TWO MOMENTA: A TRAINING-DYNAMICS VIEW OF THE EMA TEACHER

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-distillation trains a student against a teacher that is an exponential moving average (EMA) of its own weights. Weight averaging is commonly interpreted as a filter: slow changes pass to the teacher largely intact, whereas fast changes are attenuated by an amount determined by the EMA momentum. In self-distillation, however, the filtered trajectory feeds back into the optimization process that generates it. We therefore analyze the EMA teacher explicitly as a filter and ask what it attenuates at each timescale. We answer that question on behavior rather than weights, from frequency-domain measurements of what both networks compute on a fixed set of images at every step. The student trajectory contains a dominant narrow oscillation, one to two orders of magnitude faster than anything the averaging can follow, so the teacher accumulates the student rather than averaging recent copies of it. That oscillation is a resonance of the student's *other* momentum, the optimizer's: a curvature fitted at one momentum predicts how its period shifts at three others, and the two momenta are coupled through the attenuation of this mode. Removing optimizer momentum at a matched effective step eliminates the identifiable resonance and sharply reduces the teacher's benefit. On CIFAR-100, removing the teacher costs 11.5 linear-probe points at the standard momentum and 0.7 without optimizer momentum. This reduction persists across three additional datasets, including ImageNet-1k, and at twice the training budget; removing the teacher also halves the model's forward FLOPs per training step.

1 INTRODUCTION

Many self-supervised methods, from data2vec (Baevski et al., 2022) to DINOv2 (Oquab et al., 2024) and I-JEPA (Assran et al., 2023), train a student network against a teacher whose weights are an exponential moving average (EMA) of those of the student (Tarvainen & Valpola, 2017; He et al., 2020; Grill et al., 2020). Only the student is trained where an optimizer such as AdamW updates its weights from the gradients of a loss that compares student and teacher outputs on different views of the same input, produced by augmentation or masking. The teacher receives no gradients and changes only through the average. Because it averages recent student iterates, the teacher thus resembles Polyak–Ruppert averaging (Polyak, 1990; Ruppert, 1988), which suppresses step-to-step optimization noise through iterate averaging. This training strategy has been very effective: DINO reports that the teacher outperforms the student throughout training (Caron et al., 2021).

Nonetheless, this training strategy depends too much on the teacher. Removing it breaks many of the methods, and the teacher momentum measurably affects the downstream accuracy of the learned representations (Grill et al., 2020; He et al., 2020; Busbridge et al., 2023). The preferred momentum also varies across settings, making generalization difficult. Complicating matters more, other studies show that an EMA teacher is not always necessary. SimSiam avoids collapse with a stop-gradient alone (Chen & He, 2021), while Tian et al. (2021) obtain teacher-free training by modifying the predictor.

This raises a sharper question: when is the EMA teacher useful and when can it be removed? To address it, we need a deeper understanding of what an EMA teacher actually does. The teacher basically follows the student and is a smoothed, delayed copy of the student's trajectory. Whatever the teacher contributes must therefore come from the trajectory itself. Is there any intrinsic structure in the student's trajectory that determines when the teacher helps learning?

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.

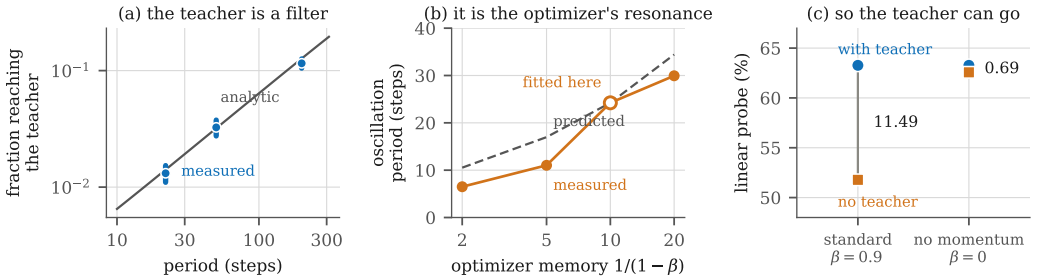


Figure 1: When is a teacher needed? (BYOL, ResNet-18, CIFAR-100.) (a) The teacher behaves as a filter of the student on the networks' outputs. (b) The student's oscillation tracks the *optimizer*'s momentum: a curvature fitted at one momentum predicts how the period shifts at three others. (c) Removing optimizer momentum at a matched step size reduces the linear probe accuracy cost from 11.5 points to 0.7. (Note: β denotes the optimizer momentum.)

We know that from signal processing perspective, the EMA update can be considered as a first-order linear low-pass filter applied to the student's trajectory, indexed by optimizer steps. The EMA teacher passes the low-frequency components of the trajectory and attenuates the high-frequency ones. The teacher's effect therefore depends on the frequency content of the student's trajectory, which is shaped in part by the optimizer. This suggests that the teacher momentum and optimizer momentum *jointly* determine how strongly the trajectory's fluctuations are suppressed, and that whether a method needs a teacher becomes a question about how its student is optimized.

To test this hypothesis, we treat training as a dynamical system. The student, its optimizer, and the teacher form a coupled system whose state evolves step by step, and the interaction between teacher momentum and optimizer momentum is highly nonlinear. We adopt Koopman operator theory here that represents a nonlinear system through a linear operator acting on observables of its state, whose eigenvalues and modes give the system's characteristic frequencies and decay rates (Mezić, 2021). Dynamic mode decomposition (DMD) estimates these modes directly from time series of observables (Schmid, 2010; Tu et al., 2014).

We take the networks' outputs on a fixed probe set as our observables. We then evaluate student and teacher on the probe set at every training step, estimate the transfer function from student to teacher outputs in the frequency domain, and use DMD to identify their shared dynamical modes. Earlier Koopman analyses of training operate in weight space (Dogra & Redman, 2020; Redman et al., 2022; Wang et al., 2026), where neural optimization trajectories have proved difficult to model accurately (Brokman et al., 2024); section A places the present work against that literature.

Figure 1 summarizes our findings for BYOL (Grill et al., 2020) with a ResNet-18 on CIFAR-100; other frameworks and a second architecture are covered in section 5.1. The student's output trajectory is dominated by a single narrow-band oscillation, one to two orders of magnitude above the teacher's cutoff. At such frequencies the EMA behaves as an integrator: it passes an attenuated, phase-lagged copy of the oscillation rather than a local average of recent student iterates (fig. 1a). The oscillation's period shifts with optimizer momentum as the classical heavy-ball model predicts (fig. 1b), and removing optimizer momentum reduces the cost of dropping the teacher from 11.5 to 0.7 linear-probe points (fig. 1c).

The contribution of the paper is three-fold: i) We identify a training-dynamics role of the EMA teacher in BYOL-like self-distillation and characterize its boundary. The student's oscillation is a resonance of its own optimizer and the teacher is a filter acting on it, so teacher momentum and optimizer momentum are linked through the suppression of the dominant oscillatory mode. This mechanism predicts how the oscillation period shifts with optimizer momentum, and that the cost of removing the teacher becomes small once optimizer momentum is removed. This mechanism does not apply to DINO or MoCo v3: in DINO the teacher prevents collapse regardless of optimizer momentum, and in MoCo v3 the teacher is removable even with optimizer momentum. ii) We measure the student-to-teacher transfer directly in the frequency domain and show that the teacher follows the student as a first-order filter that closely matches the analytic EMA response. iii) Guided

by this analysis, we obtain a teacher-free BYOL recipe: removing optimizer momentum reduces the cost of dropping the teacher from 11.49 to 0.69 linear-probe points on CIFAR-100, with similar reductions on two further datasets, and cuts it from 15.95 to 2.37 on ImageNet-1k.

2 BACKGROUND

Let $\theta_{\mathrm{S}}(t)$ be the student's weights after step t. The teacher receives no gradient and follows

$$\theta_{\mathrm{T}}(t) = \lambda\,\theta_{\mathrm{T}}(t-1) + (1-\lambda)\,\theta_{\mathrm{S}}(t), \tag{1}$$

while the student is trained to match the teacher's output on a differently augmented view of the same image (Grill et al., 2020; He et al., 2020; Caron et al., 2021). We distinguish two momentum parameters: λ for the teacher momentum and β for the optimizer momentum, with $\tau = 1/(1-\beta)$ the number of recent gradients a step averages.

