Stochastic positional embeddings improve masked image modeling

Amir Bar [1 2 3]   **Florian Bordes** [3]   **Assaf Shocher** [2]   **Mahmoud Assran** [3]   **Pascal Vincent** [3]   **Nicolas Ballas** [3]
Trevor Darrell [2]   **Amir Globerson** [1 4]   **Yann LeCun** [3 5]

Abstract

Masked Image Modeling (MIM) is a promising self-supervised learning approach that enables learning from unlabeled images. Despite its recent success, learning good representations through MIM remains challenging because it requires predicting the right semantic content in accurate locations. For example, given an incomplete picture of a dog, we can guess that there is a tail, but we cannot determine its exact location. In this work, we propose to incorporate location uncertainty into MIM by using stochastic positional embeddings (StoP). Specifically, we condition the model on stochastic masked token positions drawn from a Gaussian distribution. StoP reduces overfitting to location features and guides the model toward learning features that are more robust to location uncertainties. Quantitatively, StoP improves downstream MIM performance on a variety of downstream tasks, including $+1.7\%$ on ImageNet linear probing using ViT-B, and $+2.5\%$ for ViT-H using 1% of the data.[1]

1. Introduction

Masked Image Modeling (MIM) enables learning from unlabeled images by reconstructing masked parts of the image given the rest of the image as context. In recently years, new MIM methods have emerged (Xie et al., 2021; Bao et al., 2021; He et al., 2021; Assran et al., 2023). Masked Auto-Encoders (MAE) (He et al., 2021) are trained to minimize a reconstruction error in pixel space, and I-JEPA (Assran et al., 2023) reconstructs image features. MIM is appealing compared to invariance-based self-supervised learning methods like DINO (Caron et al., 2021) and iBOT (Zhou et al., 2021) as MIM do not suffer from the same limita-

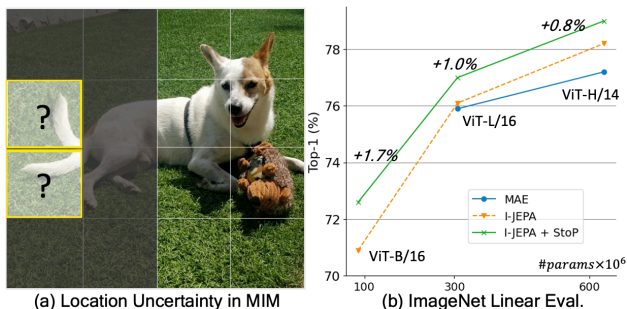(a) Location Uncertainty in MIM   (b) ImageNet Linear Eval.

Figure 1: Given a partial image of a dog, can you precisely determine the location of its tail? Existing Masked Image Modeling (MIM) models like MAE (He et al., 2021) and I-JEPA (Assran et al., 2023) predict tokens deterministically and do not model location uncertainties (a), we propose to predict the target (masked tokens) in stochastic positions (StoP) which prevents overfitting to locations features. StoP leads to improved MIM performance on downstream tasks, including linear probing on ImageNet (b).

tions, namely, it does not require heavy use of hand-crafted augmentations (Xiao et al.; He et al., 2021), mini-batch statistics, or a uniform cluster prior (Assran et al., 2022).

Despite the recent success of MIM, we argue that learning good representations using MIM remains challenging due to location uncertainties because it requires predicting the right semantic content in accurate locations. For example, given an incomplete picture of a dog (see Figure 1a), we might guess there's a tail, but we can't be sure exactly where it is, as it could realistically be in several different places. Without explicitly modeling this location uncertainty, existing MIM models like MAE and I-JEPA might overfit on semantic content in arbitrary locations (e.g, the tail location).

In this work, we propose to address location uncertainty in MIM by turning existing MIM models into stochastic ones. Instead of training the model to make predictions in exact locations, we use Stochastic Positional embeddings (StoP) to introduce noise to the masked token's positions, implicitly forcing the model to make stochastic predictions. StoP guides the model towards learning features that are more resilient to location uncertainties, such as the fact that a tail exists in a general area rather than a specific point, which improves downstream performance (Figure 1b).

Specifically, we model the position of every masked token

---

[1]Tel Aviv University [2]UC Berkeley [3]Meta AI (FAIR) [4]Now also at Google Research [5]New York University. Correspondence to: Amir Bar <amir.bar@cs.tau.ac.il>.

[1]See https://github.com/amirbar/StoP for code.

as a random variable with a Gaussian distribution where its mean is the position of the patch, and the covariance matrix is learned. We find it crucial to design StoP carefully so that the model does not collapse back to deterministic positional embeddings by scaling down the covariance matrix weights to overcome the noise.

To prevent collapse, we propose to tie between the scales of the noise and input context. With this constraint, scaling down the noise also scales down the input context, which makes the reconstruction task too hard to achieve. On the other hand, increasing the scale of the noise leads to very stochastic masked token positions, which makes the reconstruction task difficult as well. We provide a theoretical proof, showing that our solution indeed prevents collapse.

Our contributions are as follows. First, we propose the idea of Stochastic Positional embeddings (StoP) and apply it to MIM to address the location uncertainty in MIM, namely that the location of semantic features is stochastic. Second, we demonstrate that adding StoP to I-JEPA, a recent MIM approach, leads to improved performance on a variety of downstream tasks, highlighting its effectiveness. Lastly, implementing StoP for MIM requires only three extra lines of code, without adding any runtime or memory overhead.

2. Preliminaries - Masked Image Modeling

The idea in MIM is to train a model to reconstruct masked parts in an image given the rest of the image as context. In this process, a neural network f_θ learns the context representations, and a network g_ϕ is used to reconstruct the masked regions. In this section we describe the MIM algorithm, then discuss how to apply StoP to MIM in Section 3.

Patchification. Given an image, the first stage is to tokenize the image. For the case of Vision Transformers (Dosovitskiy et al., 2020), an input image $I_x \in \mathbb{R}^{H \times W \times 3}$ is first patchified into a sequence of non-overlapping image patches $\hat{p} = (\hat{p}_1, ..., \hat{p}_k)$ where $\hat{p}_i \in \mathbb{R}^{H' \times W' \times 3}$ and $K = \frac{HW}{H'W'}$ is the number of patches. Then, each patch $\hat{p}_i$ is projected to $\mathbb{R}^{d_e}$ through a linear fully connected layer and its corresponding positional embedding features are added to it, resulting in the patchified set $p = \{p_1, ...p_K\}$.

Masking. Let $x = \{p_i | i \in B_x\}$ be the set of context patches where B_x denotes the set of context indices (i.e., the visible tokens in Figure 2). We denote by B_y the indices of the target patches y. The context and target patches are chosen via random masking as in He et al. (2021) or by sampling target continuous blocks as in Assran et al. (2023).

