Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture

Mahmoud Assran[1,2,3*] **Quentin Duval**[1] **Ishan Misra**[1] **Piotr Bojanowski**[1]
Pascal Vincent[1] **Michael Rabbat**[1,3] **Yann LeCun**[1,4] **Nicolas Ballas**[1]

[1]Meta AI (FAIR) [2]McGill University [3] Mila, Quebec AI Institute [4]New York University

Abstract

This paper demonstrates an approach for learning highly semantic image representations without relying on hand-crafted data-augmentations. We introduce the Image-based Joint-Embedding Predictive Architecture (I-JEPA), a non-generative approach for self-supervised learning from images. The idea behind I-JEPA is simple: from a single context block, predict the representations of various target blocks in the same image. A core design choice to guide I-JEPA towards producing semantic representations is the masking strategy; specifically, it is crucial to (a) sample target blocks with sufficiently large scale (semantic), and to (b) use a sufficiently informative (spatially distributed) context block. Empirically, when combined with Vision Transformers, we find I-JEPA to be highly scalable. For instance, we train a ViT-Huge/14 on ImageNet using 16 A100 GPUs in under 72 hours to achieve strong downstream performance across a wide range of tasks, from linear classification to object counting and depth prediction.

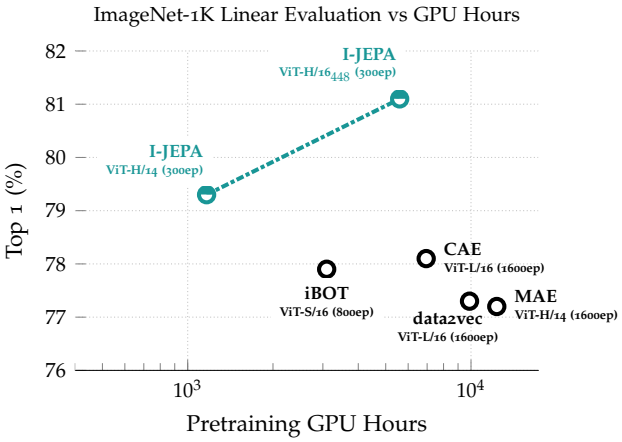


Figure 1. **ImageNet Linear Evaluation**. The I-JEPA method learns semantic image representations without using any view data augmentations during pretraining. By predicting in representation space, I-JEPA produces semantic representations while using less compute than previous methods.

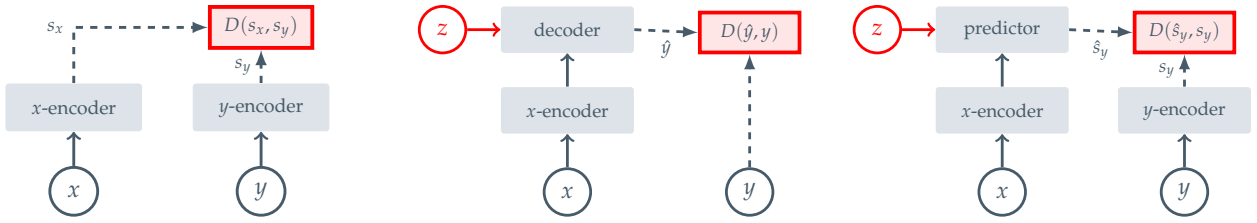
1. Introduction

In computer vision, there are two common families of approaches for self-supervised learning from images: invariance-based methods [1, 4, 10, 17, 18, 24, 35, 37, 74] and generative methods [8, 28, 36, 57].

Invariance-based pretraining methods optimize an encoder to produce similar embeddings for two or more views of the same image [15, 20], with image views typically constructed using a set of hand-crafted data augmentations, such as random scaling, cropping, and color jittering [20], amongst others [35]. These pretraining methods can produce representations of a high semantic level [4, 18], but they also introduce strong biases that may be detrimental for certain downstream tasks or even for pretraining tasks with different data distributions [2]. Often, it is unclear how to generalize these biases for tasks requiring different levels of abstraction. For example, image classification and instance segmentation do not require the same invariances [11]. Additionally, it is not straightforward to generalize these image-specific augmentations to other modalities such as audio.

Cognitive learning theories have suggested that a driving mechanism behind representation learning in biological systems is the adaptation of an internal model to predict sensory input responses [31, 59]. This idea is at the core of self-supervised generative methods, which remove or corrupt portions of the input and learn to predict the corrupted content [9, 36, 57, 67, 68, 71]. In particular, mask-denoising approaches learn representations by reconstructing randomly masked patches from an input, either at the pixel or token level. Masked pretraining tasks require less prior knowledge than view-invariance approaches and easily generalize beyond the image modality [8]. However, the

*massran@meta.com



(a) **Joint-Embedding Architecture** (b) **Generative Architecture** (c) **Joint-Embedding Predictive Architecture**

Figure 2. Common architectures for self-supervised learning, in which the system learns to capture the relationships between its inputs. The objective is to assign a high energy (large scaler value) to incompatible inputs, and to assign a low energy (low scaler value) to compatible inputs. **(a)** Joint-Embedding Architectures learn to output similar embeddings for compatible inputs x, y and dissimilar embeddings for incompatible inputs. **(b)** Generative Architectures learn to directly reconstruct a signal y from a compatible signal x, using a decoder network that is conditioned on additional (possibly latent) variables z to facilitate reconstruction. **(c)** Joint-Embedding Predictive Architectures learn to predict the embeddings of a signal y from a compatible signal x, using a predictor network that is conditioned on additional (possibly latent) variables z to facilitate prediction.

resulting representations are typically of a lower semantic level and underperform invariance-based pretraining in off-the-shelf evaluations (e.g., linear-probing) and in transfer settings with limited supervision for semantic classification tasks [4]. Consequently, a more involved adaptation mechanism (e.g., end-to-end fine-tuning) is required to reap the full advantage of these methods.

