BREAKING THE STALENESS PHASE: UNDERSTANDING AND ACCELERATING JEPA TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

The exponential moving average (EMA) mechanism, introduced to stabilize self-supervised learning, coincides with a surprising failure mode in Joint-Embedding Predictive Architectures (JEPAs): during early training, the reconstruction loss plummets while learned representations remain useless for downstream tasks. We call this the "staleness" phase. It arises from two interacting effects: the EMA keeps the target encoder tightly coupled to the main encoder in parameter space, and the predictor, which trains much faster than the encoder, fits the targets of the still uninformative encoder, minimizing the objective while starving the encoder of gradient. We derive an upper bound showing that the coupling depends on the interplay between momentum dynamics and augmentation strategy, and that escape requires the encoder's updates to outpace the EMA's smoothing. This reframes staleness not as an objective flaw but as a transient regime with predictable onset and recovery. The analysis suggests a simple fix: perturbing encoder weights at the start of training breaks the initial symmetry between encoder and target, while the perturbation is quickly absorbed by the EMA and vanishes as training matures. Experiments on images and time-series support the analysis and show that noise injection accelerates recovery and improves final performance.

1 INTRODUCTION

Joint-Embedding Predictive Architectures (JEPAs) have emerged as a recent framework for self-supervised learning, offering an alternative to contrastive methods and input-space reconstruction (LeCun, 2022). Rather than pulling together augmented views or predicting masked pixels, JEPAs operate entirely in a learned latent space, where an encoder produces representations that, through a predictor, must match targets generated by a slowly-updating Exponential Moving Average (EMA) encoder (Grill et al., 2020; He et al., 2020). This design helps in ignoring low-level noise and focuses learning semantically meaningful structures. This paradigm has shown strong results in different domains, including images (Assran et al., 2023) and video (Bardes et al., 2024), time series (Ennadir et al., 2025), and tabular data (Thimonier et al., 2025).

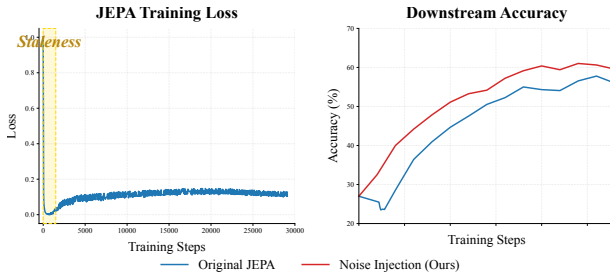


Figure 1: **The JEPA training staleness phase.** A JEPA trained on CIFAR-10 exhibits an early effect: training loss drops rapidly, while downstream accuracy remains at (or below) a randomly initialized encoder. Our noise injection strategy mitigates this phase, accelerating useful representation learning and improving final performance.

Yet JEPA training exhibits a puzzling behavior. Figure 1 illustrates this pattern on CIFAR-10: the reconstruction loss drops rapidly in early epochs while downstream accuracy remains near chance. The model appears to be learning, yet its representations are uninformative. Only after a prolonged period does performance begin to recover, eventually reaching competitive levels. This pattern, which we term the *staleness* phase, recurs across modalities and architectures (Appendix A), suggesting a systematic phenomenon rather than an implementation artifact.

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.

Understanding this effect matters for both theoretical and practical reasons. On the theoretical side, it reveals a gap in our understanding of how EMA-based objectives shape representation learning. The standard intuition holds that the slow-moving target encoder provides stable supervision, preventing the trivial solutions that plague naive self-prediction (Grill et al., 2020; Chen & He, 2021). But if this mechanism is so effective, why does training have a phase where representations carry no information? On the practical side, the staleness represents wasted computation. Training must run longer to escape collapse before useful learning begins, and without understanding what governs this duration, we cannot reliably shorten it.

In this work, we provide a principled analysis of this collapse phase, showing that it arises from two complementary effects. On one side, the EMA keeps the target encoder tightly coupled to the main encoder in parameter space, since both start from the same weights and the target only slowly integrates the encoder's updates. On the other side, the predictor, which trains much faster than the encoder, rapidly fits the targets of the still uninformative encoder, where the optimal linear solution is a constant-like map rather than the identity, thereby minimizing the objective while starving the encoder of gradient. We formalize this coupling effect through the encoder and EMA parameter dynamics, deriving bounds that connect the coupling level to the EMA update parameter and the masking ratio and we validate the predictor effect causally, showing that training only the predictor against a frozen encoder reproduces the loss plunge almost entirely. Our analysis reveals that collapse is not a failure but rather a transient regime induced by conservative EMA updates and fast predictor equilibration during early training. This perspective complements recent theoretical work on JEPA regularization (Balestriero & LeCun, 2025), which focuses on preventing complete collapse through distributional constraints rather than characterizing the transient dynamics studied in this paper.

Guided by these insights, we propose a practical intervention that perturbs the encoder weights during a brief phase at the start of training. This controlled perturbation breaks the initial symmetry between encoder and target, preventing the predictor from settling into trivial solutions and accelerating escape from staleness. We empirically validate our findings on image and time-series data, showing the effectiveness of the strategy in improving downstream performance. Our contributions are as follows.

- We provide a formal characterization of early-representation collapse and training staleness in JEPA taking into account discrepancy between encoder and EMA and effect of the predictor.
- We derive theoretical bounds connecting staleness risk to the EMA update parameter and masking ratio, explaining when and why this effect occurs.
- We propose a practical noise-injection warm-up strategy that accelerates recovery from the staleness phase, and validate its effectiveness across multiple modalities.

2 Related Work

Contrastive self-supervised methods (Chen et al., 2020; He et al., 2020) learn by pulling augmented views together while pushing apart different images; non-contrastive approaches remove the need for negatives via asymmetric predictors (Grill et al., 2020), stop-gradients (Chen & He, 2021), or redundancy reduction (Zbontar et al., 2021). In contrast, JEPAs predict the representation in latent space (LeCun, 2022). For instace, I-JEPA (Assran et al., 2023) predicts embeddings of masked regions from visible context with an EMA target network, extends to video (Bardes et al., 2024), time-series (Ennadir et al., 2025), and tabular data (Thimonier et al., 2025).

Despite strong empirical performance, JEPAs are subject to a number of theoretical challenges around collapse. Joint embedding objectives impose weaker alignment constraints than reconstruction, especially when irrelevant features dominate (Van Assel et al., 2025), yet collapse prevention remains unresolved. EMA alone is insufficient; combining it with variance-covariance regularization helps mitigate collapse (Mo & Tong, 2024), while proving that isotropic Gaussian embeddings minimize downstream risk motivates a regularizer removing the need for momentum targets (Balestriero & LeCun, 2025). Variance-covariance regularization has also been extended with norm alignment to capture higher-order structure (Kuang et al., 2025). Analyses of deep linear networks show that JEPAs implicitly favor features with large regression coefficients, an effect emerging only with sufficient depth (Littwin et al., 2024a), and providing explicit spatial information about context and target positions stabilizes training across hyperparameter ranges (Littwin et al., 2024b). Finally, trained JEPAs implicitly estimate data density, connecting representation learning to probabilistic

modeling (Balestriero et al., 2025). Recent work SALT (Li et al., 2026) have also investigated EMA, proposing to use a frozen encoder pretrained via pixel reconstruction, then training a student to predict the teacher's masked latent representations. Prior work focuses on designing objectives that avoid collapse. Similar trends appear in non-contrastive learning, where analyses emphasize steady-state properties such as eigenspace alignment, dimensional collapse, or regularization mechanisms (Tian et al., 2021; Jing et al., 2022; He et al., 2024). In contrast, we study a transient phenomenon of *staleness* that emerges during early training, where representations temporarily become uninformative before recovering. Our work analyzes this dynamical regime and shows how the EMA decay rate and masking ratio jointly control the duration and severity of this early-training collapse within JEPA.