Context encoding. The context tokens are processed via an encoder model f_θ to obtain deep representations: $s_x = f_\theta(x)$, where $s_{x_i} \in \mathbb{R}^{d_e}$ is the ith context token representation. Each token s_{x_i} is then projected from the output

dimension of the encoder d_e to the input dimension of the predictor d_p via a matrix $B \in \mathbb{R}^{d_p \times d_e}$, and it is enriched with deterministic positional embedding $\psi_i \in \mathbb{R}^{d_p}$:

$$c_i = \psi_i + Bs_{x_i} \tag{1}$$

Masked tokens. We define the set of masked tokens, where every masked token m_j for $j \in B_y$ is composed of the positional embeddings of the jth patch ψ_j and a bias term $\tilde{m}$ that is shared across all masked tokens, namely:

$$m_j = \psi_j + \tilde{m} \tag{2}$$

Prediction and loss. Finally, the predictor function g_ϕ is applied to predict the target features $\hat{s}_y = g_\phi(c, m)$. To supervise the prediction, the ground truth $s_y = \{s_{y_i}\}_{i \in B_y}$ is obtained either by using the raw RGB pixels or via a latent representation of the pixels. The loss $\frac{1}{|B_y|} \sum_{i \in B_y} L(s_{y_i}, \hat{s}_{y_i})$ is then applied to minimize the prediction error.

3. Masked Image Modeling with StoP

This section presents the StoP formulation, and how to utilize it in MIM while avoiding collapsing back to deterministic positional embeddings. A high-level schematic view of the model is included in Figure 2, and a pseudo-code implementation is included in Algorithm 1.

Stochastic Positional Embeddings (StoP). Instead of training the model to make predictions in exact locations, we propose to use stochastic positional embeddings which implicitly force the model to make stochastic predictions. This is meant to teach the model that locations cannot be predicted precisely, resulting in improved robustness.

Formulating StoP requires defining the distribution of the stochastic positions, parameterizing it appropriately, and implementing measures to prevent the model from scaling down the noise to the point where it becomes negligible.

Given a position j, we denote by $\hat{\psi}_j$ the random variable providing the position embedding. We assume that $\hat{\psi}_j$ is distributed as Gaussian whose mean is the fixed embedding ψ_j, and whose covariance matrix is $\Sigma \in \mathbb{R}^{d_p \times d_p}$:

$$\hat{\psi}_j \sim N(\psi_j, \Sigma) \tag{3}$$

Naturally, we want to learn an optimal Σ. To parameterize Σ, we use a general formulation of a low-rank covariance matrix:

$$\Sigma = \sigma A A^T \tag{4}$$

Where $A \in \mathbb{R}^{d_p \times d_e}$ is a learned matrix and $\sigma \in \mathbb{R}^+$ is a positive scalar hyperparameter used to control the Noise to Signal Ratio (NSR).[2] By learning the matrix A, this for-

---

[2]At this point, it may seem unnecessary to have an additional σ parameter. However, later we will tie A to other model parameters, and thus σ will not be redundant and determine the scale of the noise.
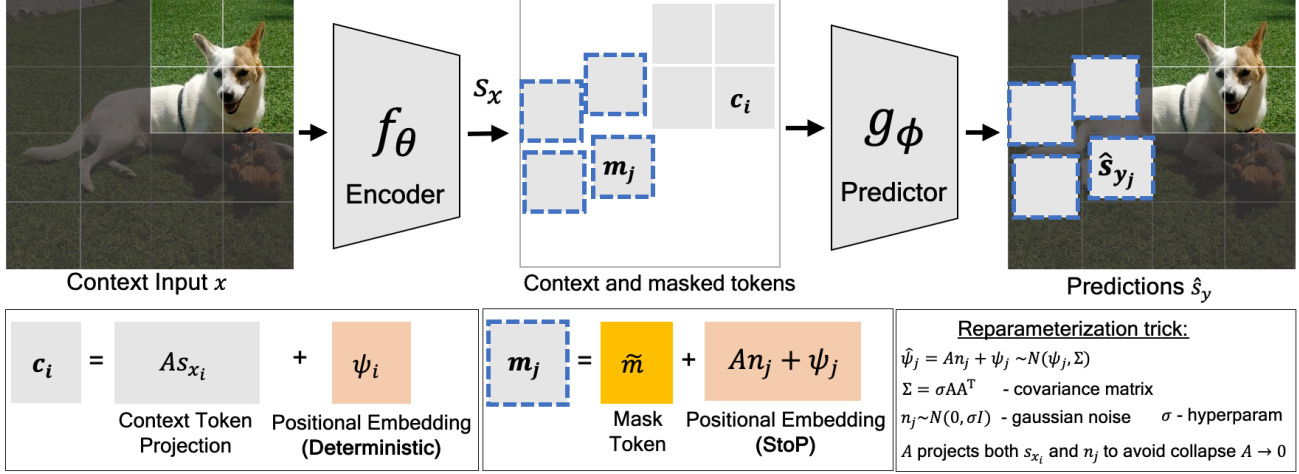
Figure 2: **Masked image modeling using stochastic positional embeddings (StoP).** g_ϕ predicts target tokens given masked tokens with stochastic positions m_j and context tokens c_i obtained via f_θ. StoP is applied to masked tokens only, leading to features that are more robust to location uncertainties.

mulation allows assigning different noise levels to different location components (e.g., high and low resolution), as well as capturing correlations between location features.

Using this formulation is challenging for two reasons. First, the sampling process of $\hat{\psi}$ is non-differential w.r.t A, and therefore we cannot derive gradients to directly optimize it with SGD. Second, learning might result in the optimization process setting the values of Σ to zero, leading to no randomness. Next, we move to solve these issues.

Reparametrization Trick. Since $\hat{\psi}_j$ is sampled from a parameterized distribution, it is non-differentiable in A. However, a standard trick in these cases is to reparameterize the distribution so that the sampling is from a fixed distribution that does not depend on A (e.g., see Kingma & Welling (2013)). Specifically, we generate samples from $\hat{\psi}_j$ by first sampling a vector $n_j \in \mathbb{R}^{d_e}$ from a standard Gaussian distribution: $n_j \sim N(0, \sigma I)$. Then, $\hat{\psi}_j$ is set to:

$$\hat{\psi}_j = An_j + \psi_j \qquad (5)$$

The resulting distribution of $\hat{\psi}_j$ is equal to that in Equation 3, however, we can now differentiate directly through A.

Collapse to deterministic positions (A=0). Intuitively, adding noise to an objective hurts the training loss, and thus if A appears only in (5), training should set it to zero. We indeed observe this empirically, suggesting that A cannot only appear in a single place in the model. In what follows we propose an approach to overcoming this issue.

Avoiding collapse by weight tying A=B. To avoid the collapse to deterministic positions, we propose to tie the weights of A and B (originally defined in Eq. 1), such that the same matrix A projects both the context tokens s_{x_i} and

Algorithm 1 MIM w/ StoP pseudo-code. requires only a minor implementation change, highlighted in light gray.