For fixed λ, eq. (1) is a linear time-invariant filter, so how much of the student reaches the teacher depends on the frequency content of the student's trajectory. Decomposing the student's trajectory into sinusoidal components, a component of period $T = 2\pi/\omega$ steps passes to the teacher at the same frequency, scaled and delayed by the frequency response of a one-pole low-pass filter (Oppenheim & Schafer, 1999; Yau et al., 2026),

$$H(\omega) = \frac{1-\lambda}{1-\lambda e^{-i\omega}}, \tag{2}$$

where $H(\omega)$ is a complex number whose magnitude gives the gain or attenuation of that component, and phase gives the corresponding lag. A single parameter λ sets both the gain and the lab. The closer λ is to one, the more the teacher attenuates fast components, but the further back in the student's history it reaches, and the more it lags. Let $\varepsilon = 1-\lambda$. For components much faster than the teacher can follow, with periods well below $2\pi/\varepsilon$ steps but still longer than a few steps, the response simplifies to

$$|H(\omega)| \approx \frac{\varepsilon}{\omega} = \frac{\varepsilon\,T}{2\pi}, \qquad \arg H(\omega) \to -\tfrac{\pi}{2}, \tag{3}$$

The teacher then passes such a component with an amplitude proportional to its period, so a component twice as fast comes through at half the amplitude, and delays it by a quarter period. In this regime the EMA behaves as a scaled integrator: it sums the student's fast motion rather than averaging it out.

The student's updates have their own dynamics: the optimizer momentum can make them oscillate. For a locally quadratic loss with heavy-ball momentum β and learning rate η, a Hessian eigendirection with curvature h is underdamped whenever $(1+\beta-\eta h)^2 < 4\beta$, and then oscillates with period

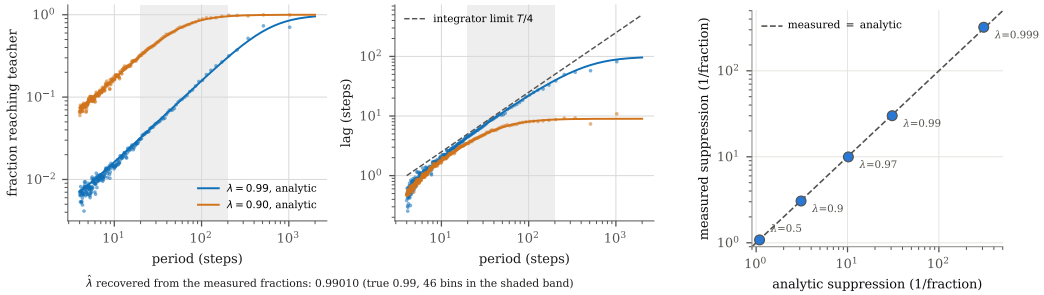
$$T = \frac{2\pi}{\arccos\!\big((1+\beta-\eta h)/(2\sqrt{\beta})\big)}, \tag{4}$$

a classical result from Polyak (1964); Qian (1999); Goh (2017). Such oscillations have been observed in supervised training (Kunin et al., 2024). Our recipe optimizes with LARS, whose layer-wise rescaling leaves this form approximately intact and enters through the product ηh (section C). For β close to one and $(1-\beta)^2 \ll \eta h \ll 1$, the oscillation frequency is $\omega^\star \approx \sqrt{\eta h - (1-\beta)^2/4}$, and the teacher passes this component with gain $|H(\omega^\star)|$, which depends on $\omega^\star$ only through the ratio $(1-\lambda)/\omega^\star$. Holding that gain fixed therefore requires $1-\lambda \propto \omega^\star(\eta, h, \beta)$, coupling the teacher momentum to the optimizer's dynamics, including its momentum.

3 THE TEACHER AS A FILTER

In section 1, we hypothesized that the EMA teacher acts as a low-pass filter on the student's trajectory. In this section, we verify it by first choosing observables of the networks' outputs and checking that they are reliable, then using them to estimate the student-to-teacher transfer and recovering the teacher momentum from the output alone.

Observing the networks during training. At every training step we pass a fixed probe set of images through both networks and summarize each network's outputs by a few scalar time series, the



(a) (b)

Figure 2: Measured against analytic, with no fitting and no free parameter. BYOL, ResNet-18, CIFAR-100. (a) At each timescale, against $|H(\omega)|$ (Oppenheim & Schafer, 1999) for the λ each run used; shading marks the oscillation band the claim covers, 20–200 steps. (b) Suppression of that band against the same expression over a sweep of λ; the dashed line is equality, not a fit.

observables of our Koopman analysis. We consider two kinds. The first describes the geometry of each network's representation: the logarithms of the leading eigenvalues of the feature covariance, *log eig1*, measures how spread out the representations are along their dominant direction. Summaries of this kind are standard for monitoring self-supervised representations (Garrido et al., 2023). The second is the *agreement* between the two networks: the mean cosine similarity between the student predictor's and the teacher projector's outputs, which equals 1 when the two behave identically. Section C details how each series is recorded and estimated.

An observable is useful only if its fluctuations reflect the network rather than the particular images in the probe set. An estimate from a finite probe set can fluctuate for either reason, and its spectrum looks similar in both cases. We therefore split the probe set into two disjoint halves, compute each observable separately on each half, and measure how well the two estimates agree at each frequency using the magnitude-squared coherence (Bendat & Piersol, 2010). High coherence means the fluctuation is reproducible across probe subsets and so reflects the network. Only *log eig1* and the agreement pass this test; the next four eigenvalues do not, and we discard them (table 7). The same concern for reliable readouts guides how we detect training failure. We diagnose collapse by representation rank rather than by linear-probe accuracy: three repeats of one collapsed configuration all have rank 1.00, while their probe accuracies differ by eight points.

Recovering the Teacher Momentum from Outputs. The EMA is defined on weights. If the student's weights change little over the teacher's averaging window of about $1/(1-\lambda)$ steps, a first-order expansion makes the teacher's outputs approximately the same EMA of the student's outputs, so we expect the filter of eq. (2) to hold approximately on outputs as well.

We test this frequency by frequency. From the student's and teacher's observable series we estimate the gain and phase of the student-to-teacher transfer by cross-spectral analysis over a 5000-step window, in two runs that differ only in λ. Both follow eq. (2), evaluated at each run's λ with no fitted gain or phase, over periods from 20 to about 5000 steps, more than two orders of magnitude, and match it to within 1% in gain from 20 to 200 steps (fig. 2a); section G gives per-period values and explains why the range stops at 20 steps. This band lies above the teacher's cutoff frequency, where eq. (3) gives $|H(\omega)| \propto 1/\omega$: the teacher is $(1-\lambda)$ times a running integral of the student, not an average of recent student iterates. The agreement persists across a sweep of λ from 0.5 to 0.999, which changes the attenuation of the oscillation band by more than two orders of magnitude; the measured attenuation follows the analytic value throughout (fig. 2b).

Because the measured gain matches eq. (2) this closely, it can be inverted to recover λ from the outputs alone, to four decimal places. The inversion requires the measured band to lie above the cutoff: below it, $|H| \approx 1$ regardless of λ, so the gain carries no information about λ. At $\lambda = 0.5$ the cutoff lies above the band, and the recovery fails, as expected (table 9). Controls for the inversion are in section C.

4 WHAT THE TEACHER FILTERS

The teacher can matter only at frequencies where the student's trajectory carries variation that the EMA substantially attenuates. Drift lies in the pass band and is transmitted with little attenuation, so drift alone would leave λ inconsequential. We therefore look for faster, structured components of the student's dynamics.

4.1 WHERE THE OSCILLATION COMES FROM

The reliable observables carry one dominant narrow, damped oscillation rather than a smooth drift, shared across the encoder and the projector. In its band the encoder and projector series move together at coherence 0.44–0.64, against 0.04–0.11 between independent runs, at essentially zero phase. The step size and the curvature the student occupies set its period, and it drifts from roughly 14 steps early in training to about 40 late, passing through about 22 in mid-training of this recipe. Throughout training that period stays one to two orders of magnitude shorter than the filter's corner period, placing the oscillation in the teacher's attenuated regime.

To test whether the oscillation arises from the optimizer's dynamics, we compare its period with the heavy-ball prediction of eq. (4). We fit the single curvature parameter h on the $\beta = 0.9$ run *alone* and evaluate eq. (4) at three held-out momenta, over a range in which the period moves by a factor of 4.6, with nothing fitted to those three. A constant period with the same one free parameter fits them far worse (fig. 1). Along the learning rate the period drifts as the same formula requires, though more shallowly than a fixed curvature implies, and the three held-out periods fall below the prediction by 13 to 38%, so eq. (4) captures the scale and ordering of the shift rather than its exact value. This is consistent with the effective curvature changing across runs as the learning rate changes with β. The one-step map among the observables puts 2.2% of its energy off the diagonal, indicating that the reliable observables are dominated by a single shared band. That single-band structure is what lets the two momenta of section 2 be linked through a single $\omega^\star$: had the student's motion been spread over several bands, there would be no single suppression to hold fixed. A per-mode teacher, giving every mode its own decay rate, brings no benefit in this regime (section D).