3 PRELIMINARIES

Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ denote a dataset of N samples; the goal is to learn, without labels, a mapping $f : \mathcal{X} \to \mathcal{H} \subseteq \mathbb{R}^d$ whose representations transfer to downstream tasks. A JEPA partitions the input into observed and masked regions and trains an encoder to predict latent representations of masked content from observed context; let $\mathcal{N}$ (resp. $\mathcal{M}$) index observed (resp. masked) patches. Three components define the architecture.

Encoder f_θ. The encoder maps observed patches to latent representations. Given unmasked patches $\{p_i\}_{i\in\mathcal{N}}$, it produces embeddings $z_\mathcal{N} = f_\theta(\{p_i\}_{i\in\mathcal{N}}) \in \mathbb{R}^{|\mathcal{N}|\times d}$, with d being the dimension. Depending on the application, the encoder may be instantiated as an MLP, CNN, or Transformer.

EMA (Target) Encoder $g_{\bar\theta}$. The target encoder shares the main encoder's architecture but follows a distinct update rule. Rather than receiving gradients directly, its parameters $\bar\theta$ evolve as an exponential moving average of the main encoder's weights:

$$\bar\theta^{(t)} \leftarrow \beta\,\bar\theta^{(t-1)} + (1-\beta)\,\theta^{(t)}, \tag{1}$$

where $\beta \in [0, 1]$ is the EMA decay rate. Values near 1 produce a slowly evolving target, shown to provide stable supervision that is critical for self-supervised learning without negative samples. We assume identical initialization of the encoder and EMA encoder: $\theta^{(0)} = \bar\theta^{(0)} = \theta_0$.

Predictor P_ϕ. The predictor transforms context representations to align with the target encoder's outputs at masked positions. Given encoded context, it produces predictions $\hat z_\mathcal{M} = P_\phi(z_\mathcal{N}) \in \mathbb{R}^{|\mathcal{M}|\times d}$ for the masked patches.

Training objective is minimizing the latent-space discrepancy between predictions and targets:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \|\hat z_i - \bar z_i\|^2,$$

where $\bar z_i = g_{\bar\theta}(p_i)$ is the target representation of masked patch p_i. This objective encourages learning meaningful structure without pixel-space reconstruction.

4 ON THE STALENESS PHASE OF JEPA

As illustrated in Figure 1, JEPA models trained with an EMA-based target encoder often exhibit a staleness phase during training. This phase typically occurs early, within the first epochs (often between 10 and 100). Empirically, it is characterized by a rapid decrease in the reconstruction loss, followed by an increase or stagnation effect, while downstream performance remains low (or degrades). This mismatch between reconstruction loss and downstream performance suggests that the learned representations are temporarily uninformative. In this section, we aim to provide a theoretical understanding and explanation for this phenomenon in the context of EMA-based JEPA training.

4.1 CHARACTERIZING STALENESS

To analyze this behavior, we first need a precise way to characterize and measure this effect. Empirically, staleness corresponds to a regime in which the training objective improves while the learned representations remain uninformative. Formally, let $r^{(t)} = \mathbb{E}\|P_\phi(f(x)) - g(\hat x)\|^2/\mathbb{E}\|g(\hat x)\|^2$ denote the prediction residual, normalized by the target energy, and let $\mathrm{acc}(\theta^{(t)})$ denote the linear-probe

accuracy of the encoder at iteration t. We say that training is in a *staleness phase* when the residual is small, that is $r^{(t)} \leq \delta$ for a threshold δ, while the probe fails to improve over a randomly initialized encoder, that is $\mathrm{acc}(\theta^{(t)}) \leq \mathrm{acc}(\theta^{(0)})$. The *staleness duration* is the first iteration at which training exits this regime. Both criteria are measurable (as illustrated in Figures 1 and 2). In this work, we hypothesize that this regime arises from the interplay between the early training dynamics of the encoder, its EMA target and the predictor.

During the initial epochs, the parameters of the encoder and the EMA encoder remain very close, empirically within a few percent in relative norm (Section 6.1). Parameter closeness, however, controls only part of the output discrepancy between the two networks. Their outputs also depend on how the encoder responds to the two views, and early features concentrate in a low-rank subspace whose direction evolves faster than the EMA tracks it. Our analysis is therefore extended to take into account the predictor and its effect of the collapse. Specifically, since the predictor can equilibrate much faster than the encoder, it minimizes the reconstruction loss with a trivial, constant-like solution together with a quickly learned correction, independently of any meaningful structure in the encoder. This yields a low reconstruction error, but it also starves the encoder of gradient and produces representations that lack semantic structure and are therefore ineffective for downstream tasks. We formalize this two-part mechanism as follows.

Mechanism. *(i) Coupling:* the EMA keeps the target encoder close to the encoder in parameter space during early training. *(ii) Absorption:* the predictor, whose optimization given current encoders is fast, converges to a trivial solution (empirically a constant-like map with a rapidly learned correction) that minimizes the loss regardless of representation quality, starving the encoder of gradient.

We start by considering the first part (i), where we track the encoder–EMA output discrepancy during training. Given two views $x, \hat{x} \sim \mathcal{D}_{\mathcal{X}}$, we define the coupling functional:

$$\mathcal{C}_{f,g} = \mathbb{E}_{x,\hat{x}\sim\mathcal{D}_{\mathcal{X}}}\left[d_{\mathcal{Y}}\big(f(x), g(\hat{x})\big)\right], \tag{2}$$

where $d_{\mathcal{Y}}$ is a representation-space distance (ℓ_2 in our analysis). We note that this formulation covers masking and augmented-view settings alike: x and $\hat{x}$ denote two views of the same input, assumed close in input space, $d_{\mathcal{X}}(x, \hat{x}) = \|x - \hat{x}\| \leq \epsilon$, where $\epsilon > 0$ depends on the masking strategy (a connection we make explicit in Section 4.3). Overall, small values of $\mathcal{C}_{f,g}$ reflects a tightly coupled encoder–EMA pair, resulting in a simple regime that can easily reach small loss values through a trivial predictor.

Definition 4.1 (γ-coupled regime)**.** The encoder–EMA pair is γ-*coupled* at epoch t if $\mathcal{C}_{f,g}^{(t)}(\epsilon) \leq \gamma^{(t)}$.

Directly computing $\mathcal{C}_{f,g}$ is generally intractable without strong assumptions. Therefore, in practice, we rely on upper bounding this quantity. A smaller bound $\gamma^{(t)}$ indicates a smaller discrepancy between the encoder and EMA encoder outputs, and thus a higher likelihood of staleness at epoch t.

4.2 Connecting Staleness to Hyperparameters

Building on the previous characterization of staleness, we now derive an upper bound on the staleness risk γ and analyze its dependence on the model hyperparameters. While Definition 4.1 is general and applies to a broad class of encoders, we consider a simplified setting that enables a tractable theoretical analysis while remaining representative of standard JEPA formulations used in practice.