1: **Input:** num iterations K, image dist S, hyperparam σ, positional embeddings ψ
2: **Params**: $A, \tilde{m}$, encoder f_θ, predictor g_ϕ
3: **for** $itr = 1, 2, ..., K$ **do**
4: $\quad I_x \sim S$
5: $\quad p \leftarrow \text{patchify}(I_x)$
6: $\quad (x, B_x), (y, B_y) \leftarrow \text{mask}(p)$
7: $\quad s_x \leftarrow f_\theta(x)$
8: $\quad$ # apply StoP on a sequence of tokens
9: $\quad n_j \sim \mathcal{N}(0, \sigma I)$
10: $\quad$ # ψ_{B_x}, ψ_{B_y} - masked/context positional embeddings
11: $\quad m = An + \psi_{B_y} + \tilde{m}$
12: $\quad c = As_x + \psi_{B_x}$
13: $\quad$ # predict targets
14: $\quad \hat{s}_y \leftarrow g_\phi(c, m)$
15: $\quad s_y \leftarrow \text{get\_target}(y)$
16: $\quad \text{loss} \leftarrow L(\hat{s}_y, s_y)$
17: $\quad \text{sgd\_step}(\text{loss}; \{\theta, \phi, A, \tilde{m}\})$
18: **end for**

the noise tokens n_j:

$$c_i = As_{x_i} + \psi_i \quad m_j = An_j + \psi_j + \tilde{m} \qquad (6)$$

This tying means that the scale of the noise and the input are both determined by A, and thus the noise cannot be set to zero, without affecting other parts of the model. This can be understood by considering two extreme cases:

- If $A = 0$, there is complete certainty about the positional embeddings but all context is lost ($As_{x_i} = 0$).

- If A has large magnitude, the context information is preserved but the noise is amplified and camouflages masked tokens positional embeddings ($An_j \gg \psi_j$).

This dual role of A forces the model to trade-off between the positions of the masked tokens and the context tokens.[3]

In the following proposition, we formally show that if the weights A and B are tied then A cannot collapse. More specifically, $A = 0$ occurs only if in the original deterministic setting B goes to zero and doesn't utilize the context anyway. Formally, consider a regression task where F predicts some target y_j given a stochastic position $An_j + \psi_j + \tilde{m}$ where $n_j \sim N(0, \sigma I)$ and projected context token Bx_i. Denote J_{tied}, J_{det} the loss functions when tying the weights A and B, and when using deterministic positional embeddings respectively:

$$J_{tied}(A) = \sum_{i,j} \mathbb{E}_{n_j}[(F(An_j + \psi_j + \tilde{m}, Ax_i) - y_j)^2]$$

$$J_{det}(B) = \sum_{i,j}[(F(\psi_j + \tilde{m}, Bx_i) - y_j)^2]$$

Proposition 3.1. *If the weights of A and B are tied (namely $A = B$) then $\frac{dJ_{tied}}{dA}\big|_{A=0} = 0$ iff $\frac{dJ_{det}}{dB}\big|_{B=0} = 0$*

Proof is included in Appendix A.

Optimal Predictor. Our approach relies on using stochastic positional embeddings. Here we provide further analysis, showing that the optimal predictor performs spatial smoothing. Consider a random variable X (corresponding to the context in our case. For simplicity assume X is just the positional embedding of the context) that is used to predict a variable Y (corresponding to the target in our case). But now instead of predicting from X, we use a noise variable Z that is independent of both X, Y, and provide the predictor with only the noisy result $R = g(X, Z)$. Here g is some mixing function (in our case $g(x, z) = x + z$). We next derive the optimal predictor $f(R)$ in this case. Formally we want to minimize:

$$E_{R,Y}[(f(R) - Y)^2] \tag{7}$$

Proposition 3.2. *If Z is a Gaussian with zero mean and unit variance, the optimal predictor that minimizes Equation 7 is:*

$$f(r) = \int_x E[Y|X = x]\frac{1}{\sqrt{2\pi}}e^{-0.5(x-r)^2}dx$$

Thus, the optimal predictor amounts to a convolution of the clean expected values with a Gaussian. See Appendix B for the proof.

---

[3]Note that an implicit assumption here is that ψ and s_x have fixed magnitude. This is true for sine-cosine features and for s_x which are layer normalized by the transformer last layer.

4. Experiments and Results

Next, we turn to discuss the main experiments presented in the paper. In Section 4.1, we describe the application of StoP to various downstream tasks including image recognition, dense prediction, and low-level vision tasks. In Section 4.2 we discuss the ablation study and design choices. The full implementation details are included in Appendix C.

4.1. Downstream Tasks

We conducted pre-training of StoP on top of I-JEPA, which is a state-of-the-art MIM model. We train on IN-1k for a period of 600 epochs using ViT-B/16 and ViT-L/16 architectures for the encoder and predictor or for 300 epochs when using ViT-H/14. Subsequently, we proceeded to evaluate the model's performance on a variety of downstream tasks. Additional results and comparison to invariance-based approaches are included Appendix C.2.

Image recognition. For image classification, we perform a linear probing evaluation of StoP on multiple datasets, including ImageNet (IN-1k) (Russakovsky et al., 2015), Places 205 (Zhou et al., 2014a), iNaturalist 2018 (Van Horn et al., 2018), and CIFAR 100 (Krizhevsky, 2009). These datasets vary in their size, their purpose, and the geographical environments from which the images were captured. For example, IN-1k contains over 1.2 million images compared to CIFAR-100 which contains only $60,000$ images, and while IN-1k is focused on object recognition, iNaturalist and Places are focused on scene and species recognition.

In Table 1, we present the linear probing image classification results conducted on IN-1k under different linear evaluation protocols using different amounts of data, and by aggregating features from different layers. E.g, "100%, last 4 layers" applies linear probing on the entire IN-1k data and the representation of each image is comprised of a concatenation of four feature vectors, each one summarizes information from its corresponding layer via average pooling. In Table 2 we compare linear probing results of common MIM methods on IN-1k, reporting past published performance. In Table 2 all perform linear probing over the output from the last layer.