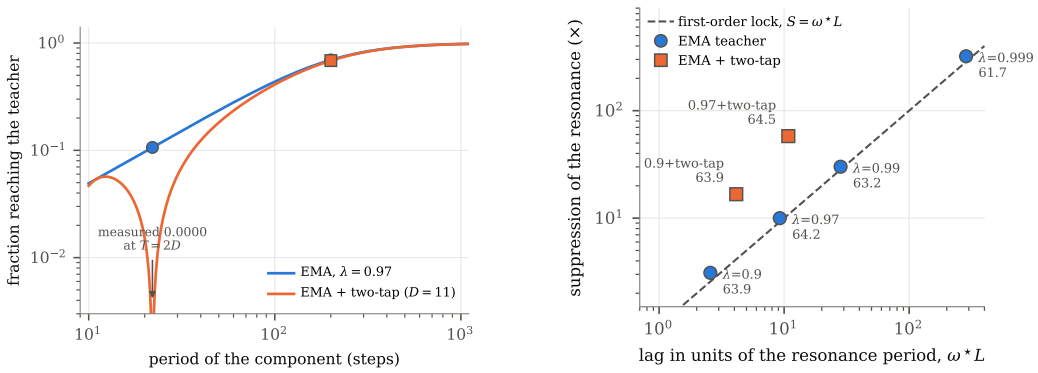
4.2 SUPPRESSION, NOT MEMORY

To separate suppression from long-term memory, we construct a teacher that strongly suppresses the observed oscillation while retaining only a short history: averaging the student with its own state eleven steps earlier, $\theta_{\mathrm{T}}(t) = \frac{1}{2}[\theta_{\mathrm{S}}(t) + \theta_{\mathrm{S}}(t-11)]$, gives a *two-tap* teacher. It cancels a 22-step oscillation exactly and keeps nothing older than eleven steps, against 250 for the EMA it replaces, and its measured transfer matches the analytic response to four decimal places, the predicted zero at the 22-step period included (fig. 3a). It recovers 89 to 96% of what removing the teacher costs, where an EMA of comparable lag recovers about 60% (table 1). Long-horizon averaging is therefore not necessary to recover most of the teacher's benefit in this setting. Because the resonance period drifts during training, the exact cancellation holds only where that period passes through 22 steps, but the experiment still bounds the memory a teacher needs.

Table 1: Standard recipe, BYOL / ResNet-18 / CIFAR-100, at fixed λ; accuracies are means over n seeds. R is the fraction of the teacher's benefit recovered, defined so that the $\lambda = 0.9$ teacher is 1 and no teacher is 0, and given as the range over seeds. Remaining teachers in table 12.

teacher	n	linear	kNN	R linear	R kNN
$\lambda = 0.9$ (EMA reference)	1	63.92	59.21	—	—
two-tap only ($\lambda = 0$)	2	62.66	58.01	0.89–0.93	0.94–0.96
$\lambda = 0.5$ (comparable lag)	2	58.51	49.92	0.55–0.68	0.57–0.69
no teacher ($\lambda = 0$)	2	49.73	34.05	—	—

A teacher's lag is the age of the student it reflects, $\lambda/(1-\lambda)$ steps on average, and it is not interchangeable with suppression: both extremes are worse than the middle. With no lag at all the target is the student itself and training degrades to the teacher-free level (table 1). At very large lag the target reflects an outdated student state and accuracy falls. Which of the two carries that cost can



(a) (b)

Figure 3: BYOL, ResNet-18, CIFAR-100. (a) Analytic response of an EMA at $\lambda = 0.97$ and of the same EMA followed by a two-tap average, $|H(\omega)|\,|\cos(\pi D/T)|$, with points measured in frequency domain. (b) Suppression against lag, the lag in units of the resonance period so that the first-order lock $S = \omega^\star L$ is the diagonal; annotations give each teacher's accuracy. The resonance is about 22 steps in mid-training of this recipe.

be separated, because the two-tap teacher reaches the suppression of a large λ at a fraction of its lag and loses nothing by it (fig. 3b): within the range we tested, more suppression does not reduce accuracy while more lag does, and the teachers we ran put the harmless range at one to two orders of magnitude of lag. That is the lock a first-order teacher works under: suppression and lag cannot be set separately, since raising one raises the other along eq. (3), which is why λ has an interior optimum at all. The coupling can be avoided either by a filter that is not first-order, such as the two-tap teacher, or by removing the oscillation at its source.

Two practices already in the literature follow from the lock. BYOL's reference implementation raises λ as the training budget grows while lowering the learning rate (DeepMind, 2020), moving λ the way holding the suppression fixed requires: a smaller η moves $\omega^\star$ down, and $1 - \lambda$ with it. And the optimum in λ is broad (Grill et al., 2020; He et al., 2020), as a harmless range of lag gives without a tuned value. Section E adds a third.

5 Removing the resonance

We next test whether the teacher remains useful once the identifiable resonance is gone. At $\beta = 0$ the heavy-ball iteration has no complex mode, and the resonance disappears from the measured band. In the spectra the $\beta = 0.9$ peak sits at 24 steps wherever the search window is put, while at $\beta = 0$ the apparent peak follows the window's lower edge, from 7 steps to 12, so no stable period is selected. The band the teacher works in holds 0.8% of the student's motion at $\beta = 0$ against 50.9% at $\beta = 0.9$ (table 8). We test the consequence with a 2×2 crossing the optimizer's momentum with the presence of a teacher, at a matched effective step.

In the quartet, dropping the teacher with momentum costs 11.49 linear-probe points and substantially degrades the representation; without it, the same change costs 0.69 (table 2), inside the spread of repeating a configuration. Teacher-trained accuracy is unchanged across the two momentum settings, so the difference is confined to the teacher-free arm. A small gap is informative only if both arms remain healthy, since two degraded runs can agree as easily as two good ones (section H), so we check each side differently: the run with a teacher against the recipe's reference accuracy, the one without for collapse by representation rank. Training loss cannot serve as that check, being the effect under measurement: without a teacher it falls as the student's momentum grows while accuracy degrades (section H). At $\beta = 0$ the teacher-free runs vary by 0.10 points over four seeds against 0.23 with a teacher, so dropping the teacher does not cost repeatability.

Matching the effective step makes the learning rate ten times larger at $\beta = 0$, so the two arms differ in gradient noise as well as in resonance, and the teacher's value could lie in averaging that

Table 2: Linear probe for both teacher settings and their gap, all BYOL with a ResNet-18 on CIFAR-100, with n the number of seeds and *rank* the representation rank of the teacher-free run: 1.00 is a constant output. Rows vary $\tau = 1/(1-\beta)$ at fixed $\eta/(1-\beta)$; the last two repeat the ends of that range at twice the budget. Spreads over repeats and the kNN column are in table 13.

β	τ	n	teacher	no teacher	Δ	rank
0.9 (standard)	10	2	63.27	51.78	11.49	1.50
0.5	2	2	63.28	62.60	0.68	1.53
0.8	5	1	64.18	60.14	4.04	1.58
0.95	20	1	63.85	44.01	19.84	1.54
0	1	4	63.27	62.57	0.69	1.51
0.9, 400 ep	10	1	67.68	59.81	7.87	1.66
0, 400 ep	1	1	67.40	66.64	0.76	2.08

noise away, the Polyak–Ruppert reading of section 1, rather than in suppressing a resonance. That matching does more than it appears to: writing the gradient as a mean plus noise, the mean step and the diffusion over N steps are both fixed by $\eta/(1-\beta)$ and so are equal on the two arms; what differs is the per-step noise, larger at $\beta = 0$ than at $\beta = 0.9$ by $\sqrt{(1+\beta)/(1-\beta)} = 4.4$ against 4.5 measured (section C). Momentum low-passes the injected noise with a one-pole filter of the same form as the teacher's, so the extra motion at $\beta = 0$ lies at periods below the band the teacher works in. A denoising account therefore predicts a teacher worth *more* at $\beta = 0$, where there is more noise to remove. It is worth 0.69 there, against 11.49.