Problem Setup. We model the encoder as a single-layer MLP with a 1-Lipschitz activation (satisfied by ReLU and TanH (Virmaux & Scaman, 2018)), an assumption common in theoretical analyses of self-supervised learning (Van Assel et al., 2025); the setting captures the essential optimization dynamics and extends to more expressive architectures. Under this setup, the collapse quantity is controlled by two interacting factors: the sensitivity of the representations to masking (an input-space perturbation), governed by the Lipschitz constant $L_f^{(t)}$, and the parameter mismatch induced by the delayed EMA updates (detailed in Appendix B). Analyzing the EMA update dynamics yields the following bound on the coupling level.

Theorem 4.2. *Let f satisfy the above setup; then the encoder–EMA pair is γ-coupled with*

$$\gamma = L_f^{(t)}\epsilon + \|\beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\Delta^{(\ell)}\|\|\hat{x}\|,$$

where $\Delta^{(t)}$ denotes the encoder weight update at epoch t, and $\beta \in [0, 1]$ is the EMA decay parameter.

Theorem 4.2 decomposes the output difference into two channels, namely an *input-geometry* term $L_f^{(t)}\epsilon$ and a *parameter-coupling* term governed by the deviation between the current update $\beta\Delta^{(t)}$ and the EMA's exponentially weighted history. Early in training, updates are small and smooth, so the resulting coupling term stays small. The coupling channel is then open and the fast-equilibrating predictor closes the loss with a constant-like solution, sustaining staleness, while the measured discrepancy of Section 6.1 is driven by the input-geometry term. As updates grow or change direction, the mismatch with the EMA history increases and the model escapes this regime.

Effect of the EMA parameter. The parameter β enters the bound through the deviation between the current update $\beta\Delta^{(t)}$ and the exponentially weighted history retained by the target, and the two limits of this term are instructive. At $\beta = 0$ the target copies the encoder at every step and the coupling term vanishes identically, so the coupling channel stays open regardless of how the encoder moves. At $\beta = 1$ the target is frozen at initialization and the term reduces to the encoder's own accumulated motion, which grows as soon as learning begins. The bound therefore predicts that a slower target opens the coupling channel earlier along the realized trajectory, while a fast target keeps it closed. We empirically notice this (Section 6), with the escape time decreasing monotonically in β. Overall, staleness is not an inherent failure mode of JEPA training but a transient regime governed by how quickly the target releases the encoder.

4.3 Connecting Staleness to Masking

Having connected the coupling channel perspective to the EMA dynamics, we now study how the masking strategy itself is also a parameter controlling this coupling. Without loss of generality, we focus on the I-JEPA setting, where $\hat{x} \in \mathbb{R}^{N\times d}$ denote the full sequence of token embeddings extracted from an input image, with each row $\hat{x}_i \in \mathbb{R}^d$ corresponding to a token. We assume that token embeddings are uniformly bounded, i.e., $\|\hat{x}_i\|_2 \leq B \quad$ for all $i \in [0, N]$, which is realistic due to input normalization and bounded positional encodings.

In I-JEPA, masking is applied by selecting contiguous blocks of tokens, with an expected masking ratio $p \in [0, 1]$. Let $M \in \{0, 1\}^{N\times N}$ denote the diagonal masking operator, where masked tokens are retained and visible tokens are removed. Under this formulation, the masked input can be written as $x = M\hat{x}$, while $\hat{x}$ represents the unmasked input. This induces an input-space discrepancy that depends explicitly on the masking ratio p. According to this formulation, we can refine the coupling bound by incorporating the effect of masking.

Proposition 4.3. *Under the assumptions of Theorem 4.2, for any input $\hat{x}$ and masking with expected coverage p, the encoder–EMA pair is γ-coupled at epoch t, with*

$$\gamma = L_f^{(t)}\, B\sqrt{pN} + \|\beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\Delta^{(\ell)}\|\,\|\hat{x}\|.$$

The bound highlights two distinct masking controls on the staleness mechanism. The first term arises from the input mismatch induced by masking, scaling with the masking ratio p, the number of tokens N, and the Lipschitz constant $L_f^{(t)}$. The second term corresponds to the parameter mismatch between the encoder and the EMA encoder, as characterized previously. Specifically, the bound implies:

- Increasing the masking ratio p amplifies the input mismatch term, which can delay escape from the staleness phase by increasing the overall upper-bound.
- Contiguous block masking, as used in I-JEPA, induces spatial correlations between masked tokens that increase variance in staleness dynamics across runs.

Overall, the effects of masking and EMA dynamics depend on the interaction between the masking ratio p, EMA decay β and encoder smoothness $L_f^{(t)}$, suggesting a complementary control over training staleness across different training stages.

Remark. In practice, Transformer-based architectures such as I-JEPA operate only on visible tokens by extracting them into a shorter sequence, rather than explicitly forming the zero-padded

representation $(I - M)\hat{x}$ used in our analysis. However, masking operators are non-expansive, i.e.:

$$\|(I - M)v\| \leq \|v\| \quad \text{for any } v,$$

which ensures that our bound on $\|M\hat{x}\|$ provides a valid upper bound on the discrepancy between masked and full representations, even if it is not tight in this setting.

5 Noising as a Staleness Accelerator

Previous sections analyzed the origin and dynamics of training staleness in JEPA pre-training. Although this staleness is a transient phase and JEPA methods achieve strong empirical performance, shortening this phase could improve both representation quality and training efficiency, as early training produces representations with limited utility. Our analysis identifies two factors driving this phenomenon: (i) an input mismatch induced by the masking strategy and (ii) a parameter mismatch governed by the EMA dynamics. In the following, we focus on controlling the parameter mismatch in the early training phase, beyond standard EMA regularization.

Weight-noise injection. The analysis indicates that a primary driver of staleness is the strong encoder–EMA coupling during early training: both networks start identical, and the EMA closely tracks the encoder in the initial epochs, keeping the parameter mismatch small and enabling the trivial predictor solution. Motivated by this, we relax the coupling by perturbing the encoder weights early in training. We propose to apply a set of random perturbation during the initial phase of training:

$$\theta^{(0)} = \theta_0 + \eta, \qquad \bar{\theta}^{(0)} = \theta_0, \tag{3}$$

where η denotes a noise perturbation. Following the same derivation as in the noiseless case, the encoder and EMA difference at epoch t can be formulated as the following:

$$\theta^{(t)} - \bar{\theta}^{(t)} = \beta^t \eta + \beta \Delta^{(t)} - (1 - \beta) \sum_{\ell=1}^{t-1} \beta^{t-\ell} \Delta^{(\ell)}, \tag{4}$$

where $\Delta^{(\ell)} = \theta^{(\ell)} - \theta^{(0)}$ measures deviation from the perturbed encoder initialization. The additional term $\beta^t \eta$ introduced by the noise has two main properties. First, it increases the parameter mismatch between the encoder and the EMA during the early epochs, precisely when the learning-driven terms $\Delta^{(\ell)}$ are still small. Second, it decays exponentially with training time at a rate set by the EMA parameter β, so that the injected gap is absorbed within $\mathcal{O}(1/(1 - \beta))$ iterations, a few hundred for typical values such as $\beta \approx 0.996$, and thus well before the staleness phase ends. The lasting effect of the perturbation is therefore best understood as a *symmetry breaking of the initial condition* rather than a sustained decoupling of encoder and target. The noise changes where the coupled system of encoder, EMA, and predictor starts relative to the trivial regime, while leaving the long-term training dynamics asymptotically unaltered. This interpretation is supported by our ablations in Table 2, where performance is largely insensitive to the noise magnitude across an order of magnitude, while very small perturbations, too weak to break the initial symmetry, rarely improve over the baseline.