In this work, we explore how to improve the semantic level of self-supervised representations without using extra prior knowledge encoded through image transformations. To that end, we introduce a joint-embedding predictive architecture [48] for images (I-JEPA). An illustration of the method is provided in Figure 3. The idea behind I-JEPA is to predict missing information in an abstract representation space; e.g., given a single context block, predict the representations of various target blocks in the same image, where target representations are computed by a learned target-encoder network.

Compared to generative methods that predict in pixel/token space, I-JEPA makes use of abstract prediction targets for which unnecessary pixel-level details are potentially eliminated, thereby leading the model to learn more semantic features. Another core design choice to guide I-JEPA towards producing semantic representations is the proposed multi-block masking strategy. Specifically, we demonstrate the importance of predicting sufficiently large target blocks in the image, using an informative (spatially distributed) context block.

Through an extensive empirical evaluation, we demonstrate that:

- I-JEPA learns strong off-the-shelf representations without the use of hand-crafted view augmentations (cf. Fig.1). I-JEPA outperforms pixel-reconstruction methods such as MAE [36] on ImageNet-1K linear probing, semi-supervised 1% ImageNet-1K, and semantic transfer tasks.
- I-JEPA is competitive with view-invariant pretraining approaches on semantic tasks and achieves better performance on low-level visions tasks such as object counting and depth prediction (Sections 5 and 6). By using a simpler model with less rigid inductive bias, I-JEPA is applicable to a wider set of tasks.
- I-JEPA is also scalable and efficient (Section 7). Pretraining a ViT-H/14 on ImageNet requires less than 1200 GPU hours, which is over $2.5\times$ faster than a ViT-S/16 pretrained with iBOT [79] and over $10\times$ more efficient than a ViT-H/14 pretrained with MAE. Predicting in representation space significantly reduces the total computation needed for self-supervised pretraining.

2. Background

Self-supervised learning is an approach to representation learning in which a system learns to capture the relationships between its inputs. This objective can be readily described using the framework of Energy-Based Models (EBMs) [49] in which the self-supervised objective is to assign a high energy to incompatible inputs, and to assign a low energy to compatible inputs. Many existing generative and non-generative approaches to self-supervised learning can indeed be cast in this framework; see Figure 2.

Joint-Embedding Architectures. Invariance-based pretraining can be cast in the framework of EBMs using a Joint-Embedding Architecture (JEA), which learns to output similar embeddings for compatible inputs, $\boldsymbol{x}, \boldsymbol{y}$, and dissimilar embeddings for incompatible inputs; see Figure 2a. In the context of image-based pretraining, compatible $\boldsymbol{x}, \boldsymbol{y}$ pairs are typically constructed by randomly applying hand-crafted data augmentations to the same input image [20].

The main challenge with JEAs is representation collapse, wherein the energy landscape is flat (i.e., the encoder produces a constant output regardless of the input). During the past few years, several approaches have been investi-

gated to prevent representation collapse, such as contrastive losses that explicitly push apart embeddings of negative examples [15,24,37], non-contrastive losses that minimize the informational redundancy across embeddings [10, 74], and clustering-based approaches that maximize the entropy of the average embedding [4, 5, 18]. There are also heuristic approaches that leverage an asymmetric architectural design between the x-encoder and y-encoder to avoid collapse [8, 24, 35].

Generative Architectures. Reconstruction-based methods for self-supervised learning can also be cast in the framework of EBMs using Generative Architectures; see Figure 2b. Generative Architectures learn to directly reconstruct a signal $\boldsymbol{y}$ from a compatible signal $\boldsymbol{x}$, using a decoder network that is conditioned on an additional (possibly latent) variable $\boldsymbol{z}$ to facilitate reconstruction. In the context of image-based pretraining, one common approach in computer vision is to produce compatible $\boldsymbol{x}, \boldsymbol{y}$ pairs using masking [9, 38] where $\boldsymbol{x}$ is a copy of the image $\boldsymbol{y}$, but with some of the patches masked. The conditioning variable $\boldsymbol{z}$ then corresponds to a set of (possibly learnable) mask and position tokens, that specifies to the decoder which image patches to reconstruct. Representation collapse is not a concern with these architectures as long as the informational capacity of $\boldsymbol{z}$ is low compared to the signal $\boldsymbol{y}$.

Joint-Embedding Predictive Architectures. As shown in Figure 2c, Joint-Embedding Predictive Architectures [48] are conceptually similar to Generative Architectures; however, a key difference is that the loss function is applied in embedding space, not input space. JEPAs learn to predict the embeddings of a signal $\boldsymbol{y}$ from a compatible signal $\boldsymbol{x}$, using a predictor network that is conditioned on an additional (possibly latent) variable $\boldsymbol{z}$ to facilitate prediction. Our proposed I-JEPA provides an instantiation of this architecture in the context of images using masking; see Figure 3.

In contrast to Joint-Embedding Architectures, JEPAs do not seek representations invariant to a set of hand-crafted data augmentations, but instead seek representations that are predictive of each other when conditioned on additional information $\boldsymbol{z}$. However, as with Joint-Embedding Architectures, representation collapse is also a concern with JEPAs; we leverage an asymmetric architecture between the $\boldsymbol{x}$- and $\boldsymbol{y}$-encoders to avoid representation collapse.

