-spaced between 192 and $67{,}584$, Figure 2c uses 21 perplexities log-spaced between 1 and 32, and Figure 2d uses 21 weights α log-spaced between 0.001 and 0.99. The trajectories of Figure 2e and the heatmaps of Figure 7 use $\alpha \in \{0, 0.01, 0.1, 0.5, 1\}$, the heatmaps over 10 perplexities log-spaced between 1.5 and 32 and 8 numbers of contexts log-spaced between 128 and 4096. The top-k experiments of Figures 3 and 8 use $n = 512$ contexts and $k \in \{1, 4, 16, 64\}$, and Figure 8 uses the same five weights α as Figure 7.

C Additional experimental results

C.1 The identity across model families and pretraining stages

Figure 4 only covers one family of models, OLMo 2 (OLMo, 2025), and only at the end of its pretraining. Figure 9 extends it to the Pythia family of models (Biderman et al., 2023) and to an earlier point of OLMo 2's pretraining, averaging entropy and cross-entropy over about two million tokens per checkpoint.

OLMo 2 is pretrained in two stages, a long first one on a large web-dominated mixture and a short second one on a smaller and higher quality mixture, over which the learning rate is annealed to zero. The light points are the checkpoints at the end of the first stage, measured on the first-stage mixture, and the dark ones the final pretrained models, measured on the second-stage mixture, which is the configuration Figure 4 reports. The pretraining mixture of Figure 4 is therefore the second-stage mixture, the last data the pretrained models were trained on. We draw the same number of tokens from every source of the mixture, but average entropy and cross-entropy at the published proportions of that mixture.

We use the standard Pythia models rather than their deduplicated variants. They are trained in a single stage on the Pile (Gao et al., 2020), whose training stream is released pre-tokenized and in the order the models consumed it, so the tokens we measure on are exactly the ones the models were trained on. Since that run is a single epoch, position in the stream is position in training time, and we sample three chunks of equal size from its beginning, middle and end.

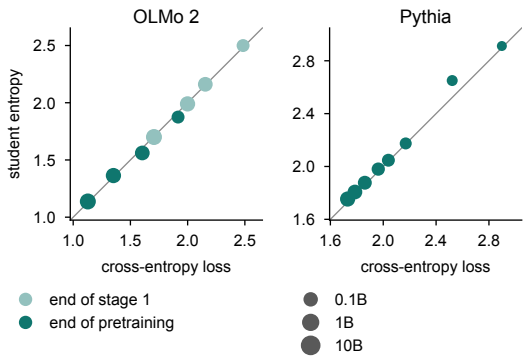

Figure 9: **At the end of pretraining, a model's mean entropy equals its mean cross-entropy.** Similar to Figure 4 (left), for different model families and at different stages of pretraining. (left) OLMo 2 at 1B, 7B, 13B and 32B, at the end of stage 1 of pretraining (light) and at the end of pretraining (dark). (right) the eight Pythia models from 70M to 12B on the Pile (Gao et al., 2020).

C.2 Additional on- vs. off-policy distillation results

Figure 5 only reports the entropy of the student on its own completions. Figure 10 completes it with the entropy of the student on frozen teacher completions, where the contexts cannot move and only the weights can, and with its accuracy. Accuracy is the in-training evaluation, averaged over one sample on 200 MATH500 problems at the evaluation budget of Table 1, and is therefore noisier than the endpoint evaluation.

On frozen teacher completions, forward KL inflates entropy and reverse KL deflates it, and off-policy reverse KL deflates it most. On the student's own completions the split has the same sign, with on-policy forward KL inflating entropy most. Reverse KL slightly improves accuracy over the base student while forward KL, in particular on-policy, degrades it.