Practically, the intervention is lightweight and incurs no additional computational overhead. The perturbations are sampled from a Gaussian distribution $\mathcal{N}(0, \Sigma)$, with the covariance Σ scaled per layer to the magnitude of the corresponding parameters. The noise is applied during an initial phase whose duration is a fraction ρ of the total training budget, chosen commensurate with the share of training that the staleness phase is expected to occupy, so that $\rho = 10\%$ for instance injects noise during 10% of training. Because each perturbation is absorbed on the fast timescale identified above, the single-perturbation analysis applies to every perturbation in the phase, and repeated injection maintains the symmetry breaking until the learning-driven terms dominate.

Beyond noise injection, the derived bound suggests the masking strategy as a second, complementary lever. Its input mismatch term scales as $L_f^{(t)} B \sqrt{pN}$, and although this term governs the view discrepancy $\|f(x) - f(\hat{x})\|$ rather than the parameter mismatch, the masking ratio p also controls the difficulty of the prediction task. From this perspective, staleness can be read as a failure mode of an overly easy early regime, in which low masking ratios provide ample context and easy prediction while the encoder and its target have not yet diverged. A time-varying schedule $p(t)$ that starts high and decreases over training would make the trivial constant-like solution less attainable during the critical early epochs, pushing the encoder toward more discriminative representations, and would then

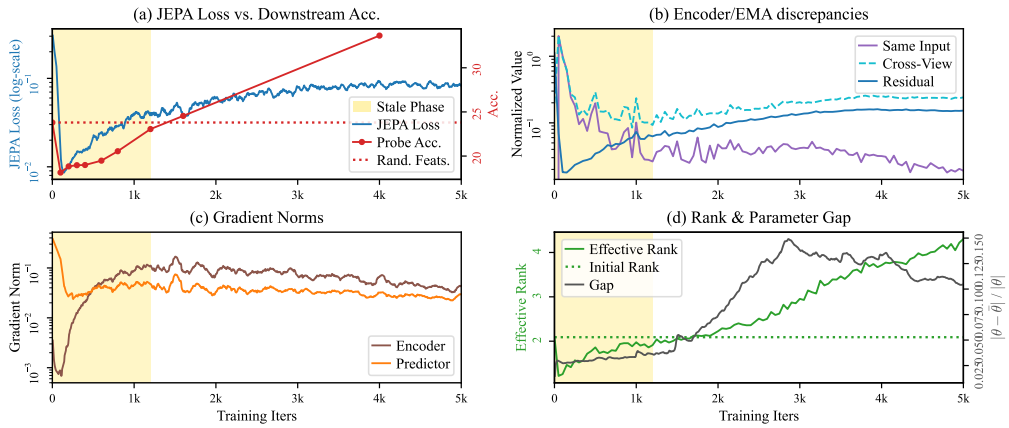


Figure 2: **Anatomy of the staleness phase.** Yellow region identifies staleness, as the linear-probe performance is below initialization. (a) Pre-training loss drops $35\times$ in 100 iterations while probe performance degrades below the random-feature baseline. (b) Encoder–EMA outputs diverge (cosine ≈ 0) despite a small parameter gap ($\approx 3\%$), while $\|P(f(x)) - g(\hat{x})\|$ collapses, showing predictor compensation. (c) Encoder gradients remain small (initially $\sim 100\times$ below predictor gradients) until escape. (d) Effective rank transiently drops (2.1 to 1.2) and recovers near escape as the encoder–EMA parameter gap opens. Feature norms remain stable, excluding scale collapse.

encourage finer-grained prediction and representation refinement as the learning-driven mismatch grows. Nonetheless, this strategy needs careful scheduling and can therefore introduce additional hyperparameters and therefore we prefer our noise injection technique as it's not expensive and directly applicable given its simplicity.

6 Experimental Validation

We now aim to validate empirically the derived insights and show: (i) the validity of our introduced staleness assumption, (ii) the validity of our insights regarding the EMA parameters and role of predictor and (iii) the effectiveness of our proposed noise injection to accelerate this phase.

Experimental Setting. Following evaluation protocol from previous work, in our experiments we used a ViT-Base, and the evaluation is done through linear probing, where the encoder is frozen and only a linear classification head is trained. Our code is provided in the supplementary materials and additional details about the hyper-parameters and experimental setting are provided in Appendix F.

6.1 On the Staleness Hypothesis

As discussed in Section 4.1, our analysis considers a two-part mechanism, namely a tight coupling between the encoder and the EMA in parameter space and a fast predictor that *absorbs* the prediction task, starving the encoder. We examine both parts experimentally on the CIFAR-10 dataset, where we compute all the metrics every 50 iterations and Figure 2 provides the results of the considered metrics. We clearly see in panel (a) that the JEPA loss collapses while the probe *degrades below the accuracy of a randomly initialized encoder*, showing that the loss is uninformative about representation quality. The coupling channel behaves exactly as Theorem 4.2 describes. The relative parameter gap between the encoder and the EMA stays near 3%, as shown in panel (b), and the encoder representations remain almost identical to those at initialization, with a centered kernel alignment (CKA) (Kornblith et al., 2019) of ≈ 0.95 (1 indicates no change). Yet (b) also shows the *output* discrepancy spiking at training start, as early features occupy a transiently low-rank subspace, visible in (d). Consequently, the loss collapses because the predictor, whose gradients exceed the encoder's by two orders of magnitude in panel (c), absorbs the gap and drives the residual to ≈ 0.02 in panel (b) while the encoder remains gradient-starved. Escape is a single coherent transition visible across all four panels, in which gradients awaken, the parameter gap doubles, the rank recovers, the probe recovers and the JEPA loss *rises*. The observed results shows and confirm the overall interaction between the coupling effect of the encoder/EMA and the predictor.

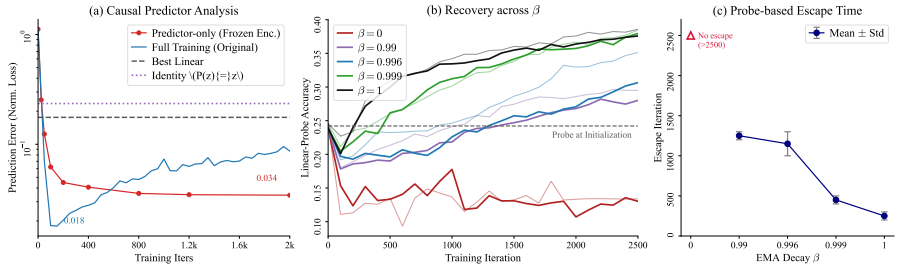


Figure 3: **Left (a): causal analysis of staleness** (CIFAR-10). Prediction error is the anchor-set loss normalized by target energy, $\|P_\phi(f(x)) - g(\hat{x})\|^2 / \mathbb{E}\|g(\hat{x})\|^2$. Predictor-only training with a frozen random encoder (red) reproduces the full-training plunge (blue) with zero encoder motion. Horizontal lines denote the optimal linear/mean predictor and identity $P(z) = z$. **Right: effect of EMA parameter** β on recovery: (b) linear-probe accuracy; (c) escape time (first iteration above initialization). β controls staleness duration: $\beta = 0$ collapses, whereas $\beta = 1$ recovers fastest in this window.