What the optimizer's memory changes. We next separate two axes: the optimizer's memory τ, which controls the oscillatory dynamics, and the effective step $\eta/(1-\beta)$, which controls how far each update moves. Sweeping the first while holding the second turns the comparison above into a curve (fig. 4a). We set the learning rate to $10(1-\beta)$ at every point, keeping the effective step identically 10 along the sweep. Removal costs the same at one and two steps of memory, 0.69 and 0.68, then grows past the standard $\beta = 0.9$, in proportion to the memory from $\tau = 5$ upward. A kNN readout traces the same curve at a larger amplitude, so the dependence is not specific to the fitted linear probe. The teacher-trained runs span 0.44 points across the sweep, less than the recipe's run-to-run spread, so the response is concentrated in the teacher-free arm. We also hold the optimizer's momentum and vary how far a step moves (table 14). Removal costs most at the largest step, while the two below it are indistinguishable at the run-to-run variability of these cells.

A longer budget, and two more datasets. To test whether the effect changes with training duration, we ran the same quartet at 400 epochs (table 2). The ratio is 9.7% against the 30% fixed in advance, with no collapsed configuration, and doubling the budget moved the cost at high momentum while leaving the one at $\beta = 0$ where it was. The result is also insensitive to holding the teacher's momentum at its short-budget value (section E). The pattern also holds on Tiny-ImageNet at 64×64 with 200 classes, on CIFAR-10, and on ImageNet-1k (table 3a). The linear probe bears the prediction out on each; the kNN margin is narrowest on Tiny-ImageNet, 28.8% against the 30% fixed in advance, and still inside it. The ImageNet-1k quartet uses a ResNet-50, one seed at 40 epochs, so its absolute level is below published ones and we read it as a qualitative generalization test rather than a benchmark result: the teacher arms differ by less than half a point while the teacher-free arms differ by about 14. CIFAR-10 gives a second test in which teacher-trained accuracy is insensitive to β. We report ratios between gaps only after averaging over its two seeds, since the per-seed denominator is small enough that the per-seed ratios differ in sign.

5.1 The teacher's other role

We repeat the 2×2 in MoCo v3, SimSiam and DINO to see whether the momentum dependence measured in BYOL generalizes across self-distillation frameworks, and organize them by what prevents collapse in each (table 4). BYOL differs from the other three in that its teacher-removal cost depends strongly on the optimizer's momentum, as the account predicts where the teacher is compensating and nothing else is.

Table 3: The 2×2 of table 2 beyond CIFAR-100 and beyond ResNets. (a) BYOL / ResNet-18: what removing the teacher costs at either momentum, and the ratio between them against the 30% fixed in advance; the CIFAR-10 and Tiny-ImageNet rows average two seeds; the ImageNet-1k row is a ResNet-50, one seed at 40 epochs. (b) On a ViT-Tiny the teacher-free run collapses, so the claim rests on the rank of the teacher-free run rather than on its probe: a probe fitted on a rank-1.00 representation can land anywhere, and the two seeds of that cell are 8.2 points apart where every other cell agrees to within half a point.

dataset	$\beta = 0.9$	$\beta = 0$	ratio	ratio kNN
CIFAR-100	11.49	0.69	6.0%	6.4%
Tiny-ImageNet	9.50	0.87	9.1%	28.8%
CIFAR-10	4.81	0.39	8.1%	5.5%
ImageNet-1k	15.95	2.37	14.8%	18.5%

(a)

BYOL / ViT-Tiny	n	teacher	no teacher	rank
$\beta = 0.9$	2	55.25	14.29	1.00
$\beta = 0$	2	53.24	52.28	1.42

(b)

Table 4: Each framework's standard recipe on CIFAR-100, run as a 2×2 at matched effective step; values are linear-probe points. The BYOL row is the multi-seed one of table 2; the other three are single runs of the same protocol.

framework	what prevents collapse	$\beta = 0.9$	$\beta = 0$	removable?
BYOL	predictor and BN	11.49	0.69	once β is gone
MoCo v3	negatives	0.44	0.17	at either momentum
SimSiam	predictor and output BN	1.40	1.26	at either momentum
DINO	the teacher	51.48	45.63	at neither: it collapses

DINO is the only framework in which removing the teacher collapses the representation to rank 1.00 and a constant output at both momentum settings, in both samples we have of each cell; removing β_1 does not prevent it. Because Adam has no exact analogue of the matched effective step used for BYOL, this comparison is qualitative; both teacher-free settings nevertheless collapse to rank 1.00. In MoCo v3, where negatives prevent collapse, both costs fall inside the spread of repeating that cell, two orders of magnitude below BYOL's, leaving no resolvable cost for the momentum to modulate.

SimSiam gives a complementary control, since its standard formulation has no EMA teacher. We added one at $\lambda = 0.996$, holding the effective step at 0.5 on both sides and changing no other field, and ran the same 2×2. The teacher is worth about 1.3 points there, but that gain is essentially unchanged when the optimizer's momentum is removed. The momentum dependence measured in BYOL is therefore not a generic property of EMA teachers. The frameworks differ, then, in what prevents collapse, and the compensation measured here sits on top of a baseline that does not collapse: where the teacher is instead what keeps the representation alive, no change to the optimizer replaces it.

A second architecture. To test whether any of this is specific to ResNets, we repeat the same 2×2 on a ViT-Tiny. With the optimizer's momentum, removing the teacher collapses the representation, to rank exactly 1.00 in every sample we have of that cell (table 3b). At $\beta = 0$ none of the teacher-free runs collapse, and the same removal costs 0.96 points, so on this architecture removing the optimizer's momentum prevents the collapse. We report no ratio for the cell, because the $\beta = 0.9$ teacher-free representation collapses. As in the ResNet experiments, the momentum dependence is concentrated in the teacher-free arm.

DINO gives the complementary result: its quartet matches this one in ViT-Tiny, optimizer, dataset, budget and batch size, the framework being the only difference, and there the representation collapses without a teacher at either momentum. Because the two share the same ViT-Tiny setup but differ in whether $\beta = 0$ prevents teacher-free collapse, this boundary is framework-dependent rather than architecture-dependent. DINO's teacher-trained run itself loses ground between the momenta where BYOL's barely does.

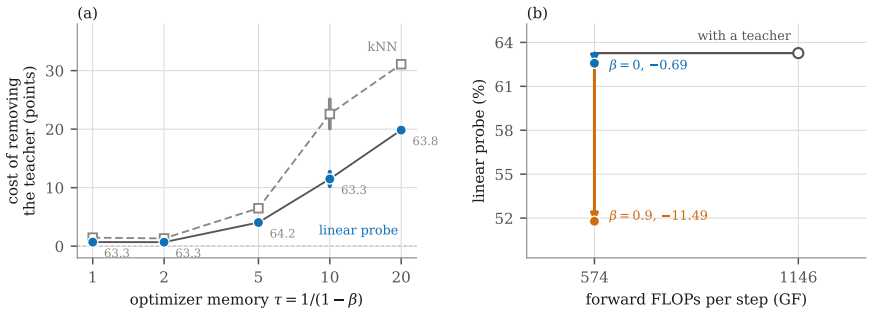


Figure 4: What the optimizer's memory decides. BYOL, ResNet-18, CIFAR-100. (a) The cost of removing the teacher against τ at a fixed effective step, seed-paired and read two ways; grey labels give the accuracy the teacher arm itself reaches. (b) The horizontal move is what removal saves and the vertical drop is what it costs, from the one point both teacher-trained runs share.

5.2 Computational savings

At $\beta = 0$ the teacher can be removed at a small accuracy cost, and its removal reduces training computation and memory. Forward FLOPs fall by half, the parameters carried during training from 30.8M to 16.4M, and peak memory with them (table 15). The removed parameters are the projector and predictor, which evaluation discards but training retains, where they are paid for at every step in the training graph, the optimizer state and the checkpoints. The saving is independent of the optimizer's momentum while the accuracy it costs is not (fig. 4b), so whether it can be taken is decided by the optimizer. The corresponding wall-clock improvement depends on implementation and input-pipeline overhead; section F reports measured step times.

The short-memory teacher addresses a different question: at a 50-epoch budget it is ahead of the standard recipe in epochs rather than in time (fig. 5). We use it as a diagnostic of how much memory a teacher needs rather than as a proposed method.

6 Discussion and limitations

Our results identify the suppression of an optimizer-associated resonance as a major role of the EMA teacher in BYOL-like self-distillation. A two-tap teacher holding only half the oscillation's period, eleven steps in this recipe, recovers most of the standard EMA's benefit, so long-horizon memory is not what that role requires. Removing the optimizer's momentum removes the identifiable resonance and sharply reduces the teacher-removal cost, on four datasets and two backbones. The account does not extend to settings where the teacher is instead what prevents collapse, as in DINO. We sweep the teacher's momentum at one training budget and vary the budget at one teacher setting; because the reference implementation raises λ as the budget grows (DeepMind, 2020), interactions between the two axes remain untested. The $\beta = 0$ comparison matches the effective step $\eta/(1-\beta)$, so its conclusions depend on that being the appropriate quantity to preserve when the optimizer's momentum changes; the one quantity it leaves unmatched, the per-step noise, predicts the opposite of what we measure (section 5). Our frequency-domain measurement covers periods of 20–200 steps, and eq. (4) serves as a within-run approximation rather than an exact cross-run predictor, because the effective curvature changes across training settings.