The lower accuracy of forward KL may largely be a consequence of the different levels of entropy it reaches rather than of what the student learns. The in-training evaluation samples at temperature 1, and a student with more entropy is more likely to sample a wrong token along a long solution. Consistently, at the lower sampling temperature of the endpoint evaluation (Appendix C.3), all twelve

runs score above the base student on MATH500, between 43.00% and 46.27% against 42.87%. We expect accuracy to vary smoothly with entropy in this regime, so that the sampling temperature, or the other hyperparameters, could likely be adjusted to recover part of this gap. We did not try to: we kept the hyperparameters of Zhao et al. (2026a) fixed across the grid and did not tune them for accuracy, as our goal is to study entropy. These results should therefore only be read as showing that some of these settings improve the accuracy of the student, not as a comparison of the best accuracy each objective can reach.

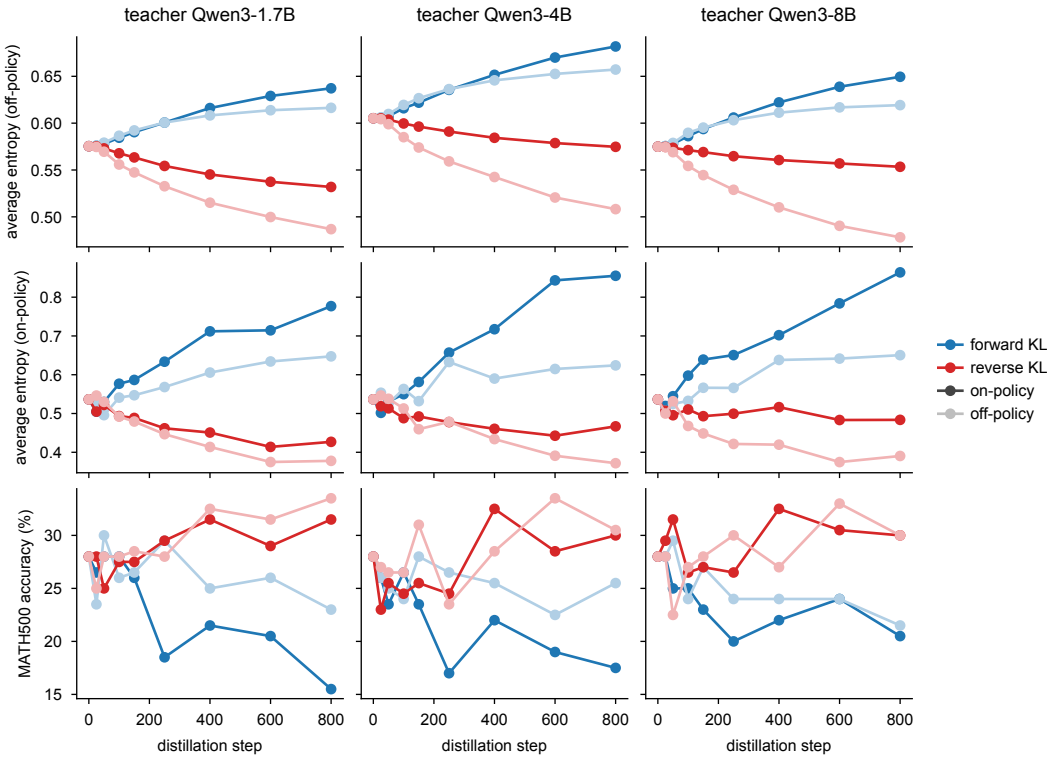

Figure 10: **Teacher-to-student distillation from non-thinking teachers.** Average entropy of the student on frozen teacher completions (top) and on its own completions (middle, as in Figure 5), and MATH500 accuracy (bottom), against distillation step, for teachers of increasing size (left to right). Color is the divergence, with the full color for on-policy and a lighter shade for off-policy. Each curve is a single seed.