3. Method

We now describe the proposed Image-based Joint-Embedding Predictive Architecture (I-JEPA), illustrated in Figure 3. The overall objective is as follows: given a context block, predict the representations of various target blocks

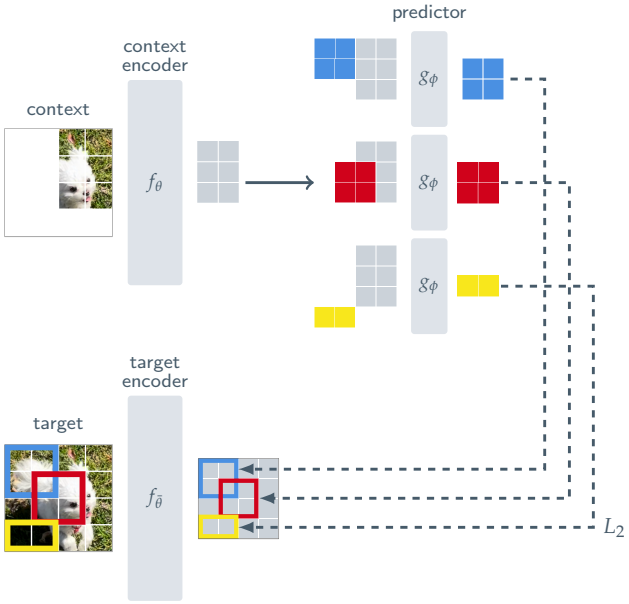


Figure 3. **I-JEPA.** The Image-based Joint-Embedding Predictive Architecture uses a single context block to predict the representations of various target blocks originating from the same image. The context encoder is a Vision Transformer (ViT), which only processes the visible context patches. The predictor is a narrow ViT that takes the context encoder output and, conditioned on positional tokens (shown in color), predicts the representations of a target block at a specific location. The target representations correspond to the outputs of the target-encoder, the weights of which are updated at each iteration via an exponential moving average of the context encoder weights.

in the same image. We use a Vision Transformer [29, 63] (ViT) architecture for the context-encoder, target-encoder, and predictor. A ViT is composed of a stack of transformer layers, each consisting of a self-attention [66] operation followed by a fully-connected MLP. Our encoder/predictor architecture is reminiscent of the generative masked autoencoders (MAE) [36] method. However, one key difference is that the I-JEPA method is non-generative and the predictions are made in representation space.

Targets. We first describe how we produce the targets in the I-JEPA framework: in I-JEPA, the targets correspond to the representations of image blocks. Given an input image $\boldsymbol{y}$, we convert it into a sequence of N non-overlapping patches, and feed this through the target-encoder $f_{\bar{\theta}}$ to obtain a corresponding patch-level representation $\boldsymbol{s}_y = \{\boldsymbol{s}_{y_1}, \ldots, \boldsymbol{s}_{y_N}\}$ where $\boldsymbol{s}_{y_k}$ is the representation associated with the k^{th} patch. To obtain the targets for our loss, we randomly sample M (possibly overlapping) blocks from the target representations $\boldsymbol{s}_y$. We denote by B_i the mask corresponding of the i^{th} block and by $\boldsymbol{s}_y(i) = \{\boldsymbol{s}_{y_j}\}_{j \in B_i}$ its

Why do u sample a context, not just the non-target blocks?

Figure 4. Examples of our context and target-masking strategy. Given an image, we randomly sample 4 target blocks with scale in the range $(0.15, 0.2)$ and aspect ratio in the range $(0.75, 1.5)$. Next, we randomly sample a context block with scale in the range $(0.85, 1.0)$ and remove any overlapping target blocks. Under this strategy, the target-blocks are relatively semantic, and the context-block is informative, yet sparse (efficient to process).

patch-level representation. Typically, we set M equal to 4, and sample the blocks with a random aspect ratio in the range $(0.75, 1.5)$ and random scale in the range $(0.15, 0.2)$. Note that the target blocks are obtained by masking the *output* of the target-encoder, not the input. This distinction is crucial to ensure target representations of a high semantic level; see, e.g., [8].

Context. Recall, the goal behind I-JEPA is to predict the target block representations from a single context block. To obtain the context in I-JEPA, we first sample a single block $\boldsymbol{x}$ from the image with a random scale in the range $(0.85, 1.0)$ and unit aspect ratio. We denote by B_x the mask associated with the context block $\boldsymbol{x}$. Since the target blocks are sampled independently from the context block, there may be significant overlap. To ensure a non-trivial prediction task, we remove any overlapping regions from the context block. Figure 4 shows examples of various context and target blocks in practice. Next, the masked context block, $\boldsymbol{x}$, is fed through the context encoder f_θ to obtain a corresponding patch-level representation $\boldsymbol{s}_x = \{\boldsymbol{s}_{x_j}\}_{j \in B_x}$.