The filter view identifies when suppressing the student's fast dynamics makes a teacher useful, but it does not settle the route by which that suppression improves the representation. It also leaves the framework-dependent size of the effect unexplained: the same slowdown costs an order of magnitude less in SimSiam than in BYOL (section 5.1), and DINO's teacher carries further mechanisms, centering and sharpening among them, that we did not test.

AI USE STATEMENT

We used generative AI tools to assist with code debugging and organization, literature search, and language editing. The authors developed the research questions, methodology, experimental design, interpretation of results, and scientific conclusions. The authors reviewed and verified all AI-assisted outputs and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix E give the full training and evaluation protocol, including the linear-probe details that change absolute accuracy by up to 10.6 points. The configurations of every run in table 2 are given here in full, and we will release the code upon acceptance.

ETHICS STATEMENT

This work studies training dynamics of self-supervised vision models on public benchmarks (CIFAR-100, CIFAR-10, Tiny-ImageNet, ImageNet-1k) and raises no specific ethical concerns beyond those of the datasets themselves.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning (ICML)*, 2022.

Julius S. Bendat and Allan G. Piersol. *Random Data: Analysis and Measurement Procedures*. Wiley, 4th edition, 2010.

Adam Block and Cyril Zhang. EMA without the lag: Bias-corrected iterate averaging schemes. *arXiv preprint arXiv:2508.00180*, 2025.

Jonathan Brokman, Roy Betser, Rotem Turjeman, Tom Berkov, Ido Cohen, and Guy Gilboa. Enhancing neural training via a correlated dynamics model. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2312.13247.

Dan Busbridge, Jason Ramapuram, Pierre Ablin, Tatiana Likhomanenko, Eeshan Gunesh Dhekane, Xavier Suau, and Russ Webb. How to scale your EMA. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6, 2022.

DeepMind. BYOL reference implementation, `byol/configs/byol.py`. `https://github.com/google-deepmind/deepmind-research`, 2020. EMA and learning-rate presets by training length. accessed 2026-09-23.

Akshunna S. Dogra and William T. Redman. Optimizing neural networks via Koopman operator theory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ye Du, Yujun Shen, Haochen Wang, Jingjing Fei, Wei Li, Liwei Wu, Rui Zhao, Zehua Fu, and Qingjie Liu. Learning from future: A novel self-training framework for semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2209.06993.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, 2023.

Gabriel Goh. Why momentum really works. *Distill*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Daniel Kunin, Javier Sagastuy-Brena, Lauren Gillespie, Eshed Margalit, Hidenori Tanaka, Surya Ganguli, and Daniel L. K. Yamins. The limiting dynamics of SGD: Modified loss, phase space oscillations, and anomalous diffusion. *Neural Computation*, 2024. arXiv:2107.09133.

Igor Mezić. Koopman operator, geometry, and learning of dynamical systems. *Notices of the American Mathematical Society*, 68(7):1087–1105, 2021.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*, 2024.

Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, 1999.

Maxime Oquab et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jonathan Patsenker, Henry Li, and Yuval Kluger. Exponential weight averaging as damped harmonic motion. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. arXiv:2310.13854.

Trung Pham, Chaoning Zhang, Axi Niu, Kang Zhang, and Chang D. Yoo. On the pros and cons of momentum encoder in self-supervised visual representation learning. *arXiv preprint arXiv:2208.05744*, 2022.

Boris Polyak. New stochastic approximation type procedures. In *Autom. i Telemekh.*, 1990.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12 (1):145–151, 1999.

William T. Redman, Maria Fonoberova, Ryan Mohr, Ioannis G. Kevrekidis, and Igor Mezić. An operator theoretic view on pruning deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2110.14856.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *International Conference on Learning Representations (ICLR)*, 2025. Outstanding Paper.

David Ruppert. Efficient estimations from a slowly convergent robbins–monro process. In *Technical report, Cornell University Operations Research and Industrial Engineering*, 1988.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning (ICML)*, 2021.

Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

Fanqi Wang, Weisheng Tang, Landon Harris, Hairong Qi, Dan Wilson, and Igor Mezić. Predictive differential training guided by training dynamics. In *International Conference on Learning Representations (ICLR)*, 2026.

Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

Chung-Yiu Yau, Dawei Li, Athanasios Glentis, Valentyn Boreiko, Hoi-To Wai, and Mingyi Hong. EMA-nesterov: Stabilizing nesterov's lookahead for accelerated deep learning optimization. *arXiv preprint arXiv:2605.25395*, 2026.

A Related work

Why momentum teachers work. BYOL (Grill et al., 2020), MoCo (He et al., 2020; Chen et al., 2021) and DINO (Caron et al., 2021) use a scalar EMA teacher, as did mean teacher (Tarvainen & Valpola, 2017). Tian et al. (2021) analyse non-contrastive learning in a linear model, where the EMA acts as an automatic curriculum; Busbridge et al. (2023) derive how λ must be rescaled with batch size to preserve the dynamics; Pham et al. (2022) find momentum on the projector alone sufficient and attribute it to noisier gradients near the output. These accounts reason from models or from gradient statistics. We measure what the teacher does to the student's behaviour while training runs, which is what turns the question into one about the optimizer.

Momentum, and teachers one can do without. Heavy-ball momentum produces underdamped oscillations in curved directions (Polyak, 1964; Qian, 1999; Goh, 2017), whose signature Kunin et al. (2024) identify in supervised training; we find the same resonance in self-supervised training and connect it to the teacher. That a teacher is dispensable is also known: SimSiam (Chen & He, 2021) trains without a momentum encoder, and Tian et al. (2021) remove it by changing the predictor. What an existence result cannot give is a condition saying when the teacher in a particular method can be dropped, and where that must fail. Ours changes one optimizer hyperparameter inside an existing recipe, leaving architecture and objective alone, and it says the teacher goes when it was compensating the optimizer and stays when it is what prevents collapse. SimSiam reaches teacher-free training by architecture instead, and adding an EMA teacher to it is worth 1.40 points with the optimizer's momentum and 1.26 without, so its teacher's value is real and does not follow the knob (section 5.1).

Averaging, monitoring, and operators. Iterate averaging is classical (Polyak & Juditsky, 1992) and underlies SWA (Izmailov et al., 2018) and the weight EMAs used at evaluation (Morales-Brotons et al., 2024; Block & Zhang, 2025), where reading the average as a low-pass filter is already standard, down to the same transfer function (Yau et al., 2026) and where the averaged model does not feed back into training as it does in self-distillation. That is where the reading has stayed; we take it to a teacher that does feed back, and use it to say what the teacher removes. Patsenker et al. (2023) also read a weight EMA as an oscillating system, through a spring and damper between two particles rather than through frequency, and without the optimizer's momentum as the source of what oscillates. Monitoring through a fixed probe set is likewise established (Garrido et al., 2023); we add per-step, student–teacher-paired measurement with a reliability check on the observables. Koopman operators have been used to forecast weights and skip optimizer steps (Dogra & Redman, 2020; Wang et al., 2026), to explain magnitude pruning (Redman et al., 2022), and are reported a poor model of weight trajectories (Brokman et al., 2024), all on weights; we apply the lens to what the network computes and find the relevant dynamics essentially one-dimensional. Ren & Sutherland (2025) likewise read training as a dynamics to be decomposed, there per example rather than per timescale.

B Reproduction fidelity of the recipes we start from

Every accuracy in this paper is a linear probe, or a kNN classifier, on frozen self-supervised features; none is supervised training, and the two are not on the same scale.

The recipe is solo-learn's `scripts/pretrain/cifar/byol.yaml` with the dataset set to CIFAR-100: LARS with momentum $\beta = 0.9$, teacher momentum rising from 0.99 to 1 on a cosine schedule, asymmetric augmentations, a 4096/256 projector, 200 epochs (39,000 steps). The cells that study the teacher hold λ fixed instead of annealing it. The $\beta = 0$ runs of section 5 make one further change, raising the learning rate tenfold so that $\eta/(1-\beta)$, how far a step actually moves once momentum's amplification is counted, is unchanged; LARS trust clipping does not equalise the two learning rates (section C).