StoP improves the baseline performance using all architectures examined. For example, $+2.5\%$ linear probing performance gains with ViT-H using 1% of the labeled data and 1.6% when using features from the last 4 layers using ViT-B on the full IN-1k data. Furthermore, using StoP leads to improvements in downstream linear probing tasks (see Table 4). For example, StoP leads to 3.3% improvement on iNAT using ViT-H and 1.3% on counting. This confirms that the learned representations lead to improvements in a large variety of image recognition tasks. On full finetuning using 1% of the labeled data, we observe similar performance improvements (see Table 5), e.g, $+2.3\%$ improvements on

4

Arch	Method	1%, last layer	100%, last layer	100%, last 4 layers
ViT-B/16	I-JEPA	57.1	70.9	72.9
	+StoP	60.3 (+3.2%)	72.6 (+1.7%)	74.5 (+1.6%)
ViT-L/16	I-JEPA	64.2	76.1	77.5
	+StoP	65.1 (+0.9%)	77.1 (+1.0%)	78.5 (+1.0%)
ViT-H/14	I-JEPA	62.9	78.2	79.3
	+StoP	65.4 (+2.5%)	79.0 (+0.8%)	79.6 (+0.3%)

Table 1: **StoP compared to deterministic sinusoidal positional embeddings on IN-1k**. StoP leads to consistent linear probing improvement in all settings. When applying linear probing on a trained ViT-H model with StoP, using only 1% of the labeled data and using averaged pooled features from the last layer, StoP results in an +2.5% improvement. The baseline I-JEPA uses sinusoidal positional embeddings.

Method	Arch.	Epochs	Top-1
data2vec	ViT-L/16	1600	77.3
MAE	ViT-B/16	1600	68.0
	ViT-L/16	1600	75.8
	ViT-H/14	1600	76.6
I-JEPA	ViT-B/16	600	70.9
	ViT-L/16	600	76.1
	ViT-H/14	300	78.2
+StoP (ours)	ViT-B/16	600	72.6
	ViT-L/16	600	77.1
	ViT-H/14	300	**79.0**

Table 2: **Linear-evaluation on IN-1k**. Replacing sinusoidal positional embeddings with StoP in I-JEPA significantly improves linear probing results.

Method	Arch.	J-Mean	F-Mean	J&F Mean
MAE	ViT-B/16	49.4	52.6	50.9
	ViT-L/16	52.5	54.3	53.4
	ViT-H/14	54.0	57.0	55.5
I-JEPA	ViT-B/16	56.1	56.2	56.1
	ViT-L/16	56.1	55.7	55.9
	ViT-H/14	58.5	60.9	59.7
+StoP	ViT-B/16	56.6	57.3	57.0
	ViT-L/16	58.1	58.7	58.4
	ViT-H/14	**58.9**	**61.2**	**60.1**

Table 3: **Video objects semi-supervised segmentation.** MIM with StoP learns features with a finer level of granularity. Results are reported on DAVIS 2017 dataset.
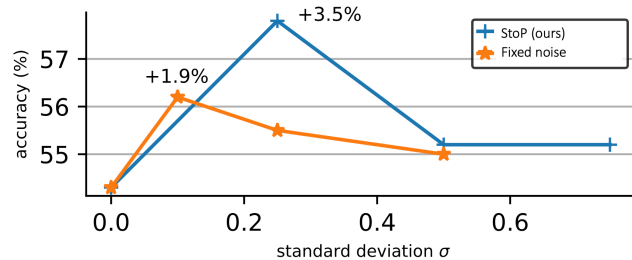
Top-1 accuracy using ViT-L model. We provide the full finetuning results in Table 17, Appendix C.2.

Counting and depth ordering. We assess the downstream performance on tasks that require fine-grained objects representations like counting and depth ordering using the CLEVR (Johnson et al., 2017) dataset. Table 4 provides evidence that using StoP significantly improve counting (+1.3%) and slightly improve depth ordering (+0.1%).

Dense prediction. To evaluate how well StoP performs on dense prediction tasks, e.g, tasks that require fine-grained spatial representations, we utilized the learned models for semi-supervised video object segmentation on the DAVIS 2017 (Pont-Tuset et al., 2017) dataset. We follow previous works (e.g Jabri et al. (2020); Caron et al. (2021)) and use the pretrained model to extract frames features and use patch-level affinities between frames to track the first segmentation mask. We include video semi-supervised video-object segmentation by tracking results in Table 3. We find that StoP significantly improves over I-JEPA with deterministic sinusoidal location features. For example, we observe an improvement of +2.5% in $J\&F$ using ViT-L.

Figure 3: **Learned vs. predefined stochastic positions.** Using the learned covariance matrix as in StoP, e.g, $\Sigma = \sigma A A^T$ leads to +3.5% improvement compared to smaller gains with a fixed covariance matrix $\Sigma = \sigma I$. Accuracy is reported based on linear probing evaluation using 1% of the data from IN-1k.

4.2. Ablation Study

Our primary focus is to evaluate the effectiveness of StoP. To demonstrate this, we assess various design options using ViT-B architecture for the encoder and predictor. We pre-train for 300 epochs on IN-1k based on the I-JEPA (Assran et al., 2023) MIM model. We then assessed the linear probing performance on IN-1k using only 1% of the labels.

StoP compared to deterministic positional embeddings. The most common choices for positional embeddings for Vision Transformers are sine-cosine location features (also

Method	Arch.	CIFAR100	Places205	iNat18	CLEVR/Count	CLEVR/Dist
data2vec	ViT-L/16	81.6	54.6	28.1	85.3	71.3
MAE	ViT-B/16	68.1	49.2	26.8	86.6	70.8
	ViT-L/16	77.4	54.4	33.0	**92.1**	73.0
	ViT-H/14	77.3	55.0	32.9	90.5	72.4
I-JEPA	ViT-B/16	69.2	53.4	43.4	82.2	70.7
	ViT-L/16	83.6	56.5	48.4	85.6	71.2
	ViT-H/14	87.5	58.4	47.6	86.7	72.4
+StoP	ViT-B/16	81.2	54.3	44.7	83.7	71.3
	ViT-L/16	84.7	57.2	49.2	85.7	70.2
	ViT-H/14	**87.7**	**58.4**	**50.9**	88.0	**72.5**

Table 4: **Linear-probe transfer for various downstream tasks**. Linear-evaluation on downstream image classification, object counting, and depth ordering tasks. Using StoP instead of sinusoidal deterministic positions leads to improvements on all tasks. E.g., +3.3% on iNAT18 and +1.3% on Counting.

Method	Epochs	Top-1
Sine Cosine	600	69.4
StoP (ours)	600	**71.7**

Table 5: **Finetuning results over IN-1k with 1% labels.** StoP significantly improves finetuning performance compared to using sine-cosine positional embeddings. Using ViT-L/16 architecture.

Method	Top-1
Sine Cosine	54.3
Learned Pos. Embedding	54.4
Stochastic Positions (StoP)	**57.8**

Table 6: **Different positional embeddings**. Linear probing on IN-1K using only 1% of the labels. Stochastic Positions (StoP) outperforms other common deterministic variants by 3.3%.

used in MAE, I-JEPA) and learned positional embedding. We evaluate the MIM downstream performance using each of these options and using StoP (see Table 6). The results indicate that using StoP improves the performance by +3.2% compared to sinusoidal and learned positional embeddings.