Prediction. Given the output of the context encoder, $\boldsymbol{s}_x$, we wish to predict the M target block representations $\boldsymbol{s}_y(1), \ldots, \boldsymbol{s}_y(M)$. To that end, for a given target block $\boldsymbol{s}_y(i)$ corresponding to a target mask B_i, the predictor $g_\phi(\cdot, \cdot)$ takes as input the output of the context encoder $\boldsymbol{s}_x$ and a mask token for each patch we wish to predict, $\{\boldsymbol{m}_j\}_{j \in B_i}$, and outputs a patch-level prediction $\hat{\boldsymbol{s}}_y(i) = \{\hat{\boldsymbol{s}}_{y_j}\}_{j \in B_i} = g_\phi(\boldsymbol{s}_x, \{\boldsymbol{m}_j\}_{j \in B_i})$. The mask tokens are parameterized by a shared learnable vector with an added positional embedding. Since we wish to make predictions for M target blocks, we apply our predictor M times, each time conditioning on the mask tokens corresponding to the target-block locations we wish to predict, and obtain predictions $\hat{\boldsymbol{s}}_y(1), \ldots, \hat{\boldsymbol{s}}_y(M)$.

Loss. The loss is simply the average L_2 distance between the predicted patch-level representations $\hat{\boldsymbol{s}}_y(i)$ and the target patch-level representation $\boldsymbol{s}_y(i)$; i.e.,

$$\frac{1}{M} \sum_{i=1}^{M} D\left(\hat{\boldsymbol{s}}_y(i), \boldsymbol{s}_y(i)\right) = \frac{1}{M} \sum_{i=1}^{M} \sum_{j \in B_i} \|\hat{\boldsymbol{s}}_{y_j} - \boldsymbol{s}_{y_j}\|_2^2.$$

The parameters of the predictor, ϕ, and context encoder, θ, are learned through gradient-based optimization, while the parameters of the target encoder $\bar{\theta}$ are updated via an exponential moving average of the context-encoder parameters. The use of an exponential moving average target-encoder has proven essential for training JEAs with Vision Transformers [18, 25, 79], we find the same to be true for I-JEPA.

4. Related Work

A long line of work has explored visual representation learning by predicting the values of missing or corrupted sensory inputs. Denoising autoencoders use random noise as input corruption [67]. Context encoders regress an entire image region based on its surrounding [57]. Other works cast image colorization as a denoising task [46, 47, 77].

The idea of image denoising has recently been revisited in the context of masked image modelling [9, 36, 71], where a Vision Transformer [29] is used to reconstruct missing input patches. The work on Masked Autoencoders (MAE) [36] proposed an efficient architecture that only requires the encoder to process visible image patches. By reconstructing missing patches in pixels space, MAE achieves strong performance when fine-tuned end-to-end on large labeled datasets and exhibits good scaling properties. BEiT [9] predicts the value of missing patches in a tokenized space; specifically, tokenizing image patches using a frozen discreteVAE, which is trained on a dataset containing 250 million images [58]. Yet, pixel-level pre-training has been shown to outperform BEiT for fine-tuning [36]. Another work, SimMIM [71], explores reconstruction targets based on the classic Histogram of Gradients [27] feature space, and demonstrates some advantage over pixel space reconstruction. Different from those works, our representation space is learned during training through a Joint-Embedding Predictive Architecture. Our goal is to learn semantic representations that do not require extensive fine-tuning on downstream tasks.

Closest to our work is data2vec [8] and Context Autoencoders [25]. The data2vec method learns to predict the rep-

Method	Arch.	Epochs	Top-1
Methods without view data augmentations			
data2vec [8]	ViT-L/16	1600	77.3
MAE [36]	ViT-B/16	1600	68.0
	ViT-L/16	1600	76.0
	ViT-H/14	1600	77.2
CAE [22]	ViT-B/16	1600	70.4
	ViT-L/16	1600	78.1
I-JEPA	ViT-B/16	600	72.9
	ViT-L/16	600	77.5
	ViT-H/14	300	79.3
	ViT-H/16_{448}	300	**81.1**
Methods using extra view data augmentations			
SimCLR v2 [21]	RN152 (2×)	800	79.1
DINO [18]	ViT-B/8	300	80.1
iBOT [79]	ViT-L/16	250	**81.0**

Table 1. **ImageNet**. Linear-evaluation on ImageNet-1k (the ViT-H/16_{448} is pretrained at at a resolution of 448×448). I-JEPA improves linear probing performance compared to other methods that do not rely on hand-crafted view data-augmentations during pretraining. Moreover, I-JEPA demonstrates good scalability — the larger I-JEPA model matches the performance of view-invariance approaches without requiring view data-augmentations.

Method	Arch.	Epochs	Top-1
Methods without view data augmentations			
data2vec [8]	ViT-L/16	1600	73.3
MAE [36]	ViT-L/16	1600	67.1
	ViT-H/14	1600	71.5
I-JEPA	ViT-L/16	600	69.4
	ViT-H/14	300	73.3
	ViT-H/16_{448}	300	**77.3**
Methods using extra view data augmentations			
iBOT [79]	ViT-B/16	400	69.7
DINO [18]	ViT-B/8	300	70.0
SimCLR v2 [35]	RN151 (2×)	800	70.2
BYOL [35]	RN200 (2×)	800	71.2
MSN [4]	ViT-B/4	300	**75.7**

Table 2. **ImageNet-1%**. Semi-supervised evaluation on ImageNet-1K using only 1% of the available labels. Models are adapted via fine-tuning or linear-probing, depending on whichever works best for each respective method. ViT-H/16_{448} is pretrained at at a resolution of 448×448. I-JEPA pretraining outperforms MAE which also does not rely on hand-crafted data-augmentations during pretraining. Moreover, I-JEPA benefits from scale. A ViT-H/16 trained at resolution 448 surpasses previous methods including methods that leverage extra hand-crafted data-augmentations.

resentation of missing patches computed through an online target encoder; by avoiding handcrafted augmentations, the method can be applied to diverse modalities with promising results in vision, text and speech. Context Autoencoders use an encoder/decoder architecture optimized via the sum of a reconstruction loss and an alignment constraint, which enforces predictability of missing patches in representation space. Compared to these methods, I-JEPA exhibits significant improvements in computational efficiency and learns more semantic off-the-shelf representations. Concurrent to our work, data2vec-v2 [7] explores efficient architectures for learning with various modalities.