6.2 On the Predictor's Role in Staleness

The previous section examined the first of the two complementary effects, namely the encoder and EMA coupling formalized by Theorem 4.2. We now turn to the second effect and validate the predictor's role causally. We freeze the encoder, and hence the EMA target, at its random initialization and train *only* the predictor, so that any drop in the objective can only come from the predictor itself. Figure 3 (left) shows that this setting reproduces the loss plunge almost entirely. The normalized residual falls from 1.12 to 0.06 by iteration 100, matching the timing of the full run and asymptotes at 0.034, within a factor of two of the full-training trough of 0.018, despite zero encoder motion. The plunge therefore requires no encoder learning, establishing the predictor as the complementary driver of staleness alongside the coupling channel.

6.3 On the Effect of Hyperparameters

We now test the effect of the EMA parameter β. The limiting cases of Theorem 4.2 predict that a fast target keeps the coupling channel closed, the coupling term vanishing identically at $\beta = 0$, while a slow target opens it through the encoder's own evolution. We train across classily used values in practice for β, using multiple seeds to reduce the effect of randomness and measure the probe-based escape time, namely the first iteration at which the probe exceeds its initialization value. Figure 3 (right) confirms the predicted direction, with the escape time decreasing monotonically from roughly $1{,}250$ iterations to 250, while $\beta = 0$ never escapes the considered window. The two limits illustrate the mechanism. At $\beta = 0$ the target copies the encoder at every step, the residual is trivially zero, and the pair co-collapses to a rank-one representation stuck below the random-init probe, the coupling channel in its purest form. At $\beta = 1$ the frozen target escapes staleness fastest, yet it anchors learning to a random objective whose quality never improves, which is expected to collapse the representation and therefore resulting in a worse final performance in the standard full-training regime; the choice of β therefore trades a shorter staleness phase against target quality at longer horizons.

6.4 Empirical Analysis of the Noise Injection

We evaluate the effectiveness of the proposed noise injection mechanism. For images, models are pre-trained on ImageNet-1K or ImageNet-100, and evaluated on CIFAR-10/100, Oxford-IIIT Pets and ImageNet-100 with linear probing, similar to I-JEPA, where only a linear classifier is trained on top of a frozen encoder. Table 1 reports the corresponding downstream accuracies. Across all evaluated datasets, noise injection consistently improves downstream performance. In particular, gains of up to 2% are observed compared to the original EMA-based regularization. These improvements are consistent with the symmetry-breaking account of Section 5: the perturbation alters where the coupled system starts relative to the trivial regime (as reflected in the early loss landscape on ImageNet; Appendix E), allowing the encoder to enter the representation-learning regime sooner. These tables report downstream accuracy; directly quantifying the reduction in staleness *duration*, with seed-level uncertainty, is a complement our diagnostics (Section 4.1) enable. Low-shot evaluation (10% of

Table 1: Downstream accuracy (%). Left: ImageNet-1K pre-training transfer learning and low-shot (10% of labels). Right: ImageNet-100 and time-series pre-training (FordA, FaultDetectionA); "In-d." denotes evaluation on the pre-training dataset. "Ours" is the proposed noise injection.

Pre-tr.	Setting	Dataset	Orig.	Ours
IN-1K	Transfer	CI-10	90.83	**92.24**
		CI-100	74.92	**76.10**
		Pets	87.93	**88.36**
		IN-100	81.94	**82.44**
	Low-shot	CI-10	84.91	**85.30**
		CI-100	59.44	**59.79**
		Pets	76.51	**79.45**

Pre-tr.	Setting	Dataset	Orig.	Ours
IN-100	Transfer	CI-10	66.22	**71.52**
		CI-100	47.29	**50.16**
		IN-100	59.92	**62.76**
FordA	In-d.	FordA	90.61	**91.93**
	Transfer	FordB	73.23	**78.94**
Fd-A	In-d.	Fd-A	95.64	**97.80**
	Transfer	Fd-B	50.39	**54.73**

labels) shows the same improvements. We additionally observe that the same enhancement trend for the time-series results where models pre-trained on FordA (resp. FaultDetectionA) are evaluated in-distribution and on FordB (resp. FaultDetectionB) following the TS-JEPA protocol (Table 1).

Ablation Study on Noise Injection. Since our proposed method relies on noise injection, selecting an appropriate perturbation schedule is crucial to preserve meaningful semantics during pre-training. We therefore conduct a joint ablation over the total training budget T, the duration ρ of the initial noise-injection phase, and the perturbation magnitude σ. The results are provided in Table 2. Across all configurations, the method remains robust, with relatively small performance variations between the best and worst settings, indicating that the gains do not depend on a fragile hyperparameter choice. We observe that the optimal schedule depends on the available training budget: shorter runs favor smaller noise-injection phases, while longer schedules benefit from a larger ρ combined with a moderate perturbation strength. In particular, $\sigma = 2.10^{-2}$ consistently provides the strongest and most stable results across ImageNet-100, CIFAR-10, and CIFAR-100, whereas very small perturbations rarely achieve the best performance. Overall, the results suggest a clear and intuitive sweet spot for the proposed strategy: a moderate noise magnitude together with a noise-injection duration that scales with the total pre-training budget.

Table 2: Ablation of the noise-injection schedule on a ViT-Base pre-trained on ImageNet-100: budget T, injection-phase duration ρ (fraction of T), and magnitude σ.

		$T = 2500$			$T = 5000$			$T = 7500$		
ρ	σ	IN-100	CI-10	CI-100	IN-100	CI-10	CI-100	IN-100	CI-10	CI-100
0.1%	10^{-1}	60.36	70.90	49.06	60.82	71.98	49.92	60.98	71.05	49.02
0.1%	$2{\cdot}10^{-2}$	62.22	**72.26**	**50.60**	60.76	72.15	49.98	61.94	70.97	48.64
0.1%	$2{\cdot}10^{-3}$	60.48	71.24	49.90	60.40	71.90	49.87	61.74	71.87	49.96
1%	10^{-1}	62.04	71.90	49.95	61.60	71.61	49.88	60.56	71.13	49.63
1%	$2{\cdot}10^{-2}$	**62.38**	71.32	50.21	61.60	72.08	50.05	60.72	71.23	50.28
1%	$2{\cdot}10^{-3}$	60.68	71.27	49.46	61.30	71.38	49.26	60.54	70.99	49.40
10%	10^{-1}	61.19	71.52	50.16	**62.76**	71.45	50.37	62.02	71.91	50.17
10%	$2{\cdot}10^{-2}$	62.32	72.15	50.42	61.78	70.96	50.04	**63.46**	**72.67**	**51.07**
10%	$2{\cdot}10^{-3}$	60.60	71.02	49.15	61.60	**72.43**	**50.47**	60.80	71.30	49.59

7 Conclusion

JEPA models often spend early training minimizing their objective while producing representations that fail downstream evaluations. We traced this behavior to two main interacting mechanisms, one of which was supposed to prevent it. Namely, the EMA keeps encoder and target tightly coupled in parameter space and the fast predictor absorbs the prediction task, settling at the linear level into a constant-like solution and starving the encoder of gradient. Probes can even fall below a randomly initialized encoder while a transient rank collapse unfolds. Our theoretical analysis provided an upper bound that directly disentangles the coupling channel, with one term capturing how quickly the encoder diverges from its EMA pair and the other how masking modulates prediction difficulty; together they explain why staleness resolves and what prolongs it. In practice, a brief weight-perturbation phase through noise injection at the start of training, with no architectural or loss changes, breaks this symmetry and improves final representations across the modalities we study.