Run at the budget the reference uses, our recipe reproduces it. At 1000 epochs our standard BYOL configuration reaches 69.42 linear-probe accuracy against the 70.46 that solo-learn reports for the same recipe and backbone (da Costa et al., 2022), with the probe's learning rate interior to its grid. The linear probe is the metric the paper's tables use, so this is agreement on the measurement that carries the claims and not on a substitute for it. The reference values throughout this appendix are the Acc@1 entries of the CIFAR-10 and CIFAR-100 results tables in the solo-learn repository, at ResNet-18 and 1000 epochs. Those tables no longer distribute checkpoints and state that re-running the scripts reproduces the numbers, so the reference is supported by a configuration rather than by weights we could have evaluated ourselves. That source does not state the evaluation protocol behind its Acc@1, so the agreement of table 5 is in value at a matched budget and backbone, not in procedure.

Table 5: At the reference budget, the recipe reproduces. BYOL / ResNet-18 / CIFAR-100, 1000 epochs; ours is a linear probe on frozen features, the reference is the reported Acc@1.

1000 epochs	ours	da Costa et al. (2022)	difference
linear probe	69.42	70.46	-1.04

The experiments in the body run for 200 epochs, a fifth of that, which is where the absolute numbers in our tables come from and why they sit five to six points below published ones (table 6). This is the cost of running a design of two-hundred-odd training runs rather than a handful.

Every comparison in this paper is made between runs of one configuration pair carried out together, which is worth one sentence of justification. Running one configuration in two places moved the linear probe by 0.11 and 0.57 points in two settings that are stable, and by 4.12 in one whose teacher-free run is close to collapsing and which varies by 1.77 on repeats of its own. Three observations cannot separate the environment from that arm's own dispersion, and we do not try: they are the reason a gap is only ever read inside a pair, not a measurement of anything.

Table 6: The body's numbers are 200-epoch numbers; references are the 1000-epoch Acc@1 values of da Costa et al. (2022) for the same recipe and backbone, so the comparison is between budgets rather than between implementations.

recipe (CIFAR-100 unless noted)	ours, 200 ep	reference, 1000 ep	difference
BYOL, ResNet-18	63.90	70.46	-6.6
BYOL, ResNet-18, CIFAR-10	87.49	92.58	-5.1
SimSiam, ResNet-18	63.72	66.04	-2.3
MoCo v3, ResNet-18	60.06	68.83	-8.8
DINO, ViT-Tiny	53.67	no matched reference (backbone differs)	

One asymmetry in the Tiny-ImageNet runs is worth recording. In one of its two seeds the $\lambda = 0$ cells were resumed from epoch 85 while those with a teacher ran straight through; the loss change at each restart, 0.0024 and 0.0001, lies inside the three-standard-deviation band of the preceding ten epochs, and the other seed's $\lambda = 0$ cells ran straight through and agree with it, at gaps of 8.84 and 10.16.

Two rows need more than the budget. MoCo v3 is 8.8 points below its reference, further than a fifth of the budget accounts for, so it is the weaker reproduction of the three frameworks; it carries boundary evidence only—its teacher is worth about half a point at either momentum—and nothing in the paper rests on its level. Our DINO runs use a ViT-Tiny where the reference figure is a ResNet-18, so no matched reference exists, and comparing across backbones would not make one; DINO enters the argument through the collapse of its teacher-free runs, which is a difference within our own runs.

C How the series were recorded and estimated

Recording. A fixed set of 512 CIFAR-100 test images passes through student and teacher at every training step, with BatchNorm using the probe batch's own statistics and the running buffers saved and restored around the probe so that training is unaffected. The frequency-domain analysis uses 5000 consecutive steps in mid-training, steps 19,500–24,500 of 39,000, on two seeds; over that window the scheduled teacher momentum runs from 0.99500 to 0.99696, with mean $\bar{\lambda} = 0.99599$, and it is this $\bar{\lambda}$ that the analytic curves use.

Three estimators, on the same series. A forward filter driven by the schedule's actual $\lambda(t)$ maps the student's series to a predicted teacher series, scored by R^2 and by residuals against two controls, a frozen teacher and the unfiltered student. The gain and lag at each timescale come from the cross-spectrum of the two series, by Welch's method (Welch, 1967) with 512-sample Hann windows, 50% overlap and per-segment detrending; A peak detector with a significance test will report peaks where there are none: run on the $\beta = 0$ series it returned $p = 0.000$ at periods of 6.2 to 7.0 steps, but those peaks sat on the lower edge of the search band and followed it down when the edge was moved to 4, 3 and 2.5 steps, while the $\beta = 0.9$ peak stayed at 24.2 under all four windows. Whether a peak moves with the window is the test; a p-value is not. The null itself must come from the dynamics rather than from shuffling, since shuffled increments are white and the estimator divides by a momentum low-pass, which manufactures high-frequency peaks exactly where the momentum is strongest. Readings are kept only where magnitude-squared coherence (Bendat & Piersol, 2010) shows the teacher's variation to be explained by the student's. The student's own dynamics are fitted per coordinate with autoregressive models, whose slowest mode gives a period and a decay time.

What the forward filter is scored against. The predicted teacher series is compared with two controls: a teacher frozen at the start of the window, and the unfiltered student. On two seeds the filter's residuals are 130–150 times smaller than the frozen teacher's and 33–41 times smaller than the unfiltered student's, at $R^2 \geq 0.995$.

Where the band comes from. Coherence between student and teacher falls away below about 10 steps. Two checks separate a numerical floor from measurement noise on the student's side and

point to the latter, which is why the claims are stated for periods of 20–200 steps rather than for the whole spectrum.

Why the momentum is read frequency by frequency. Inverting the measured gain band by band recovers the momentum the run used. Fitting the same recursion in the time domain does not: it is biased by periods outside the analysis band, where most of the forward-residual power sits, and returns a momentum that describes that slow content instead. Anyone repeating the measurement should invert in the frequency domain.

Trust clipping under LARS. The $\beta = 0$ runs raise the learning rate tenfold to hold $\eta/(1-\beta)$ fixed, which would be undone if LARS were clipping most updates. It is not: under 1% of weight tensors are clipped on average after warm-up, and a few percent around the peak learning rate, so the two differ in the step size we intended them to differ in.

We also record the step-to-step change $1-\cos(z(t), z(t-1))$ on the same probes. It carries no claim in the body, but it reproduces across probe halves at 0.95–0.98 in every band, which is expected of a quantity comparing two states on the same probes: the probe-specific part cancels.

Recording does not perturb what it records. One configuration was trained twice from the same seed, differing only in whether the per-step probes were taken. The gap it produced was 0.48 points without recording and 0.39 with, well inside what repeating that cell moves it by.

Reading the optimizer before the momentum field. Our DINO configurations carry `optimizer_momentum` 0.9 in every cell of a momentum sweep, which reads as a sweep that never happened. They use AdamW, whose momentum is `adam_beta1`; the other field belongs to LARS and SGD and is inert there, so the DINO sweep varies β_1.

Where the student's motion sits, at either momentum. The claim of section 5 is scoped to periods of 20–200 steps, so what matters at $\beta = 0$ is not only that no peak survives there but how much motion is left in that band at all. Table 8 measures it. Without a momentum to low-pass it the gradient noise stays white and the increment spectrum piles up at the short periods, which is what the algebra gives, and the step-to-step standard deviations stand in the ratio 4.5 where matching $\eta/(1-\beta)$ predicts 4.4. Section 5 draws the consequence for a denoising account of the teacher.

A peak that moves with the window is not a peak. A peak finder returns a maximum and a significance whether or not the spectrum picks out a period, so the test we use is whether the maximum moves when the search window does. Widening the window's lower edge from 6 to 10 to 12 steps, the $\beta = 0.9$ runs peak at 24.2, 24.2, 24.2 and at 17.2, 17.2, 17.2 steps, while the four $\beta = 0$ runs give 7.0, 10.5, 12.2; 6.4, 11.0, 12.2; 6.5, 11.0, 12.6; and 6.2, 11.8, 12.2, tracking the edge each time.