C.3 Setup of the on- vs. off-policy distillation experiments

These experiments vary the sampling distribution and the divergence independently, as in Section 2, on a standard reasoning distillation task. Following Zhao et al. (2026a), we distill the Qwen3 instruct models with thinking disabled, at 1.7B, 4B and 8B parameters, into Qwen3-0.6B-Base (Yang et al., 2025), on the math problems of DeepScaleR (Luo et al., 2025). The prefixes on which the token-level divergence is evaluated are either completions sampled from the teacher once and cached before training (off-policy), or completions sampled from the current student at every step (on-policy). The divergence is either the forward or the reverse KL, computed on the full vocabulary at every position of the prefix. Crossing the two prefix sources with the two divergences gives four objectives, and with three teachers the grid consists of twelve runs, all starting from the same base student. Table 1 lists the hyperparameters, which follow Zhao et al. (2026a) except for the choices motivated below.

The prompt frame. A token-level divergence compares the two models on the same prefix, so they have to share a prompt frame, yet they work best in different ones. The base student degrades inside the chat template, from 42.87% to 14.93% accuracy on MATH500 (Lightman et al., 2024), whereas the instruct teachers remain accurate on the bare problem text but almost never stop, as there

Table 1: **Hyperparameters of the on- vs. off-policy distillation experiments.** Values not motivated in the text follow Zhao et al. (2026a), with three teachers rather than their two and one seed rather than their three.

Setup	
Trainer	generalized distillation trainer of TRL 1.9.2 (von Werra et al., 2020)
Reference implementation	Zhao et al. (2026a)
Generation engine	vLLM (Kwon et al., 2023), colocated
Dataset	DeepScaleR (Luo et al., 2025)
Student model	Qwen3-0.6B-Base (Yang et al., 2025)
Teacher models	Qwen3-1.7B, Qwen3-4B, Qwen3-8B, thinking disabled
Prompt frame	bare problem text
Completions generated	40000 per teacher
Usable targets	25758 to 27463
Teacher corpus	
Sampling temperature	1.0
Top-p, top-k	none
Generation budget	4096
Truncation rule	first balanced boxed answer
Completions without answer	dropped from training
Filtering on correctness	none
Objective	
Divergence	forward and reverse KL
Prefix source	cached teacher corpus, or on-policy student rollouts
Token restriction	none, full vocabulary
Maximum prompt length	1024
Maximum completion length	4096
Optimization	
Optimizer	AdamW, weight decay 0, clipping 1.0
Learning rate	$5 \cdot 10^{-7}$
Schedule	constant, 45 warmup steps
Optimizer steps	800
Questions per optimizer step	32
Seeds	1
Precision	bfloat16
Hardware	1 H100
In-training evaluation	
Entropy measured on	frozen teacher completions, and student rollouts
Evaluation sets	held-out DeepScaleR, MATH500 (Lightman et al., 2024)
Evaluation sampling	temperature 1.0, top-p 1.0, pinned seed
Evaluation budget	2048 tokens
Endpoint evaluation	
Sampling	temperature 0.6, top-p 0.95, 8192 tokens
Benchmarks	MATH500, GSM8K (3 samples), AMC23 (5 samples)
Verifier	symbolic equivalence, with a numeric fallback

is no turn to end. We therefore prompt both models with the bare problem text, cut each teacher completion right after its first boxed answer, and append the end-of-text token of the student. This avoids a format warmup stage, which would amount to off-policy forward KL distillation, that is, to one of the four objectives being compared.

The teacher corpus and the training length. Teacher completions are sampled at temperature 1 without top-p or top-k truncation, unlike in Zhao et al. (2026a), because the objective is an expectation under the distribution of the teacher. Completions without a boxed answer within 4096 tokens are dropped, which keeps about two thirds of them and biases the corpus towards problems that the teacher solves concisely. Training runs for 800 steps of 32 prompts, just below one pass over the smallest corpus, so that no teacher benefits from more repetitions than another. We add Qwen3-1.7B to the two teachers of Zhao et al. (2026a) because Qwen3-4B and Qwen3-8B are statistically indistinguishable on AMC23 (66.2% against 66.6% accuracy).