We also compare I-JEPA with various methods based on joint-embedding architectures; e.g., DINO [18], MSN [4] and iBOT [79]. Theses methods rely on hand-crafted data augmentations during pretraining to learn semantic image representations. The work on MSN [4], uses masking as an additional data-augmentation during pretraining, while iBOT combines a data2vec-style patch-level reconstruction loss with the DINO view-invariance loss. Common to these approaches is the need to process multiple user-generated views of each input image, thereby hindering scalability. By contrast, I-JEPA only requires processing a single view of each image. We find that a ViT-Huge/14 trained with I-JEPA requires less computational effort than a ViT-Small/16 trained with iBOT.

5. Image Classification

To demonstrate that I-JEPA learns high-level representations without relying on hand-crafted data-augmentations, we report results on various image classification tasks using the linear probing and partial fine-tuning protocols. In this section, we consider self-supervised models that have been pretrained on the ImageNet-1K dataset [60]. Pretraining and evaluation implementation details are described in the Appendix A. All I-JEPA models are trained at resolution 224×224 pixels, unless stated otherwise.

ImageNet-1K. Table 1 shows performance on the common ImageNet-1K linear-evaluation benchmark. After self-supervised pretraining, the model weights are frozen and a linear classifier is trained on top using the full ImageNet-1K training set. Compared to popular methods such as Masked Autoencoders (MAE) [36], Context Autoencoders (CAE) [22], and data2vec [8], which also do not rely on extensive hand-crafted data-augmentations during pretraining, we see that I-JEPA significantly improves linear probing performance, while using less computational effort (see section 7). By leveraging the improved efficiency of I-JEPA, we can train larger models that outperform the best CAE model while using a fraction of the compute. I-JEPA also benefits from scale; in particular, a ViT-H/16 trained at resolution 448×448 pixels matches the performance of view-

Method	Arch.	CIFAR100	Places205	iNat18
Methods without view data augmentations				
data2vec [8]	ViT-L/16	81.6	54.6	28.1
MAE [36]	ViT-H/14	77.3	55.0	32.9
I-JEPA	ViT-H/14	**87.5**	**58.4**	**47.6**
Methods using extra view data augmentations				
DINO [18]	ViT-B/8	84.9	57.9	55.9
iBOT [79]	ViT-L/16	**88.3**	**60.4**	**57.3**

Table 3. **Linear-probe transfer for image classification**. Linear-evaluation on downstream image classification tasks. I-JEPA significantly outperforms previous methods that also do not use augmentations (MAE and data2vec), and decreases the gap with the best view-invariance-based methods that leverage hand-crafted data augmentations during pretraining.

Method	Arch.	Clevr/Count	Clevr/Dist
Methods without view data augmentations			
data2vec [8]	ViT-L/16	85.3	71.3
MAE [36]	ViT-H/14	**90.5**	**72.4**
I-JEPA	ViT-H/14	86.7	**72.4**
Methods using extra data augmentations			
DINO [18]	ViT-B/8	86.6	53.4
iBOT [79]	ViT-L/16	85.7	62.8

Table 4. **Linear-probe transfer for low-level tasks**. Linear-evaluation on downstream low-level tasks consisting of object counting (Clevr/Count) and depth prediction (Clevr/Dist). The I-JEPA method effectively captures low-level image features during pretraining and outperforms view-invariance based methods on tasks such object counting and depth prediction.

invariant approaches such as iBOT [79], despite avoiding the use of hand-crafted data-augmentations.

Low-Shot ImageNet-1K. Table 2 shows performance on the 1% ImageNet benchmark. Here the idea is to adapt the pretrained models for ImageNet classification using only 1% of the available ImageNet labels, corresponding to roughly 12 or 13 images per class. Models are adapted via fine-tuning or linear-probing, depending on whichever works best for each respective method. I-JEPA outperforms MAE while requiring less pretraining epochs when using a similar encoder architecture. I-JEPA, using a ViT-H/14 architecture, matches the performance of a ViT-L/16 pretrained with data2vec [8], while using significantly less computational effort (see Section 7). By increasing the image input resolution, I-JEPA outperforms previous methods including joint-embedding methods that do leverage extra hand-crafted data-augmentations during pretraining, such as MSN [4], DINO [17], and iBOT [79].

Transfer learning. Table 3 shows performance on various downstream image classification tasks using a linear probe. I-JEPA significantly outperforms previous methods that do not use augmentations (MAE and data2vec), and decreases the gap with the best view-invariance-based methods, which leverage hand-crafted data augmentations during pretraining, even surpassing the popular DINO [18] on CIFAR100 and Place205 with a linear probe.

6. Local Prediction Tasks

As demonstrated in Section 5, I-JEPA learns semantic image representations that significantly improve the downstream image classification performance of previous methods, such as MAE and data2vec. Additionally, I-JEPA benefits from scale and can close the gap, and even surpass, view-invariance based methods that leverage extra hand-crafted data augmentations. In this section, we find that I-JEPA also learns local image features and surpasses view-invariance based methods on low-level and dense prediction tasks, such as object counting and depth prediction.