Table 7: The split-half check of section 3, on the standard BYOL / ResNet-18 / CIFAR-100 recipe. Each candidate is measured on two disjoint halves of the probe set and the two estimates are compared band by band; the range is over frequency bands. Only the first two track the network rather than the probe sample.

candidate observable	split-half coherence	kept
log eig1	0.52–0.86	yes
student–teacher agreement	0.53–0.88	yes
log eig2 to log eig5	0.01–0.22	no

D PER-MODE KOOPMAN MOMENTUM DOES NOT WORK

The design. The natural Koopman-based teacher replaces the scalar in eq. (1) with one rate per mode. Weights are tracked through the LoRA factors (Hu et al., 2022) of the $k = 8$ layers with the largest gradient norm, giving an observable ψ; DMD (Schmid, 2010) on a 200-snapshot window with rank $p \leq 32$ and a Tikhonov-regularized inverse (relative ridge 10^{-2}) gives modes U and

Table 8: Where the student's motion sits, at either momentum, from the increment spectrum on the same probes at $\lambda = 0.99$; BYOL / ResNet-18 / CIFAR-100. The 20–200-step band is the one the claim of section 5 is scoped to.

	seeds	step-to-step sd	power at $T \leq 8$	power in 20–200
$\beta = 0$	4	2.46×10^{-2}	92.8–93.2%	0.7–0.9%
$\beta = 0.9$	2	5.49×10^{-3}	10.6–13.4%	48.4–53.3%

Table 9: The teacher's momentum read back off behaviour on the standard BYOL / ResNet-18 / CIFAR-100 recipe, by inverting $|H|$ at each frequency of the 20–200-step band where the split-half coherence reaches 0.5 and taking the median over bins; the last column is the 10th to 90th percentile across those bins. The recovery holds to four decimal places for every teacher whose corner period lies at or above the measured band, and fails for $\lambda = 0.5$, whose corner sits at 12.6 steps, below it: there the teacher passes almost everything, $|H|$ is close to one, and inverting it is ill conditioned.

run	λ	recovered	error	across frequencies
$\lambda = 0.5$	0.500	0.42654	-0.073	0.359–0.478
$\lambda = 0.9$	0.900	0.89894	-0.0011	0.894–0.902
$\lambda = 0.97$	0.970	0.96993	-0.0001	0.969–0.971
$\lambda = 0.99$	0.990	0.99010	$+0.0001$	0.9898–0.9904
$\lambda = 0.99$, seed 1	0.990	0.99011	$+0.0001$	0.9897–0.9907
$\lambda = 0.999$	0.999	0.99906	-0.0001	0.99902–0.99911
$\lambda = 0.99$, $\beta = 0$	0.990	0.99010	$+0.0001$	0.9896–0.9905

eigenvalues μ_i. With $\delta = \psi_{\mathrm{S}} - \psi_{\mathrm{T}}$,

$$\begin{aligned} \psi_{\mathrm{T}} &\leftarrow \psi_{\mathrm{T}} + U\,\mathrm{diag}(h)\,U^{\top}\delta + (1-\lambda_{\max})(I - UU^{\top})\delta, \\ h_i &= 1 - \left[\lambda_{\min} + (\lambda_{\max} - \lambda_{\min})e^{-\beta \max(|\mu_i|-1,0)}\right], \end{aligned} \tag{5}$$

with $\lambda_{\max} = 0.996$, $\lambda_{\min} = 0.5$, $\beta = 10$: modes with $|\mu_i| \leq 1$ get the scalar rate, growing modes are tracked faster. A rank regularizer with weight $\alpha = \alpha_0 e^{-\gamma r}$, r the number of modes with $|\mu_i| \leq 1$, completes the method. The ablation ladder: A0 plain method; A adds the regularizer ($\alpha_0 = 0.01$); B adds LoRA with scalar EMA throughout; C, D, G use a static, periodic and adaptive spectrum; E is adaptive without the regularizer; F fixes α.

The map never acts. Over a 200-epoch BYOL run, the largest of 38,805 estimated spectral radii was 0.99992; all five committed spectra gave mean update rate $0.004 = 1 - \lambda_{\max}$. The Koopman teacher is therefore numerically the scalar EMA. The adaptive re-estimation trigger never fired either: every re-commit, in BYOL and in two DINO runs, fell on the 1000-step layer re-selection schedule, so C, D and G are the same teacher. The regularized estimator at this setting cannot see growth below $|\mu| \approx 1.05$, and without regularization ill-conditioned windows fabricate growth up to $|\mu| = 781$, so the absence of growing modes is not established by this experiment. The function-space analysis of section 4 explains why no per-mode rule of this kind has anything to act on: the student has one structured component, in coordinates that are already nearly diagonal.

D.1 Forecasting the trajectory: DMD loses to doing nothing

A different use of the operator is to extrapolate the student and use the forecast as the teacher target (a "look-ahead teacher"). This is only worthwhile if the forecast beats trivial baselines. On a real BYOL trajectory we fit on the preceding 200-step window and score each predictor by $\|\hat{\psi}_{t+k} - \psi_{t+k}\| / \|\psi_{t+k} - \psi_t\|$, so that persistence (weights stay put) scores exactly 1 (table 10).

DMD is 20–117$\times$ worse than persistence at every horizon, in line with Brokman et al. (2024), who argue that SGD trajectories are poorly described by sums of exponentials. Linear extrapolation of the last increment, by contrast, beats persistence up to about five steps.

These numbers come from a single BYOL run at step stride. A look-ahead teacher built from a virtual gradient step rather than an operator forecast has been proposed before (Du et al., 2022).

Table 10: Relative forecast error on a BYOL / ResNet-18 / CIFAR-100 LoRA trajectory, step-level sampling. 1.0 = persistence. Lower is better.

horizon k	persistence	linear extrapolation	DMD
1	1.000	**0.414**	117.4
5	1.000	**0.628**	33.7
20	1.000	1.096	20.8
50	1.000	1.685	22.7

E How much the linear-probe protocol moves the numbers

Every cost we report is a paired difference between two runs of one recipe from one seed. Read instead against the recipe's own reference accuracy, the teacher-free runs sit 1.33 points below it at $\beta = 0$ and 12.12 below at $\beta = 0.9$, so the ordering does not turn on which baseline is chosen.

Accuracies are of the teacher encoder unless stated otherwise; at 200 epochs the student's differ from them by at most 0.05 points. The probe trains an SGD head on frozen features with the learning rate chosen on a held-out split.

Absolute accuracies in this literature are not comparable across papers unless the probe is the same. Our first protocol was L-BFGS logistic regression with a fixed inverse regularization $C = 0.316$ on ℓ_2-normalized features; the protocol used throughout the paper trains a linear head with SGD (momentum 0.9, no weight decay, 100 epochs, step decay at 60% and 80%), choosing the learning rate on a held-out 10% of the training set and retraining on the full set. Scoring the *same* frozen features both ways:

Table 11: The probe, not the representation, accounts for up to ten points. Same features, two protocols (CIFAR-100, 200 epochs, the plain recipe of each framework).

	fixed-C logistic	SGD head, held-out lr	kNN-20
BYOL	48.43	59.04	53.34
DINO	47.07	54.13	49.75

The protocol moves BYOL by 10.6 points and DINO by 7.1, and it reverses the ordering of the linear probe and kNN: under the fixed-C probe the linear probe sits *below* kNN, which is the opposite of what the SGD probe gives. What it does not do is reorder the methods: across the nineteen runs of the ablation the two protocols agree at Spearman $\rho = 0.965$, and the SGD probe agrees with kNN at $\rho = 0.912$. Comparisons within this paper are therefore safe under either protocol; comparisons of absolute numbers with other papers are not.

Table 12: The full recovery sweep behind table 1, two seeds; R is the fraction of the cost of removing the teacher that each teacher recovers, with $\lambda = 0.9$ (seed 42) as the reference for both seeds.

teacher	linear (s42 / s1)	kNN (s42 / s1)	R linear / kNN (s42; s1)
$\lambda = 0.9$ (EMA reference)	63.92 / —	59.21 / —	1.00
$\lambda = 0.5$ + two-tap	63.44 / 63.55	58.79 / 58.90	0.96 / 0.98; 0.98 / 0.99
two-tap only ($\lambda = 0$)	62.47 / 62.85	57.73 / 58.29	0.89 / 0.94; 0.93 / 0.96
$\lambda = 0.7$	61.30 / 62.15	55.60 / 56.37	0.80 / 0.85; 0.88 / 0.89
$\lambda = 0.5$	57.90 / 59.11	48.55 / 51.28	0.55 / 0.57; 0.68 / 0.69
no teacher ($\lambda = 0$)	50.53 / 48.92	34.29 / 33.80	0.00

The 400-epoch result does not turn on holding the teacher's momentum at the value the 200-epoch recipe uses, which the reference implementation does not do: raising it to $\lambda = 0.996$ over that budget changes what removing the teacher costs by half a point.

Three practices the lock accounts for. BYOL's reference implementation raises λ with the length of training while lowering the learning rate (DeepMind, 2020), which follows if staleness matters

Table 13: Table 2 with the spreads over repeats and the kNN column. The one ± 0.00 is two runs returning the same count of correct kNN predictions, 26.73 twice, where their linear probes differ by a point.