Evaluation. We measure the entropy of the student on its own completions, whose contexts drift during training, and on a fixed set of cached teacher completions, on which only the weights of the student can change. Evaluation completions are generated with a fixed seed, so that neighboring evaluation points share contexts. We report the entropy pooled across tokens, as Zhao et al. (2026a) do. As it weights each completion by its length, on the student's own completions it also moves when which completions are long changes, and the per-sequence average, which we also log, can disagree with it in sign. The frozen MATH500 completions keep all 200 problems, whereas the held-out DeepScaleR problems go through the same filtering as the training corpus and differ from one teacher to the other, so we compare teachers on MATH500 only. Endpoint accuracy is measured at temperature 0.6 and top-p 0.95 with a budget of 8192 tokens, averaged over three samples per problem.

Pure bfloat16 training freezes the normalization gains. Following Zhao et al. (2026a), we train in bfloat16 without float32 master weights. All normalization gains then stay bit-identical throughout training, as the bfloat16 spacing at their typical magnitude is four orders of magnitude above the optimizer step, while the weight matrices do move. As the final normalization gain controls the scale of the logits, these runs cannot test any statement about that scale, and we make none.

C.4 Accuracy vs. entropy for self-distillation on the Chemistry dataset

Figure 6 only reported the entropy of the student over time on the Chemistry dataset. We report the corresponding accuracy in Figure 11 for the sake of completeness.

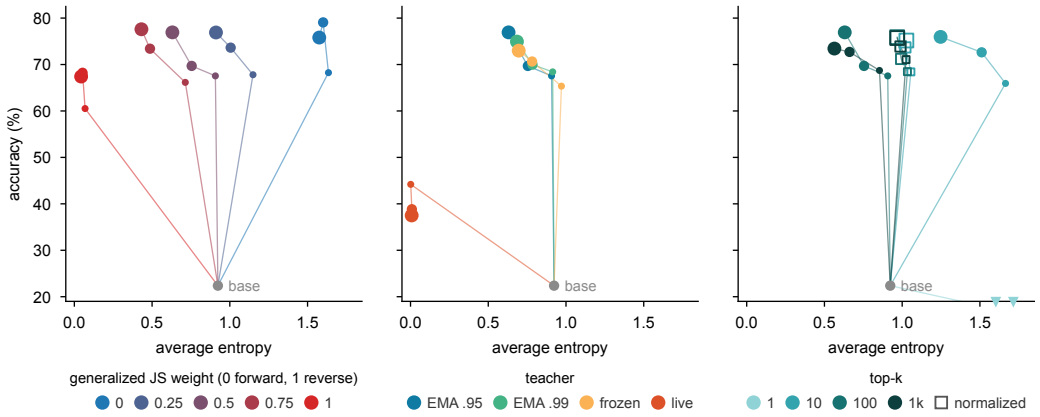

Figure 11: **Accuracy of the student model as a function of its entropy, in the self-distillation experiment of Figure 6**. Each dot is a different checkpoint, the larger the dot, the later in training it is. JS stands for Jensen–Shannon and EMA for exponential moving average.

C.5 Additional self-distillation experiments on GSM8K

The main text follows the experimental setup of Hübotter et al. (2026) on the Chemistry dataset. We repeat it here on GSM8K (Cobbe et al., 2021) with Qwen3-0.6B (Yang et al., 2025), a testbed cheap enough to sweep the learning rate and to train ten times longer; Table 2 lists the hyperparameters of both setups. Figure 12 is the counterpart of Figure 6, at a learning rate of 10^{-5}, and shows the same

qualitative behavior: entropy decreases monotonically with the generalized Jensen–Shannon weight α, a teacher that follows the student too closely collapses, and restricting the divergence to fewer tokens raises entropy.

The learning rate sweep of Figure 13 shows that the more a configuration deflates entropy, the smaller the learning rate it tolerates. Reverse KL already loses to the base model at 10^{-5}, whereas the arms closer to forward KL keep improving up to $3 \cdot 10^{-5}$, and a slowly moving or frozen teacher tolerates rates at which a live one has already collapsed.