Table 4 shows performance on various low-level tasks using a linear probe. After pretraining, the encoder weights are frozen and a linear model is trained on top to perform object-counting and depth prediction on the Clevr dataset [43]. Compared to view-invariance methods such as DINO and iBOT, the I-JEPA method effectively captures low-level image features during pretraining and outperforms them in object counting (Clevr/Count) and (by a large margin) depth prediction (Clevr/Dist).

7. Scalability

Model Efficiency. I-JEPA is highly scalable compared to previous approaches. Figure 5 shows semi-supervised evaluation on 1% ImageNet-1K as a function of GPU hours. I-JEPA requires less compute than previous methods and achieves strong performance without relying on hand-crafted data-augmentations. Compared to reconstruction-based methods, such as MAE, which directly use pixels as targets, I-JEPA introduces extra overhead by computing targets in representation space (about 7% slower time per iteration). However, since I-JEPA converges in roughly 5× fewer iterations, we still see significant compute savings in practice. Compared to view-invariance based methods, such as iBOT, which rely on hand-crafted data augmentations to create and process multiple views of each image, I-JEPA also runs significantly faster. In particular, a huge I-JEPA model (ViT-H/14) requires less compute than a small iBOT model (ViT-S/16).

Need specifics and concrete info here

Scaling data size. We also find I-JEPA to benefit from pretraining with larger datasets. Table 5 shows transfer

Pretrain	Arch.	CIFAR100	Place205	INat18	Clevr/Count	Clevr/Dist
IN1k	ViT-H/14	87.5	58.4	47.6	86.7	72.4
IN22k	ViT-H/14	**89.5**	57.8	50.5	**88.6**	**75.0**
IN22k	ViT-G/16	**89.5**	**59.1**	**55.3**	86.7	73.0

Almost every model architecture does

Table 5. **Ablating dataset and model size**. Evaluating impact of pre-training dataset size and model size on transfer tasks. I-JEPA benefits from larger more diverse datasets. When increasing the size of the pretraining dataset (IN1k versus IN22k) we see an performance improvement for the ViT-H/14 model. We observe a further performance improvement on semantic tasks by training a larger model ViT-G/16 model on ImageNet-22k. The ViT-H/14 is trained for 300 epochs on IN1k and the equivalent of 900 IN1K epochs on IN22k. The ViT-H/16 is trained for the equivalent of 600 IN1k epochs.

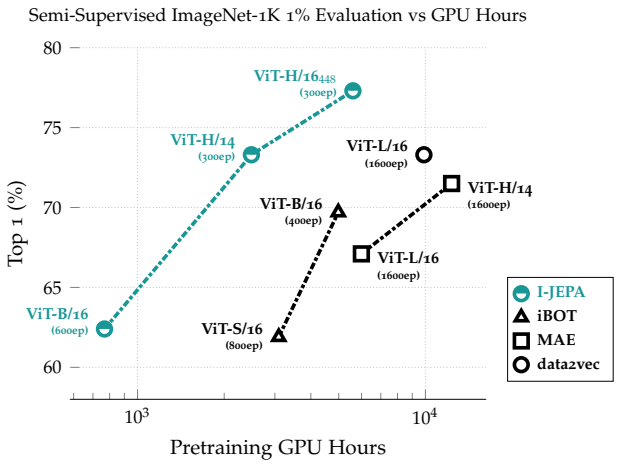


Figure 5. **Scaling**. Semi-supervised evaluation on ImageNet-1K 1% as a function of pretraining GPU hours. I-JEPA requires less compute than previous methods to achieve strong performance. Compared to MAE and data2vec, I-JEPA obtains a significant speedup by requiring fewer pretraining epochs. Compared to iBOT, which relies on hand-crafted data-augmentations, a huge I-JEPA model (ViT-H/14) requires less compute than their smallest model (ViT-S/16).

learning performance on semantic and low level tasks when increasing the size of the pretraining dataset (IN1K versus IN22K). Transfer learning performance on these conceptually different tasks improves when pretraining on a larger more diverse dataset.

Scaling model size. Table 5 also shows that I-JEPA benefit from larger model size when pretraining on IN22K. Pretraining a ViT-G/16 significantly improves the downstream performances on image classification tasks such as Place205 and INat18 compared to a ViT-H/14 model, but does not improve performance on low-level downstream tasks — the ViT-G/16 uses larger input patches, which can be detrimental for the local prediction tasks.

8. Predictor Visualizations

The role of the predictor in I-JEPA is to take the output of the context encoder and, conditioned on positional mask tokens, to predict the representations of a target black at the location specified by the mask tokens. One natural question is whether the predictor conditioned on the positional mask tokens is learning to correctly capture positional uncertainty in the target. To qualitatively investigate this question, we visualize the outputs of the predictor. We use the following visualization approach to enable the research community to independently reproduce our findings. After pretraining, we freeze the context-encoder and predictor weights, and train a decoder following the RCDM framework [13] to map the average-pool of the predictor outputs back to pixel space. Figure 6 shows decoder outputs for various random seeds. Qualities that are common across samples represent information that is contained in the average-pooled predictor representation. The I-JEPA predictor correctly captures positional uncertainty and produces high-level object parts with the correct pose (e.g., back of the bird and top of the car).

9. Ablations

Predicting in representation space. Table 7 compares low-shot performance on 1% ImageNet-1K using a linear probe when the loss is computed in pixel-space versus representation space. We conjecture that a crucial component of I-JEPA is that the loss is computed entirely in representation space, thereby giving the target encoder the ability to produce abstract prediction targets, for which irrelevant pixel-level details are eliminated. From Table 7, it is clear that predicting in pixel-space leads to a significant degradation in the linear probing performance.