AI USE STATEMENT

In this work, we used generative AI tools for editing and polishing text mainly, with some additional assistance in literature exploration and overview. We have not used generative AI tools for autonomously generating research hypotheses or experimental results and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete proofs of all theoretical results are provided in Appendix B. Datasets, architectures, and hyperparameters used in all experiments are detailed in Appendix F. The code to reproduce our experiments is also provided in the supplementary materials.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Randall Balestriero, Nicolas Ballas, Mike Rabbat, and Yann LeCun. Gaussian embeddings: How jepas secretly learn your data density. *arXiv preprint arXiv:2510.05949*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Sofiane Ennadir, Siavash Golkar, and Leopoldo Sarra. Joint embeddings go temporal. *arXiv preprint arXiv:2509.25449*, 2025.

Sofiane ENNADIR, Levente Zólyomi, Oleg Smirnov, Tianze Wang, John Pertoft, Filip Cornell, and Lele Cao. Pool me wisely: On the effect of pooling in transformer-based models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8uhXfdSJmA.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization. *Advances in Neural Information Processing Systems*, 37:95579–95606, 2024.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yilun Kuang, Yash Dagade, Deep Chakraborty, Erik Learned-Miller, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Radial-vcreg: More informative representation learning through radial gaussianization. In *NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations*, 2025.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, volume 3 1, 2016.

Xianhang Li, Chen Huang, Chun-Liang Li, Eran Malach, Joshua M. Susskind, Vimal Thilak, and Etai Littwin. Rethinking JEPA: Compute-efficient video self-supervised learning with frozen teachers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3cB9243E9i.

Etai Littwin, Omid Saremi, Madhu Advani, Vimal Thilak, Preetum Nakkiran, Chen Huang, and Joshua Susskind. How jepa avoids noisy features: The implicit bias of deep linear self distillation networks. *Advances in Neural Information Processing Systems*, 37:91300–91336, 2024a.

Etai Littwin, Vimal Thilak, and Anand Gopalakrishnan. Enhancing JEPAs with spatial conditioning: Robust and efficient representation learning. In *NeurIPS 2024 Workshop: Self-Supervised Learning - Theory and Practice*, 2024b. URL https://openreview.net/forum?id=IAqwCSv7kI.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Shentong Mo and Shengbang Tong. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. *Advances in neural information processing systems*, 37: 2348–2377, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Hugo Thimonier, José Lucas De Melo Costa, Fabrice Popineau, Arpad Rimmel, and Bich-Liên DOAN. T-JEPA: Augmentation-free self-supervised learning for tabular data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gx3LMRB15C.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pp. 10268–10278. PMLR, 2021.

Hugues Van Assel, Mark Ibrahim, Tommaso Biancalani, Aviv Regev, and Randall Balestriero. Joint embedding vs reconstruction: Provable benefits of latent space prediction for self supervised learning. *arXiv preprint arXiv:2505.12477*, 2025.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Supplementary Material
Breaking the Staleness Phase: Understanding and Accelerating JEPA Training

A Staleness Effect - Time Series

The staleness phenomenon is not limited to image-based JEPA models. We observe similar behavior in the time-series setting when applying TS-JEPA to the FordA and FaultDetectionA datasets. In these experiments, the pre-training dynamics exhibit the same characteristic pattern observed in I-JEPA: the pre-training loss initially decreases while downstream classification accuracy remains low, followed by a recovery phase in which downstream performance improves.

Figure 4 illustrates this behavior. The left panel reports the evolution of the pre-training loss, while the right panel shows the corresponding downstream accuracy. Together, these results confirm the presence of a staleness phase in time-series JEPA models, indicating that the phenomenon is not modality-specific but rather a general feature of EMA-based JEPA training.

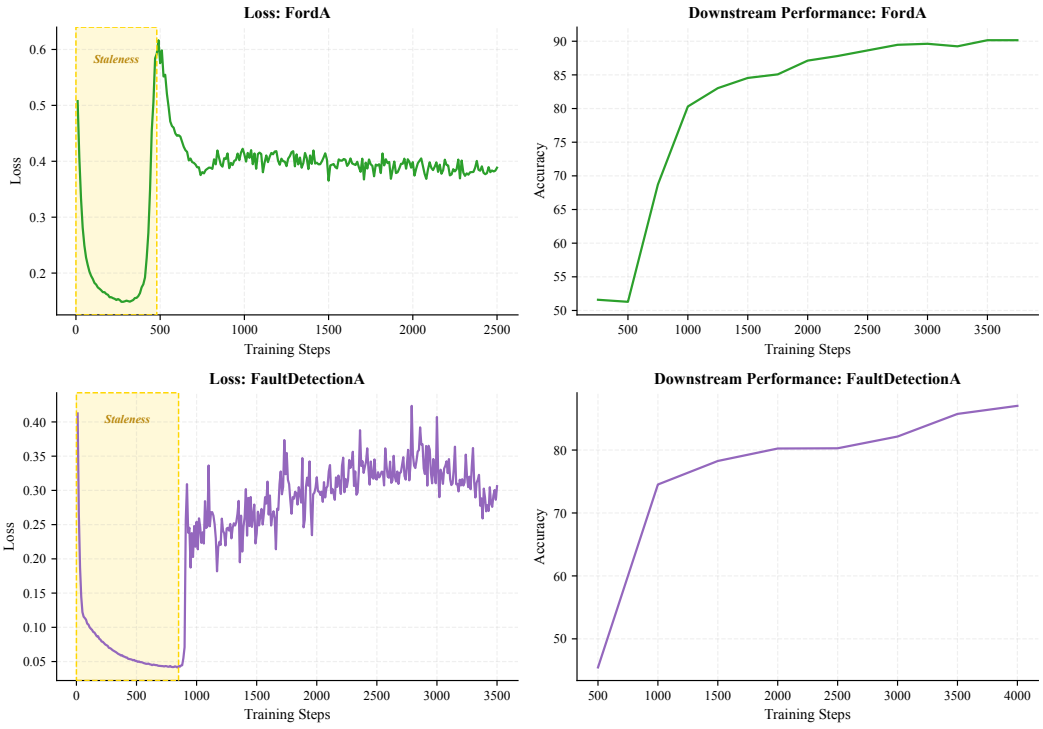


Figure 4: Illustration of a JEPA pre-training on time-series (FordA and FaultDetectionA), where the left side plot shows the pre-training JEPA loss and the right side shows the downstream accuracy performance.

B PROOF OF THEOREM 4.2

Theorem. *Let f satisfy the above setup; then the encoder–EMA pair is γ-coupled with*

$$\gamma = L_f^{(t)}\epsilon + \|\beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\Delta^{(\ell)}\|\|\hat{x}\|,$$

where $\Delta^{(t)}$ denotes the encoder weight update at epoch t, and $\beta \in [0,1]$ is the EMA decay parameter.

Proof. We start with the simpler case where the encoder is an MLP consisting of one layer followed by an activation function σ that is 1-Lipschitz.