Learned vs. predefined covariance matrix. To confirm that learning the covariance matrix $\Sigma = \sigma A A^T$ (and specifically A) is beneficial compared to using a predefined covariance matrix, we compare to stochastic positional embeddings with a predefined covariance matrix $\Sigma = \sigma I$, without any learning. We compare both options using different σ hyperparameter values. Figure 3 indicates that it is advantageous to learn Σ rather than use fixed parameters. Our findings show that setting the hyperparameter value to $\sigma = 0.25$ leads to an improvement of 3.5% points compared to deterministic positional embeddings ($\sigma = 0$).

Application of StoP to different tokens. We apply StoP to context and/or masked tokens. The results in Table 7 confirm our design choice, showing that StoP is most beneficial when it is applied solely to masked tokens, compared to context tokens, or both masked and context tokens.
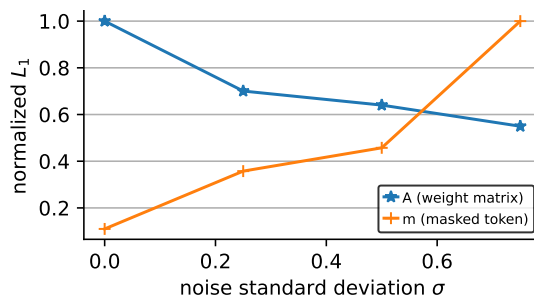

Figure 4: **Increasing σ induces regularization.** Changing the prior σ (where $\Sigma = \sigma A A^T$) induces regularization over A and increases the norm of the masked token, which preserves the masked token information in comparison to the added noise.

Method	Top-1
No Noise (Sine Cosine)	54.3
Context tokens only	55.1
Masked + context tokens	56.8
Masked tokens only	**57.8**

Table 7: **Applying noise to different tokens**. Applying learned noise to context and/or masked tokens positional embeddings (sine-cosine). Reporting linear evaluation accuracy (using 1% of IN-1k).

4.3. Analysis

To explain how StoP affects MIM, we analyze the learned model weights, visualize the stochastic positional embeddings, and visualize the predicted features.

StoP induces regularization. The matrix A is used to project both noise tokens and context embedding tokens. We hypothesize that StoP implicitly regularizes A. To test this hypothesis we train models using StoP changing only the hyperparam σ (see Figure 4). We find that increasing the value of σ leads to a decrease in the norm of A, which can be viewed as regularization. On the other hand, increasing σ leads to an increase in the norm of the masked token bias

Method	Top-1
Sine Cosine	54.3
x2 Low res (bilinear resize)	52.1
x2 Low res (max pooling)	54.1
Stochastic Positions (StoP)	**57.8**

Table 8: **Low resolution prediction**. Performance of StoP compared to models that predict features on lower scales via max pooling or bilinear resizing. Reporting linear evaluation accuracy (using 1% of IN-1k). StoP performs better than low res prediction.

$\tilde{m}$. We speculate that the masked token bias increases in scale to prevent losing its information relative to the noise.

To further analyze this phenomenon, we train additional models while applying l_1 or l_2 regularization on A while keeping the positional embeddings of masked tokens deterministic. We find that StoP leads to +2% improvement over l_1 and +2.1% over l_2 regualrization. Therefore, we conclude that StoP is superior to simple regularization.

Stochastic positional embedding visualization. To visualize how StoP affects the similarity between different positions, we plot the similarity matrix between a stochastic position embedding query and the predefined sine-cosine deterministic positions (Figure 5). With StoP, we find that query locations are more similar to a wider range of neighboring locations. Building on this observation, we train models to investigate if directly predicting lower-scale features is beneficial. We trained models to predict features in both the original scale and a downscaled version by a factor of 2, using bilinear resizing and max pooling for downscaling. However, we found that predicting lower scale features does not improve performance (see Table 8).

Prediction visualization. We include heatmap visualization to visualize the similarity of a predicted token to all other tokens within the same image (see Figure 6). For a given image, mask, and a masked patch of interest, we apply cosine similarity between the predicted patch and all other token representations within the same image, followed by a softmax. For I-JEPA with sine-cosine positional embeddings, the visualization indicates that adjacent tokens tend to share similar features, implying a correlation between the features and spatial location. In contrast, StoP produces predictions correlated with non-neighboring small areas. We speculate that using StoP leads to learning features that are more semantic and prevents overfitting to location features.

5. Related Work

Masked image modeling (MIM). There is a significant body of research exploring visual representation learning by predicting corrupted sensory inputs. Denoising autoencoders (Vincent et al., 2010), for example, use random noise as input corruption, while context encoders (Pathak et al.,
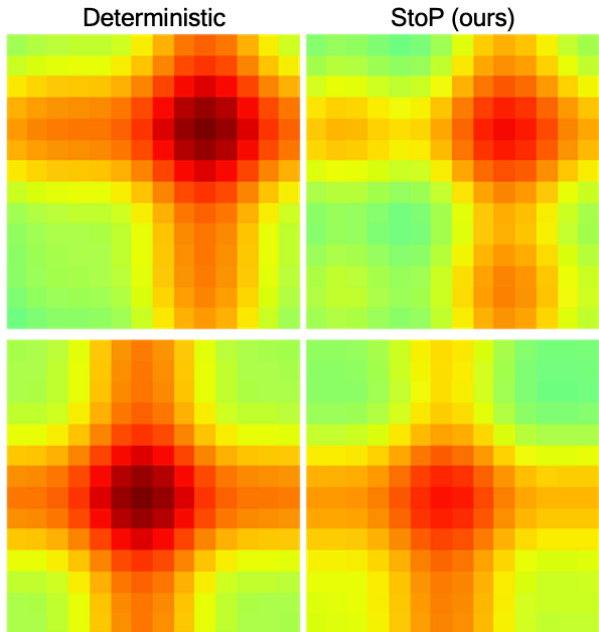

Figure 5: **Similarity matrices of deterministic and stochastic positional embedding (StoP) to a query position**. Each row represents the similarity given a different query position. StoP leads to a spatially smooth similarity matrix, thereby making it hard to distinguish the exact location of a given patch.

2016) regress an entire image region based on its surrounding. The idea behind masked image modeling (He et al., 2021; Xie et al., 2021; Bao et al., 2021) has emerged as a way to address image denoising. In this approach, a Vision Transformer (Dosovitskiy et al., 2020) is used to reconstruct missing input patches. The Masked Autoencoders (MAE) architecture (He et al., 2021), for example, efficiently reconstructs missing patches in pixel space and achieves strong performance on large labeled datasets. Other approaches, such as BEiT (Bao et al., 2021), predict a latent code obtained using a pretrained tokenizer. However, pixel-level pre-training has been shown to outperform BEiT in fine-tuning. SimMiM (Xie et al., 2021) explores simple reconstruction targets like color clusters but shows no significant advantages over pixel space reconstruction. Recently, Image-JEPA (I-JEPA) (Assran et al., 2023; LeCun, 2022) was proposed as a non-generative approach for self-supervised learning of semantic image representations. I-JEPA predicts the representations of various target blocks in an image from a single context block to guide it toward producing semantic representations. Our approach builds on this line of work and we propose to deal with location uncertainty using stochastic positional embeddings which was not explored before.