We finally use the same testbed to train ten times longer, and ask whether the sharp dependence of entropy on α that the toy model predicts at convergence (Figure 2d) appears once the runs are long enough. Figure 14 shows that entropy does settle within 500 steps. The trajectories separate the two behaviors of Section 3 cleanly: the forward KL arm has entropy much larger than its initial value, while every arm with $\alpha > 0$ settles below it. The ordering in α holds at every point of the trajectory, and it leaves the gap between $\alpha = 0$ and $\alpha = 0.25$ as by far the largest of the sweep, as a phase transition inside that interval would predict. Five weights spaced 0.25 apart are too coarse to confirm it. However, note that each run is evaluated on on-policy data, that is run-specific data, whereas the data was exactly the same in the toy setup.

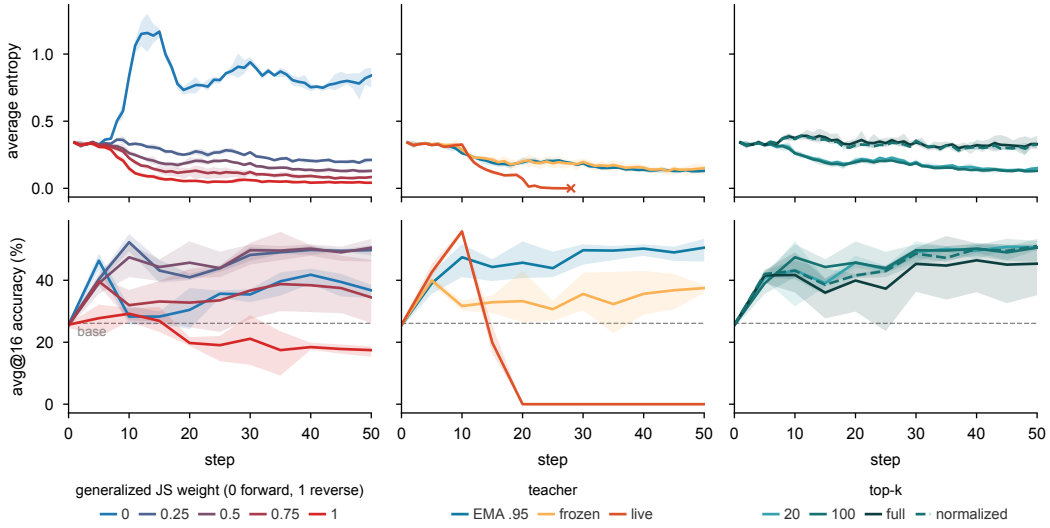

Figure 12: **Self-distillation on GSM8K behaves as on the Chemistry dataset.** Entropy of the student (top) and accuracy, averaged over 16 samples per question (bottom), against training step, varying the weight of the generalized Jensen–Shannon (JS) divergence (left), the teacher (center), and how many tokens are taken into account and whether a tail bucket is used for the remaining tokens or not (right); defaults are $\alpha = 0.5$, an exponential moving average (EMA) teacher with decay 0.95, and top-100 with tail bucket. Lines are means over three seeds and bands their range, and the dashed line is the accuracy of the base model. The cross marks when the student starts behaving randomly.