Masking strategy. Table 6 compare our `multi-block` masking with other masking strategies such as `rasterized` masking, where the image is split into four large quadrants, and the goal is to use one quadrant as a context to predict the other three quadrants, and the traditional `block` and `random` masking typically used in reconstruction-based methods. In `block` masking, the target is a single image block and the context is the

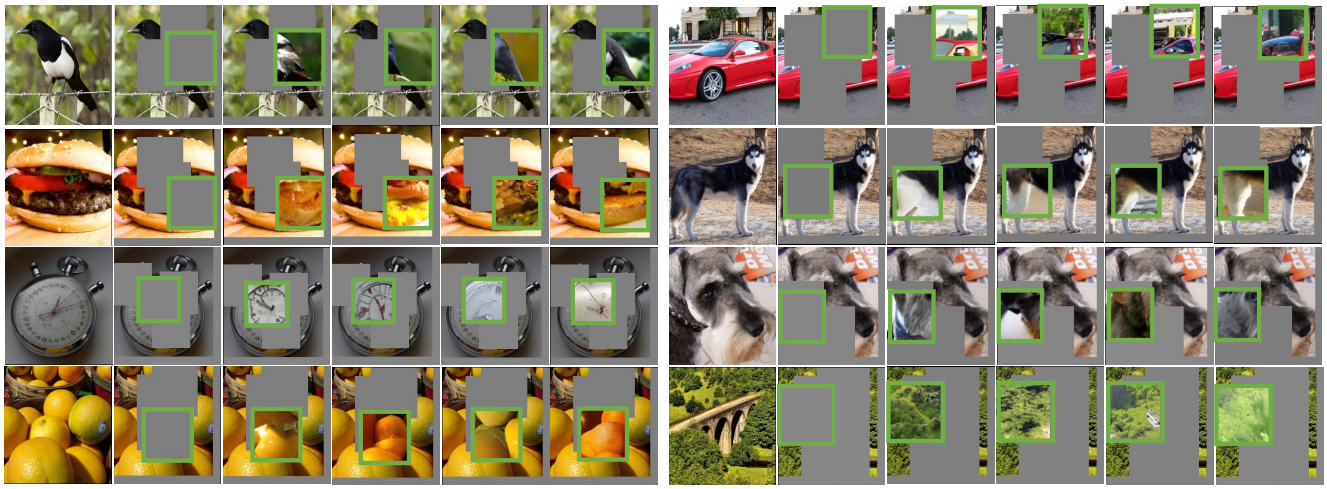


Figure 6. **Visualization of I-JEPA predictor representations**. For each image: first column contains the original image; second column contains the context image, which is processed by a pretrained I-JEPA ViT-H/14 encoder. Green bounding boxes in subsequent columns contain samples from a generative model decoding the output of the pretrained I-JEPA predictor, which is conditioned on positional mask tokens corresponding to the location of the green bounding box. Qualities that are common across samples represent information that contained is in the I-JEPA prediction. The I-JEPA predictor is correctly capturing positional uncertainty and producing high-level object parts with a correct pose (e.g., the back of the bird and top of a car). Qualities that vary across samples represent information that is not contained in the representation. In this case, the I-JEPA predictor discards the precise low-level details as well as background information.

	Targets		**Context**		
Mask	Type	Freq.	Type	Avg. Ratio*	**Top-1**
`multi-block`	Block$(0.15, 0.2)$	4	Block$(0.85, 1.0)$ $\times$ Complement	0.25	**54.2**
`rasterized`	Quadrant	3	Complement	0.25	15.5
`block`	Block(0.6)	1	Complement	0.4	20.2
`random`	Random(0.6)	1	Complement	0.4	17.6

**Avg. Ratio is the average number of patches in the context block relative to the total number of patches in the image.*

Table 6. **Ablating masking strategy**. Linear evaluation on ImageNet-1K using only 1% of the available labels after I-JEPA pretraining of a ViT-B/16 for 300 epochs. Comparison of proposed `multi-block` masking strategy. In `rasterized` masking the image is split into four large quadrants; one quadrant is used as a context to predict the other three quadrants. In `block` masking, the target is a single image block and the context is the image complement. In `random` masking, the target is a set of random image patches and the context is the image complement. The proposed `multi-block` masking strategy is helpful for guiding I-JEPA to learn semantic representations.

Targets	**Arch.**	**Epochs**	**Top-1**
Target-Encoder Output	ViT-L/16	500	**66.9**
Pixels	ViT-L/16	800	40.7

Table 7. **Ablating targets**. Linear evaluation on ImageNet-1K using only 1% of the available labels. The semantic level of the I-JEPA representations degrades significantly when the loss is applied in pixel space, rather than representation space, highlighting the importance of the target-encoder during pretraining.

image complement. In `random` masking, the target is a set of random patches and the context is the image complement. Note that there is no overlap between the context and target blocks in all considered strategies. We find `multi-block` masking helpful for guiding I-JEPA to learning semantic representations. Additional ablations on `multi-block` masking can be found in Appendix C.