Positional Embeddings in Transformers. One of the core components of the Transformer architecture (Vaswani et al., 2017) is the Self-Attention block, which is a permutation in-
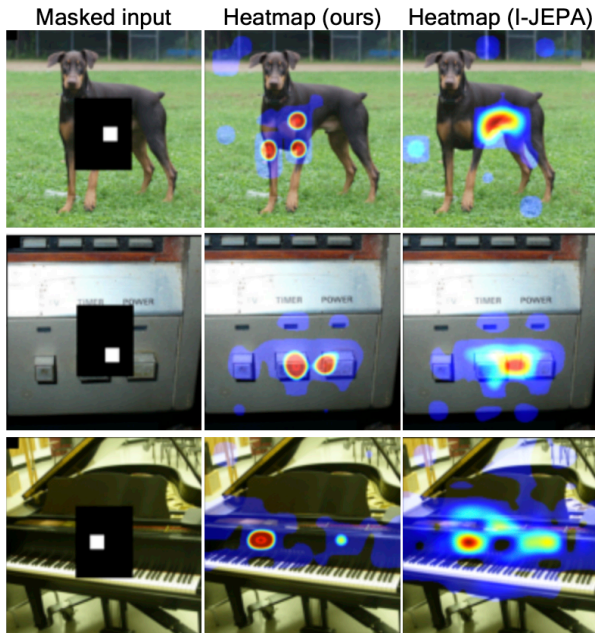
Figure 6: **Feature visualization**. We plot the similarity between the predicted features of a given patch (marked in white within the masked black area) and other features in the same image. Using StoP produces features that are less location based compared to I-JEPA baseline that have strong correlation with the target location.

variant function, e.g, changing the order of the input tokens does not change the function output. Consequently, it is necessary to feed input tokens together with their positional embedding to describe their location. Absolute positional embeddings like fixed 2D sinusoidal features (Bello et al., 2019) or learned location features are the prevalent type of positional embeddings for the Vision Transformer (Dosovitskiy et al., 2020). Relative positional embeddings have recently gained popularity in NLP due to their ability to address the gap between the training and testing sequence length (Su et al., 2021; Chu et al., 2021; Press et al., 2021). For example, (Press et al., 2021) proposed ALiBi to bias self-attention to assign higher confidence to neighboring locations, and SPE (Liutkus et al., 2021) proposed a stochastic approximation for relative positional embedding in linear transformers. Differently, we propose StoP to tackle location uncertainties in MIM, and it can be easily applied on top of any existing deterministic variant.

Invariance-based methods. These methods incorporate a loss that encourages similarity between augmented views of the the same image while avoiding a trivial solution. For example, contrastive learning prevents collapse by introducing negative examples (Hadsell et al., 2006; Dosovitskiy et al., 2014; Chen et al., 2020a; He et al., 2019; Chen et al., 2020b; Dwibedi et al., 2021). This can be achieved using a memory bank of previous instances (Wu et al., 2018; Oord et al., 2018; Tian et al., 2019; Misra & van der Maaten, 2020). However, there are also non-contrastive solutions that have

been proposed. Of particular interest, a momentum encoder has been shown to prevent collapse even without negative pairs (Grill et al., 2020; Caron et al., 2021; Salakhutdinov & Hinton, 2007). Other methods include stopping the gradient to one branch (Chen & He, 2021) or applying regularization using batch statistics (Zbontar et al., 2021; Bardes et al., 2021; 2022; Ermolov et al., 2020; Hua et al., 2021). MoCo v3 (Chen et al., 2021), then DINO (Caron et al., 2021) extended these approaches for Vision Transformer, and iBOT (Zhou et al., 2021) proposed to add a MIM loss to DINO. These approaches perform extremely well on ImageNet linear-probing, yet they rely on batch statistics, struggle under non-uniform distributions (Assran et al., 2022), and require hand-crafted image augmentations (Xiao et al.). Our approach is based on MIM that requires less assumptions on batch statistics or handcrafted invariances.

6. Limitations

We applied StoP to I-JEPA which performs image reconstruction in the feature space. However, our attempts to apply StoP to MIM that use pixel based reconstruction, mainly MAE, were not successful. We speculate that adding StoP to MAE might make pixel reconstruction too difficult to achieve. Additionally, StoP tackles location uncertainty but not appearance uncertainty, which we believe is implicitly modeled by reconstructing tokens in feature space. Also, when modeling stochastic positions it may might be possible to condition the noise on the input image, namely the context tokens. We leave this extension for future work. Lastly, while combining StoP with MIM shows significant improvements, invariance-based approaches still perform slightly better (e.g, iBOT, DINO) than MIM approaches.

7. Conclusion

In this work, we proposed to use stochastic positional embedding (StoP) to tackle location uncertainty in MIM. By conditioning on stochastic masked token positions, our model learns features that are more robust to location uncertainty. The effectiveness of this approach is demonstrated on various datasets and downstream tasks, outperforming existing MIM methods and highlighting its potential for self-supervised learning. Based on our experiments and visualizations, modeling location uncertainties with StoP reduces overfitting to location features.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

References

Assran, M., Balestriero, R., Duval, Q., Bordes, F., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., and Ballas, N. The hidden uniform cluster prior in self-supervised learning. *arXiv preprint arXiv:2210.07277*, 2022.

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Bardes, A., Ponce, J., and LeCun, Y. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3286–3295, 2019.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *preprint arXiv:2002.05709*, 2020a.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., and Shen, C. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M. A., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

Ermolov, A., Siarohin, A., Sangineto, E., and Sebe, N. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, 2020.

Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., Singh, M., Reis, V., Caron, M., Bojanowski, P., Joulin, A., and Misra, I. Vissl. `https://github.com/facebookresearch/vissl`, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742, 2006.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. On feature decorrelation in self-supervised learning. In

Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9598–9608, October 2021.

Jabri, A., Owens, A., and Efros, A. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. 2022.

Liutkus, A., Cífka, O., Wu, S.-L., Simsekli, U., Yang, Y.-H., and Richard, G. Relative positional encoding for transformers with linear complexity. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7067–7079. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liutkus21a.html.

Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Salakhutdinov, R. and Hinton, G. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pp. 412–419. PMLR, 2007.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *European Conference on Computer Vision*, 2019.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xiao, T., Wang, X., Efros, A. A., and Darrell, T. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations*.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., and Houlsby, N. A large-scale study of representation

learning with the visual task adaptation benchmark, 2019. URL https://arxiv.org/abs/1910.04867.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014a. URL https://proceedings.neurips.cc/paper/2014/file/3fe94a002317b5f9259f82690aeea4cd-Paper.pdf.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014b.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. Ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Appendix

A. Noise collapse and weight tying

Consider the following loss function where $n_j \sim N(0, \sigma I)$.:

$$J = \Sigma_{i,j} \mathbb{E}_{n_j}[(F(An_j + \psi_j + \tilde{m}, Bx_i) - y_j)^2] \tag{8}$$

Proposition A.1. *If A, B are different set of parameters then $\frac{dJ}{dA}\big|_{A=0} = 0$*

Proof.

$$\frac{\partial J}{\partial A} = \sum_{i,j} \mathbb{E}_{n_j}[\frac{\partial}{\partial A}\|F(An_j + \psi_j + \tilde{m}, Bx_i) - y_j\|^2]$$

$$= \sum_{i,j} \mathbb{E}_{n_j}[2(F(An_j + \psi_j + \tilde{m}, Bx_i) - y_j)\frac{\partial F(An_j + \psi_j + \tilde{m}, Bx_i)}{\partial(An_j + \psi_j + \tilde{m})}n_j^T]$$

Set $A = 0$, then derivative becomes:

$$\frac{\partial J}{\partial A}\bigg|_{A=0} = 2\sum_{i,j}(F(\psi_j + \tilde{m}, Bx_i) - y_j)\frac{\partial F(\psi_j + \tilde{m}, Bx_i)}{\partial(\psi_j + \tilde{m})}\mathbb{E}_{n_j}[n_j^T] = 0$$

$\square$

Define the following the loss with weight tying and the deterministic loss without noise:

$$J_{tied}(A) = J(A, A) = \sum_{i,j} \mathbb{E}_{n_j}[(F(An_j + \psi_j + \tilde{m}, Ax_i) - y_j)^2] \tag{9}$$

$$J_{det}(B) = J(A = 0, B) = \sum_{i,j}[(F(\psi_j + \tilde{m}, Bx_i) - y_j)^2] \tag{10}$$

Proposition A.2. *If $\frac{dJ_{tied}}{dA}\big|_{A=0} = 0$ iff $\frac{dJ_{det}(B)}{dB}\big|_{B=0} = 0$*

Proof. Next, we show that $A = 0$ is a critical point of J_{tied} iff $B = 0$ is a critical point of J_{det}:

$$\frac{\partial J_{tied}}{\partial A}\bigg|_{A=0} = \sum_{i,j}(F(\psi_j + \tilde{m}, 0) - y_j)\nabla F(\psi_i, 0)x_i^T \tag{11}$$

$$\frac{\partial J_{det}}{\partial B}\bigg|_{B=0} = \sum_{i,j}(F(\psi_j + \tilde{m}, 0) - y_j)\nabla F(\psi_j, 0)x_i^T \tag{12}$$

Therefore $\frac{\partial J_{tie}}{\partial A}\big|_{A=0} = 0$ iff $\frac{\partial J_{det}}{\partial B}\big|_{B=0}$ $\square$

B. Optimal Predictor

Consider a random variable X (corresponding to the context in our case. For simplicity assume X is just the positional embedding of the context) that is used to predict a variable Y (corresponding to the target in our case). But now instead of predicting from X, we use a noise variable Z that is independent of both X, Y, and provide the predictor with only the noisy

result $R = g(X, Z)$. Here g is some mixing function (in our case $g(x, z) = x + z$). We next derive the optimal predictor $f(R)$ in this case. Formally we want to minimize:

$$E_{R,Y}[(f(R) - Y)^2] \tag{13}$$

A classic result in estimation is that this is optimized by the conditional expectation $f(r) = E[Y|R = r]$.

We simplify this as follows:

$$
\begin{aligned}
E[Y|R = r] &= \sum_{x,y} y\, p(Y = y, X = x|R = r) \\
&= \sum_{x,y} y\, p(y|X = x)p(X = x|R = r) \\
&= \sum_{x} E[Y|X = x]p(X = x|R = r)
\end{aligned}
$$

where in the second line we used the fact that:

$$p(y, x|r) = p(y|x, r)p(x|r) = p(y|x)p(x|r) \tag{14}$$

To further illustrate, consider the case where z is Gaussian with zero mean and unit variance. Then $p(x|r)$ is also Gaussian with expectation r, and the expression above amounts to convolution of the clean expected values with a Gaussian:

$$E[Y|R = r] = \int_{x} E[Y|X = x]\frac{1}{\sqrt{2\pi}}e^{-0.5(x-r)^2}dx \tag{15}$$

C. Experiments and Results

We include the full implementation details, pretraining configs and evaluation protocols for the Ablations (see Appendix C.1), Downstream Tasks (Appendix C.2), as well as full results and comparisons to invariance-based methods.

C.1. Ablations

Here we pretrain all models for 300 epochs using 4 V100 nodes, on a total batch size of 2048. In all the ablation study experiments, we follow the exact recipe of (Assran et al., 2023). We include the full config in Table 9 for completeness.

To evaluate the pretrained models, we use linear probing evaluation using 1% of IN-1k (Russakovsky et al., 2015). To obtain the features of an image, we apply the target encoder over the image to obtain a sequence of tokens corresponding to the image. We then average the tokens to obtain a single representative vector. The linear classifier is trained over this representation, maintaining the rest of the target encoder layers fixed.

C.2. Downstream Tasks

Here we pretrain I-JEPA with StoP for 600 epochs using 4 V100 nodes, on a total batch size of 2048 using ViT-B (see config in Table 10) and ViT-L (see config in Table 11). For ViT-H we use float16 and train for 300 epochs and follow the config in Table 12. We follow similar configs compared to (Assran et al., 2023) except we usually use a lower learning rate. Intuitively, since StoP is stochastic it is more sensitive to high learning rates.

For evaluation on downstream tasks, we use the features learned by the target-encoder and follow the protocol of VISSL (Goyal et al., 2021) that was utilized by I-JEPA (Assran et al., 2023). Specifically, we report the best linear evaluation number among the average-pooled patch representation of the last layer and the concatenation of the last 4 layers of the average-pooled patch representations. We report full results including comparisons to invariance-based methods for IN-1k linear evaluation in Table 14, IN-1k finetuning results in Table 17 (1%) and Table 15, and other downstream tasks in Table 13.

For baselines that use Vision Transformers (Dosovitskiy et al., 2020) with a `[cls]` token (e.g, iBOT (Zhou et al., 2021), DINO (Caron et al., 2021) or MAE (He et al., 2021)), we use the default configurations of VISSL (Goyal et al., 2021) to evaluate the publicly available checkpoints on iNaturalist18 (Van Horn et al., 2018), CIFAR100 (Krizhevsky et al., 2009),

Clevr/Count (Johnson et al., 2017; Zhai et al., 2019), Clevr/Dist (Johnson et al., 2017; Zhai et al., 2019), and Places205 (Zhou et al., 2014b). Following the evaluation protocol of VISSL (Goyal et al., 2021), we freeze the encoder and return the best number among the `[cls]` token representation of the last layer and the concatenation of the last 4 layers of the `[cls]` token.