For $f(\hat{x}) = \sigma(\theta^{(t)}\hat{x})$ and $g(\hat{x}) = \sigma(\bar{\theta}^{(t)}\hat{x})$, using the Lipschitz property of σ:

$$\|f(\hat{x}) - g(\hat{x})\| = \|\sigma(\theta^{(t)}\hat{x}) - \sigma(\bar{\theta}^{(t)}\hat{x})\| \leq \|\theta^{(t)}\hat{x} - \bar{\theta}^{(t)}\hat{x}\| \tag{5}$$

Applying the submultiplicative property of the operator norm:

$$\|f(\hat{x}) - g(\hat{x})\| \leq \|\theta^{(t)} - \bar{\theta}^{(t)}\|\|\hat{x}\| \tag{6}$$

EMA unrolling: Consider the EMA update function after epoch t with parameter β. The update can be formulated as:

$$\bar{\theta}^{(t)} = \beta\bar{\theta}^{(t-1)} + (1-\beta)\theta^{(t)}. \tag{7}$$

Recursively unrolling the previous equation gives the following closed form:

$$\bar{\theta}^{(t)} = \beta^t\bar{\theta}^{(0)} + (1-\beta)\sum_{\ell=1}^{t}\beta^{t-\ell}\theta^{(\ell)}. \tag{8}$$

To compute $\theta^{(t)} - \bar{\theta}^{(t)}$, we first separate the $\ell = t$ term from the sum (noting that $\beta^{t-t} = 1$):

$$\begin{aligned}
\theta^{(t)} - \bar{\theta}^{(t)} &= \theta^{(t)} - \beta^t\bar{\theta}^{(0)} - (1-\beta)\sum_{\ell=1}^{t}\beta^{t-\ell}\theta^{(\ell)} \\
&= \theta^{(t)} - \beta^t\bar{\theta}^{(0)} - (1-\beta)\theta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\theta^{(\ell)} \\
&= \beta\theta^{(t)} - \beta^t\bar{\theta}^{(0)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\theta^{(\ell)}
\end{aligned} \tag{9}$$

By assumption $\theta^{(0)} = \bar{\theta}^{(0)} = W^{(0)}$, we define the weight deviation at epoch ℓ as:

$$\Delta^{(\ell)} := \theta^{(\ell)} - W^{(0)}$$

Substituting $\theta^{(\ell)} = W^{(0)} + \Delta^{(\ell)}$ into Equation (9):

$$\begin{aligned}
\theta^{(t)} - \bar{\theta}^{(t)} &= \beta(W^{(0)} + \Delta^{(t)}) - \beta^t W^{(0)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}(W^{(0)} + \Delta^{(\ell)}) \\
&= \underbrace{W^{(0)}\left[\beta - \beta^t - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\right]}_{\text{initialization term}} + \underbrace{\beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1}\beta^{t-\ell}\Delta^{(\ell)}}_{\text{learning dynamics term}}
\end{aligned} \tag{10}$$

The initialization term vanishes: We now show that the coefficient of $W^{(0)}$ equals zero. By applying the change of variables $j = t - \ell$, we obtain:

$$\sum_{\ell=1}^{t-1} \beta^{t-\ell} = \sum_{j=1}^{t-1} \beta^j = \beta \cdot \frac{1-\beta^{t-1}}{1-\beta}$$

Therefore:

$$(1-\beta)\sum_{\ell=1}^{t-1} \beta^{t-\ell} = \beta(1-\beta^{t-1}) = \beta - \beta^t$$

Substituting back into the initialization term coefficient:

$$\beta - \beta^t - (1-\beta)\sum_{\ell=1}^{t-1} \beta^{t-\ell} = \beta - \beta^t - (\beta - \beta^t) = 0$$

Consequently, we can write the final bound as:

$$\theta^{(t)} - \bar{\theta}^{(t)} = \beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1} \beta^{t-\ell}\Delta^{(\ell)} \tag{11}$$

Combining with Equation (6) concludes the proof:

$$\gamma = \left\| \beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1} \beta^{t-\ell}\Delta^{(\ell)} \right\| \|\hat{x}\| \tag{12}$$

□

C Proof of Proposition 4.3

Proposition. *Under the assumptions of Theorem 4.2, for any input $\hat{x}$ and masking with expected coverage p, the encoder–EMA pair is γ-coupled at epoch t, with*

$$\gamma = L_f^{(t)}\, B\sqrt{pN} + \|\beta\Delta^{(t)} - (1-\beta)\sum_{\ell=1}^{t-1} \beta^{t-\ell}\Delta^{(\ell)}\|\, \|\hat{x}\|.$$

Proof. Let $\hat{x} \in \mathbb{R}^{N\times d}$ denote the full input, where each row $\hat{x}_i \in \mathbb{R}^d$ represents a token embedding. We assume $\|\hat{x}_i\| \le B$ for all tokens $i \in [N]$, which holds due to input normalization and bounded positional encodings commonly used in practice.

Let $M = \mathrm{diag}(m_1, \ldots, m_N)$ be a binary masking operator, where $m_i = 1$ for masked (removed) tokens and $m_i = 0$ for visible (retained) tokens. The visible input is then given by:

$$x = (I - M)\hat{x}$$

This operation retains the visible tokens in their original positions while zeroing the masked tokens.

Bounding the input mismatch: The distance between the visible and full inputs is:

$$\|x - \hat{x}\| = \|(I-M)\hat{x} - \hat{x}\| = \|M\hat{x}\| \tag{13}$$

For the masked portion $M\hat{x}$, we can derive the following bounds. First, note that:

$$\|M\hat{x}\|_F^2 = \sum_{i=1}^{N} m_i \|\hat{x}_i\|^2 \le |M| B^2 \tag{14}$$

where $|M| = \sum_{i=1}^{N} m_i$ is the number of masked tokens. Using the relationship between spectral and Frobenius norms, we obtain:

$$\|x - \hat{x}\| \le \|M\hat{x}\|_F \le B\sqrt{|M|} \tag{15}$$

This provides a deterministic upper bound on the input mismatch term in our staleness analysis.

Combining input and parameter mismatch: For a fixed input $\hat{x}$, we take expectation over the masking randomness only. By the triangle inequality:

$$\mathbb{E}_M[\|f(x) - g(\hat{x})\| \mid \hat{x}] \leq L_f^{(t)} \mathbb{E}_M[\|x - \hat{x}\|] + \|f(\hat{x}) - g(\hat{x})\|. \tag{16}$$

Note that the second term has no expectation since $\hat{x}$ is fixed.

For the *input mismatch term*, since $L_f^{(t)}$ is deterministic at epoch t, applying Jensen's inequality to Equation (15):

$$L_f^{(t)} \mathbb{E}_M[\|x - \hat{x}\|] \leq L_f^{(t)} B \sqrt{\mathbb{E}[|M|]} = L_f^{(t)} B \sqrt{pN}, \tag{17}$$

where $\mathbb{E}[|M|] = pN$ holds for both Bernoulli and block masking.