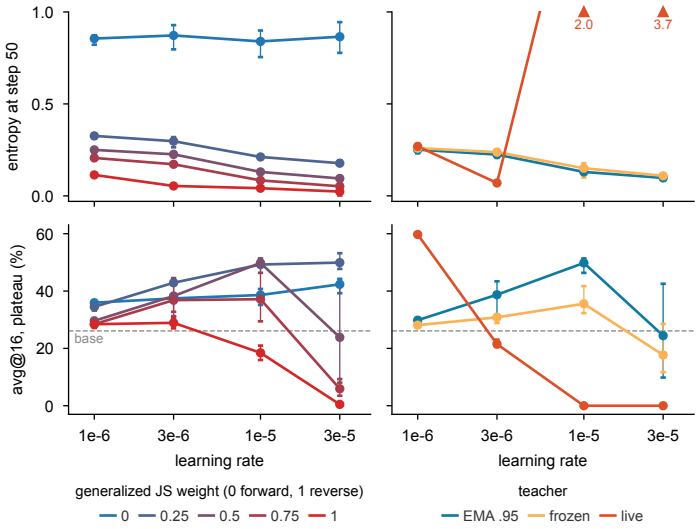

Figure 13: **The more a configuration deflates entropy, the smaller the learning rate it tolerates.** Entropy at step 50 (top) and accuracy over the last five evaluations (bottom) against learning rate, varying the divergence weight (left) and the teacher (right), on GSM8K. Markers are means over three seeds and bars their range; the carets report the two arms whose entropy leaves the axis, with their value in nats. JS stands for Jensen–Shannon and EMA for exponential moving average.

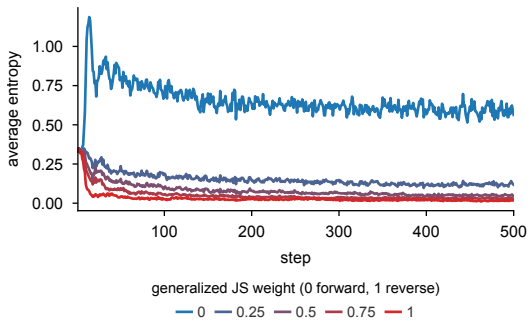

Figure 14: **Entropy settles within 500 steps, and $\alpha = 0$ is the only weight that ends up inflating it.** Entropy of the student over 500 GSM8K steps, one seed per generalized Jensen–Shannon (JS) divergence weight. Forward KL converges at a high entropy level, every other α deflates below the initial entropy, and all configurations have flattened by the end of training, in the ordering the shorter runs already show. The gap between $\alpha = 0$ and $\alpha = 0.25$ is the largest and is consistent with a sharp phase transition in that interval, but the five weights are too coarsely spaced to confirm it.

Table 2: **Default hyperparameters of the self-distillation experiments**, on the Chemistry dataset (Figure 6) and on GSM8K (Figures 12 to 14). A value spanning both columns is shared by the two setups. Defaults follow the released implementation of Hübotter et al. (2026), and each ablation moves a single field away from them. The learning rate and its warmup are the exception: Hübotter et al. (2026) report 10^{-6} without warmup for the setup we replicate and 10^{-5} with warmup for their other ones, and we use the latter, which reproduces the accuracy they report.

	Chemistry	GSM8K
Setup		
Trainer	self-distillation trainer of TRL 1.9.2 (von Werra et al., 2020)	
Reference implementation	Hübotter et al. (2026)	
Sharding, generation engine	DeepSpeed ZeRO-3, vLLM (Kwon et al., 2023)	
Dataset	SciKnowEval chemistry	GSM8K
Student model	Olmo-3-7B-Instruct	Qwen3-0.6B
Training prompts	2100	4000
Thinking mode	disabled	
Rollouts		
Questions per optimizer step	32	
Rollouts per question	8	
Sampling temperature, top-p	1.0, 1.0	
Maximum prompt length	2048	512
Maximum completion length	2048	640
Objective		
Divergence	generalized Jensen–Shannon, $\alpha = 0.5$	
Token restriction	top-100 with tail bucket	
Importance ratio clip	2.0, per token	
Teacher	moving average of the student, decay 0.95	
Privileged information	successful rollout from the same group	
Reference KL, reward scaling	none	
Optimization		
Optimizer	AdamW, weight decay 0.01, clipping 1.0	
Learning rate	10^{-5}	10^{-5}
Schedule	constant after 10 warmup steps	
Optimizer steps	150	50 (500 in Figure 14)
Seeds	1	3 (1 for the 500-step runs)
Precision	bfloat16	
Hardware	8 H100	1 H100
Evaluation		
Evaluation sampling	temperature 0.6, top-p 0.95	
Evaluation questions	210	200 held out
Evaluated every	50 steps	5 steps

C.6 Hyperparameters of the self-distillation experiments