For semi-supervised video object segmentation, we propagate the first labeled frame in a video using the similarity between adjacent frames features. To label the video using the frozen features, we follow the code and hyperparams of (Caron et al., 2021). To evaluate the segmented videos, we use the evaluation code of DAVIS 2017 (Pont-Tuset et al., 2017) and include full results in Table 16.

config	value
optimizer	AdamW
epochs	300
learning rate	$1e^{-3}$
weight decay	$(0.04, 0.4)$
batch size	2048
learning rate schedule	cosine decay
warmup epochs	15
encoder arch.	ViT-B
predicted targets	4
predictor depth	6
predictor attention heads	12
predictor embedding dim.	384
σ (noise hyperparam)	0.25

Table 9: **Pretraining setting for ablations**. Using ViT-B encoder, trained for 300 epochs, config strictly follows (Assran et al., 2023).

config	value
optimizer	AdamW
epochs	600
learning rate	$8e^{-4}$
weight decay	$(0.04, 0.4)$
batch size	2048
learning rate schedule	cosine decay
warmup epochs	15
encoder arch.	ViT-B
predicted targets	4
predictor depth	6
predictor attention heads	12
predictor embedding dim.	384
σ (noise hyperparam)	0.25

Table 10: **Pretraining setting for downstream tasks (ViT-B)**. All models trained for 600 epochs.

config	value
optimizer	AdamW
epochs	600
learning rate	$8e^{-4}$
weight decay	$(0.04, 0.4)$
batch size	2048
learning rate schedule	cosine decay
warmup epochs	15
encoder arch.	ViT-L
predicted targets	4
predictor depth	12
predictor attention heads	16
predictor embedding dim.	384
σ (noise hyperparam)	0.25

Table 11: **Pretraining setting for downstream tasks (ViT-L)**. All models trained for 600 epochs.

config	value
optimizer	AdamW
epochs	600
learning rate	$1e^{-3}$
weight decay	$(0.04, 0.4)$
batch size	2048
learning rate schedule	cosine decay
warmup epochs	40
encoder arch.	ViT-H
predicted targets	4
predictor depth	12
predictor attention heads	16
predictor embedding dim.	384
σ (noise hyperparam)	0.2

Table 12: **Pretraining setting for downstream tasks (ViT-H)**. Trained for 300 epochs.

Method	Arch.	CIFAR100	Places205	iNat18	CLEVR/Count	CLEVR/Dist
Invariance-based methods (use extra image augmentations)						
DINO	ViT-B/16	84.8	55.2	50.1	83.2	53.4
iBOT	ViT-B/16	85.5	56.7	50.0	62.1	64.6
	ViT-L/16	88.3	60.4	57.3	85.7	62.8
Masked Image Modeling Methods						
data2vec	ViT-L/16	81.6	54.6	28.1	85.3	71.3
MAE	ViT-B/16	68.1	49.2	26.8	86.6	70.8
	ViT-L/16	77.4	54.4	33.0	**92.1**	73.0
	ViT-H/14	77.3	55.0	32.9	90.5	72.4
I-JEPA	ViT-B/16	69.2	53.4	43.4	82.2	70.7
	ViT-L/16	83.6	56.5	48.4	85.6	71.2
	ViT-H/14	87.5	58.4	47.6	86.7	72.4
+StoP	ViT-B/16	81.2	54.3	44.7	83.7	71.3
	ViT-L/16	84.7	57.2	49.2	85.7	70.2
	ViT-H/14	**87.7**	**58.4**	**50.9**	88.0	**72.5**

Table 13: **Linear-probe transfer for various downstream tasks**. Linear-evaluation on downstream image classification, object counting, and tracking tasks. StoP significantly outperforms previous MIM methods that don't utilize image augmentations like I-JEPA and MAE, and decreases the gap with the best invariance-based methods that utilize data augmentations during pretraining.

Method	Arch.	Epochs	Top-1
Invariance-based methods (use extra image augmentations)			
SimCLR v2	RN152 (2×)	800	79.1
BYOL	RN200 (2×)	800	79.6
DINO	ViT-B/16	400	78.1
	ViT-B/8	300	80.1
MoCo v3	ViT-B/16	300	76.7
	ViT-BN-L/7	300	81.0
MSN	ViT-L/7	200	80.7
iBOT	ViT-B/16	250	79.8
	ViT-L/16	250	81.0
Masked Image Modeling methods			
data2vec	ViT-L/16	1600	77.3
MAE	ViT-B/16	1600	68.0
	ViT-L/16	1600	75.8
	ViT-H/14	1600	77.2
I-JEPA	ViT-B/16	600	72.9
	ViT-L/16	600	77.5
	ViT-H/14	300	79.3
+StoP (ours)	ViT-B/16	600	74.5
	ViT-L/16	600	78.5
	ViT-H/14	300	**79.6**

Table 14: **Linear-evaluation on IN-1k**. Performance of invariance based and MIM approaches.

Method	Arch.	Epochs	Top-1
I-JEPA	ViT-B/16	600	82.9
+StoP (ours)	ViT-B/16	600	**83.1**

Table 15: **Full finetuning results over ImageNet**.

Method	Arch.	J-Mean	F-Mean	J&F Mean
Invariance-based methods (use extra image augmentations)				
DINO	ViT-B/16	60.7	63.9	62.3
iBOT	ViT-B/16	60.9	63.3	62.1
	ViT-L/16	61.7	63.9	62.8
Masked Image Modeling Methods				
MAE	ViT-B/16	49.4	52.6	50.9
	ViT-L/16	52.5	54.3	53.4
	ViT-H/14	54.0	57.0	55.5
I-JEPA	ViT-B/16	56.1	56.2	56.1
	ViT-L/16	56.1	55.7	55.9
	ViT-H/14	58.5	60.9	59.7
+StoP	ViT-B/16	56.6	57.3	57.0
	ViT-L/16	58.1	58.7	58.4
	ViT-H/14	**58.9**	**61.2**	**60.1**

Table 16: **Video objects semi-supervised segmentation.** MIM and Invarianced-based methods. Results reported on DAVIS 2017 dataset.

Method	Arch.	Epochs	Top-1
Invariance-based methods (use extra image augmentations)			
DINO	ViT-B/8	300	70.0
iBOT	ViT-B/16	400	69.7
Masked Image Modeling methods			
MAE	ViT-L/16	1600	67.0
I-JEPA	ViT-L/16	600	69.4
+StoP (ours)	ViT-L/16	600	**71.7**

Table 17: **Finetuning results over ImageNet with 1% labels.** Comparison of MIM and invariance-based methods.