β	τ	n	teacher (LP/kNN)	no teacher (LP/kNN)	Δ LP	Δ kNN
0.9 (standard)	10	2	63.27±0.07/59.39±0.01	51.78±1.77/36.80±3.54	11.49	22.59
0.5	2	2	63.28±0.74/58.78±0.12	62.60±0.23/57.46±0.46	0.68	1.32
0.8	5	1	64.18/59.31	60.14/52.83	4.04	6.48
0.95	20	1	63.85/59.11	44.01/28.01	19.84	31.10
0	1	4	63.27±0.23/58.66±0.20	62.57±0.10/57.21±0.68	0.69	1.45
0.9, 400 ep	10	1	67.68/65.02	59.81/51.08	7.87	13.94
0, 400 ep	1	1	67.40/64.89	66.64/63.63	0.76	1.26

Table 14: The step-size axis of section 5, BYOL / ResNet-18 / CIFAR-100: the momentum held at $\beta = 0.9$ while the effective step $\eta/(1-\beta)$ is varied, one seed per cell. *rank* is the representation rank of each arm; no arm collapses and every probe's learning rate is interior to its grid.

effective step	teacher	no teacher	Δ	rank (teacher / none)
1	53.07	51.43	1.64	1.33 / 1.36
3	59.31	55.51	3.80	1.36 / 1.44
10	63.22	50.53	12.69	1.55 / 1.49

only against how fast the student still improves. The optimum in λ is broad (Grill et al., 2020; He et al., 2020), as a harmless range of lag gives without a tuned value. And momentum on the projector alone suffices (Pham et al., 2022), since with the encoder unfiltered its oscillation reaches the target far less suppressed.

The ImageNet-1k quartet. BYOL with a ResNet-50 on ImageNet-1k, 40 epochs at batch 512, LARS with weight decay 10^{-6} and ten warmup epochs. The learning rate is $9(1-\beta)$, so the effective step is 9 on both arms, the same construction as the $10(1-\beta)$ of section 5; the teacher's momentum is held fixed at $\lambda = 0.97$ against $\lambda = 0$, and there is one seed per cell. Accuracies are the linear probe and kNN of this appendix.

F What these recipes cost to train

The body compares a recipe with its own teacher removed, which is the comparison that speaks to the mechanism. A practitioner choosing between recipes is asking a different question, and table 15 answers that one: what each costs to train and what it returns. The rows are different networks—projector depth and width, output normalization and predictor shape all differ—so nothing here is evidence about what a teacher does; it is a price list.

Table 15: Training cost and return on CIFAR-100. Step times are model-step times on one H100, batch 256×2 views, 100 timed steps, excluding the input path; parameters are those held in the training graph. The SGD row of the teacher-free recipe is there to separate the optimizer's share of the step time from the teacher's, and was not trained to a probe, hence the dash. FLOPs and parameter counts are identical on any machine. Rows are different networks, so these numbers compare recipes rather than mechanisms.

recipe	optimizer	linear probe	ms / step	params	peak mem
BYOL, EMA teacher	LARS	63.22	42.1	30.8M	3033 MiB
BYOL, no teacher ($\beta = 0$)	LARS	62.57	32.5	16.4M	2589 MiB
BYOL, no teacher ($\beta = 0$)	SGD	—	29.5	16.4M	—
SimSiam ($\beta = 0.9$)	SGD	63.72	29.9	22.7M	—
SimSiam ($\beta = 0$)	SGD	64.36	29.9	22.7M	—

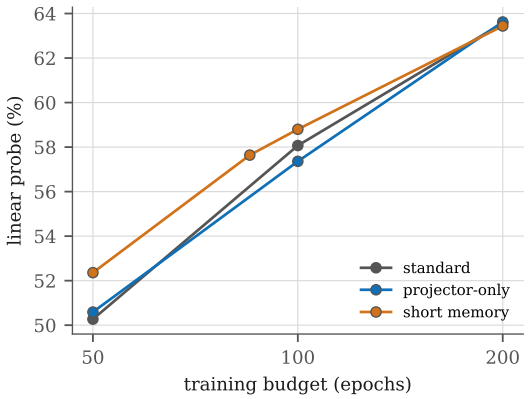


Figure 5: The short-memory teacher buys epochs, not time. BYOL / ResNet-18 / CIFAR-100; linear probe against training budget, one seed per point, each budget a full schedule rather than an early read of a longer run.

Dropping the teacher takes 47% off the training graph and the model step to $1.30\times$ faster, and leaves it smaller than the teacher-free baseline it is usually compared with, at a comparable step time once the optimizer is held fixed, LARS itself costing about a tenth of the step on this model. The model step is not the whole step: with an input path costing about 45 ms here, the end-to-end saving is 2.2%. The ratio is not a property of this card: on an RTX A6000 the same model step takes 122.0 ms against 94.4, the same $1.29\times$. We did not measure that machine's own input floor, which belongs to its CPUs and storage rather than to the model, so this checks the ratios and says nothing about what they are worth end to end on it. And SimSiam scores above both BYOL rows here, which is a fact about those two networks and their recipes rather than about teachers: the within-network statement, that removing the teacher costs 0.69 points at $\beta = 0$, is in section 5.

A floor that FLOPs do not see. Removing the teacher takes 50% off the model's arithmetic, and in this setup almost none of that reaches the clock. The input path costs about 45 ms per step, and does not move when the loader's workers go from 15 to 31 or when the run has the node to itself, while the model step is 42.1 ms, so a $1.30\times$ faster model step is absorbed almost entirely and 2.2% survives end to end. The observation generalises further than the method does: a reduction in arithmetic becomes time only where the step is compute-bound, and on small-image benchmarks it often is not. This is one setup, and the floor it measures belongs to its CPUs and storage rather than to the model.

G TRANSFER-FUNCTION VALUES

Table 16 lists the measured gain, lag and coherence behind fig. 2a for the scheduled-momentum run ($\bar{\lambda} = 0.99599$), on two seeds.

Table 16: The teacher's behaviour follows the analytic first-order response, on the standard BYOL / ResNet-18 / CIFAR-100 recipe, in gain and lag simultaneously. Measured transfer from student to teacher on log eig1 (two seeds) against eq. (2) at $\bar{\lambda} = 0.99599$.

period	gain (s0 / s1)	gain, analytic	lag in steps (s0 / s1)	lag, analytic	coherence
22	0.0145 / 0.0147	0.0141	5.0 / 5.0	4.95	0.97 / 0.98
50	0.0330 / 0.0305	0.0320	11.4 / 11.6	11.75	0.94 / 0.96
200	0.116 / 0.117	0.127	46.5 / 47.2	45.45	0.91 / 0.89

H CELLS WHERE THE GAP VANISHES BECAUSE BOTH RUNS ARE DEGRADED

Removing the student's momentum forces a choice about the learning rate. Matching the effective step $\eta/(1-\beta)$ keeps the recipe in the regime it was tuned for; not matching it runs the recipe at a tenfold smaller step, which costs about ten points at a fixed epoch budget. Table 17 lists those runs. In both settings the teacher's benefit nearly disappears, but neither run reaches the quality of the standard recipe, so the vanishing gap is not evidence that the teacher is unnecessary. This is the empirical reason for the criterion used throughout: a gap may be read only when both runs retain their quality.

The criterion is applied differently to the two. The teacher arm is compared with the recipe's reference accuracy, and its final training loss is expected in the band the healthy runs occupy. The teacher-free run cannot be checked that way: without a teacher the final loss falls monotonically as the student's momentum grows—0.300, 0.289, 0.254, 0.221, 0.205 across the sweep—while accuracy degrades, so a low loss there is the effect under study. It is checked for collapse by representation rank instead, which is the last column of table 2.

Table 17: A vanishing gap without quality is not evidence. BYOL / ResNet-18 / CIFAR-100; linear probe / kNN, where the standard recipe reaches 63.90 ± 0.39 / 59.17 ± 0.33.

setting	effective step	$\lambda = 0.99$	$\lambda = 0$
$\beta = 0.9$, $\eta = 0.1$ (two seeds)	1	53.08 / 45.69	51.90 / 43.47
$\beta = 0$, $\eta = 1$ (seed 42)	1	51.35 / 43.78	51.89 / 44.27

The second seed of the first cell gives a gap of 2.56 / 2.39, the same picture.

Spectral quantities that do not order the teachers. A single-band account is refuted by $\lambda = 0.5$, which suppresses the oscillation band by only $1.08\times$ yet recovers 55% of the no-teacher gap, while the two-tap average suppresses it far more and recovers 89%. Relative bump height, in-band amplitude and total suppression likewise fail to order the teachers across settings.