For the *EMA mismatch term*, since $\hat{x}$ is fixed, Theorem 4.2 gives deterministically:

$$\|f(\hat{x}) - g(\hat{x})\| \leq \left\| \beta \Delta^{(t)} - (1-\beta) \sum_{\ell=1}^{t-1} \beta^{t-\ell} \Delta^{(\ell)} \right\| \|\hat{x}\|. \tag{18}$$

Combining yields $\mathbb{E}_M[\|f(x) - g(\hat{x})\| \mid \hat{x}] \leq \gamma(\hat{x})$ with

$$\gamma(\hat{x}) = L_f^{(t)} B \sqrt{pN} + \left\| \beta \Delta^{(t)} - (1-\beta) \sum_{\ell=1}^{t-1} \beta^{t-\ell} \Delta^{(\ell)} \right\| \|\hat{x}\|. \tag{19}$$

□

Remark C.1 (Bernoulli masking). When each token is masked independently with probability p (as in TS-JEPA), we have $m_i \sim \text{Bernoulli}(p)$ independently. The expected number of masked tokens is:

$$\mathbb{E}[|M|] = pN, \qquad \text{Var}(|M|) = Np(1-p)$$

By Hoeffding's inequality, with probability at least $1 - \delta$:

$$|M| \leq pN + \sqrt{\frac{N}{2} \log \frac{2}{\delta}} \tag{20}$$

Therefore, with high probability:

$$\epsilon = \|x - \hat{x}\| \leq B \sqrt{pN + \mathcal{O}_p(\sqrt{N})}$$

Remark C.2 (Block masking). When masking uses contiguous blocks with expected coverage p (as in I-JEPA), the mean bound holds:

$$\mathbb{E}[\|x - \hat{x}\|] \leq B \sqrt{pN}$$

However, block masking introduces positive correlations among masked indices, selecting spatially or temporally correlated tokens. This creates higher variance in $\|M\hat{x}\|$ compared to independent Bernoulli masking, with values more frequently approaching the extremes of the deterministic bound $B\sqrt{|M|}$.

Remark on the masking operator (deferred from Section 4.3). In practice, Transformer-based architectures such as I-JEPA operate only on visible tokens by extracting them into a shorter sequence, rather than explicitly forming the zero-padded representation $(I-M)\hat{x}$ used in our analysis. However, masking operators are non-expansive, i.e.:

$$\|(I - M)v\| \leq \|v\| \quad \text{for any } v,$$

which ensures that our bound on $\|M\hat{x}\|$ provides a valid upper bound on the discrepancy between masked and full representations, even if it is not tight in this setting.

D Additional Results

In addition to the final downstream results reported in Table 1, where we compared the proposed noise injection strategy against the original TS-JEPA, we further analyze how noise injection influences representation quality throughout pre-training. To this end, we periodically evaluate the pre-trained encoder at regular checkpoints, logging downstream accuracy every 100 training steps.

Figure 5 presents the evolution of downstream performance across checkpoints. We observe that, at all stages of pre-training, models trained with noise injection consistently outperform their EMA-regularized counterparts. This behavior indicates that noise injection not only improves final performance but also accelerates the transition into the regime where meaningful representations are learned. These results provide additional empirical evidence that controlled noise injection facilitates and stabilizes the staleness phase, leading to faster and more effective representation learning.

A similar trend is also observed in the image domain, as illustrated on CIFAR-10 in the lower part of Figure 1, where noise injection enables the encoder to enter the representation-learning regime earlier during training.

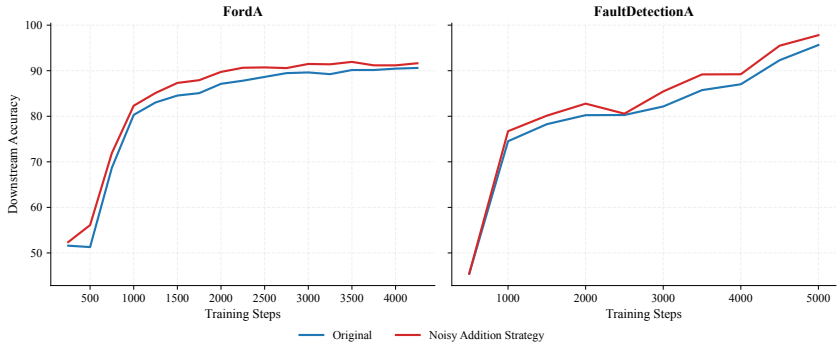


Figure 5: Comparison of the original TS-JEPA with the noise injection variant on the time series datasets FordA and FaultDetectionA for different checkpoints.

E Pre-training Loss on ImageNet under Noise Injection

Figure 6 shows the JEPA pre-training loss on ImageNet-1K and ImageNet-100 for the original I-JEPA and the noise-injection variant. Consistent with the symmetry-breaking account of Section 5, the perturbation alters the early loss landscape while the long-run dynamics coincide.

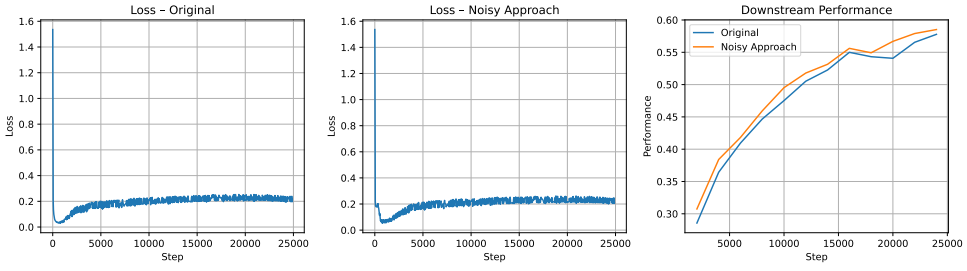


Figure 6: JEPA pre-training loss for the original I-JEPA and the noise-injection strategy on ImageNet-1K (left) and ImageNet-100 (right).

F Implementation Details

We note that the code implementation to reproduce our experimental results is provided in the supplementary materials and shall be made public upon acceptance.

F.1 DATASETS

For our empirical validation, we focused on two main modalities, namely Images and Time Series. For the first modality, in terms of pre-training, we used Imagenet-1K and ImageNet-100, which are two subsets of the larger ImageNet dataset (Russakovsky et al., 2015). For the evaluation, we focused additionally on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). For these specific results, we used an average pooling during linear probing phase, other pooling could also be used and investigated (ENNADIR et al., 2025).

For the time series related tasks, we focused on the classification task where we consider the FordA, FordB (Dau et al., 2019), FaultDetectionA and FaultDetectionB (Lessmeier et al., 2016), all of which comprise outputs from various sensors. Additional statistics about these datasets are provided in Table 3.

Table 3: Statistics of the classification datasets used in our experiments.

DATASET	#TRAINING POINT	#TEST POINTS	#LENGTH	#CLASSES
FORDA	3601	1320	500	2
FORDB	3636	810	500	2
FAULTDETECTIONA	10912	2728	5120	3
FAULTDETECTIONB	10912	2728	5120	3

F.2 ARCHITECTURE AND PARAMETERS DETAILS

All computer vision related experiments were based on using a Transformer backbone (ViT-base (Dosovitskiy et al., 2021)). We used the same predictor and masking strategy as the one used in the original I-JEPA.

Optimization For our experiments, we followed the same setting as the original I-JEPA here as well. Specifically, all the experiments were optimized using AdamW (Loshchilov & Hutter, 2019), to produce both the encoder and predictor weights. For the ImageNet-100, we used a batch size of 512, while for the ImageNet-1K, we used a batch size of 2048, and the learning rate is linearly increased from 10^{-4} to 10^{-3} during the first 15 epochs of pretraining, and decayed to 10^{-6} following a cosine schedule thereafter. All experiments were conducted on NVIDIA A100 and H100 GPUs.

Time-Series Experiments. For this specific part, we have followed again a similar setup as the original TS-JEPA paper, where the encoder and decoder are a transformer with 2 attention heads and an embedding dimension of 128. For all the considered datasets and experiments, we segment each time series into 10 patches and employ a batch size of 32. For TS-JEPA, we apply a masking ratio of 70%. Similar to the image-based tasks, we trained all the models using the AdamW optimizer (Loshchilov & Hutter, 2019). We used learning rates similar to those used in the original work, namely within the range 10^{-4}.