10. Conclusion

We proposed I-JEPA, a simple and efficient method for learning semantic image representations without relying on hand-crafted data augmentations. We show that by predicting in representation space, I-JEPA converges faster than pixel reconstruction methods and learns representations of a high semantic level. In contrast to view-invariance based methods, I-JEPA highlights a path for learning general representations with joint-embedding architectures, without relying on hand-crafted view augmentations.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *Internatinoal Conference on Learning Representations*, 2020. 1

[2] Mahmoud Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Nicolas Ballas. The hidden uniform cluster prior in self-supervised learning. *International Conference on Learning Representations*, 2023. 1, 13

[3] Mahmoud Assran, Nicolas Ballas, Lluis Castrejon, and Michael Rabbat. Supervision accelerates pre-training in contrastive semi-supervised learning of visual representations. *NeurIPS Workshop on Self-Supervised Learning*, 2020. 13

[4] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *European Conference on Computer Vision*, 2022. 1, 2, 3, 5, 6, 12, 13, 16, 17

[5] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *IEEE/CVF International Conference on Computer Vision*, 2021. 3, 13

[6] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019. 13

[7] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. *arXiv preprint arXiv:2212.07525*, 2022. 5

[8] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022. 1, 3, 4, 5, 6, 13

[9] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 3, 4, 13

[10] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1, 3, 13

[11] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022. 1, 13

[12] Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Improving deep networks generalization by removing their head. *arXiv preprint arXiv:2206.13378*, 2022. 13

[13] Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*, 2022. 7, 16

[14] John Bridle, Anthony Heading, and David MacKay. Unsupervised classifiers, mutual information and'phantom targets. *Advances in neural information processing systems*, 4, 1991. 13

[15] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993. 1, 3

[16] Zhaowei Cai, Avinash Ravichandran, Paolo Favaro, Manchen Wang, Davide Modolo, Rahul Bhotika, Zhuowen Tu, and Stefano Soatto. Semi-supervised vision transformers at scale. *arXiv preprint arXiv:2208.05688*, 2022. 13

[17] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 6

[18] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 1, 3, 4, 5, 6, 12, 13

[19] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 13

[20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *preprint arXiv:2002.05709*, 2020. 1, 2, 13

[21] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 5

[22] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 5

[23] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 12, 13

[24] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 1, 3, 13

[25] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 4

[26] Yubei Chen, Adrien Bardes, Zengyi Li, and Yann LeCun. Intra-instance vicreg: Bag of self-supervised image patch embedding. *arXiv preprint arXiv:2206.08954*, 2022. 13

[27] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 4

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 4, 12, 13

[30] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 13

[31] Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005. 1

[32] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6928–6938, 2020. 13

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 13

[34] Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. Vissl. https://github.com/facebookresearch/vissl, 2021. 12

[35] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 3, 5, 12, 13

[36] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 4, 5, 6, 12, 13, 15, 16

[37] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 3, 12, 13

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[39] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020. 13

[40] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 13

[41] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017. 13

[42] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 12

[43] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 6

[44] Andreas Krause, Pietro Perona, and Ryan Gomes. Discriminative clustering by regularized information maximization. *Advances in neural information processing systems*, 23, 2010. 13

[45] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 12

[46] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. 2016. 4

[47] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. 2017. 4

[48] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. 2022. 2, 3

[49] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2

[50] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. 13

[51] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 12

[52] Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, pages 1–26, 2022. 13

[53] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 13

[54] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *International Conference on Learning Representations*, 2021. 13

[55] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 13

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 12

[57] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 1, 4

[58] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever.

Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 4

[59] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999. 1

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5, 12

[61] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017. 12

[62] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021. 13

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[64] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019. 13

[65] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 12

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[67] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 1, 4, 13

[68] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021. 1, 13

[69] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 13

[70] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019. 13

[71] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021. 1, 4

[72] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. 12

[73] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 16

[74] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 1, 3, 13

[75] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2019. 12

[76] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Internatinoal Conference on Learning Representations*, 2018. 16

[77] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. 2016. 4

[78] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27, 2014. 12

[79] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations*, 2022. 2, 4, 5, 6, 12, 13

A. Implementation Details

A.1. Pretraining

Architectures. For I-JEPA pretraining, we use Vision Transformer [29] (ViT) architectures for the context-encoder, target-encoder, and the predictor. While the context-encoders and target-encoders correspond to standard ViT architectures, the predictor is designed as a light-weight (narrow) ViT architecture. Specifically, we fix the embedding dimension of the predictor to 384, while keeping the number of self-attention heads equal to that of the backbone context-encoder. For the smaller ViT-B/16 context-encoder, we set the depth of the predictor to 6. For ViT-L/16, ViT-H/16, and ViT-H/14 context-encoders, we set the depth of the predictor to 12. Finally, the ViT-G/16 uses a predictor of depth 16. I-JEPA is pretrained without a `[cls]` token. We use the target-encoder for evaluation and average pool its output to produce a global image representation.

Optimization. We use AdamW [51] to optimize the context-encoder and predictor weights. Our default batch-size is 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. Following [4, 18], the weight-decay is linearly increased from 0.04 to 0.4 throughout pretraining. The target-encoder weights are identical to the context-encoder weights at initialization, and updated via an exponential moving average thereafter [4, 18, 23, 35, 37, 61]. We use a momentum value of 0.996, and linearly increase this value to 1.0 throughout pretraining, following [4, 18].

Masking. By default, we sample 4 possibly overlapping target blocks masks with random scale in the range $(015, 0.2)$ and aspect ratio in the range $(0.75, 1.5)$. We sample 1 context block mask with random scale in the range $(0.85, 1.0)$ and unit aspect ratio. We subsequently eliminate any regions in the context block mask that overlap with any of the 4 target block masks. The context-block mask and target-block masks are sampled independently for each image in the mini-batch. To ensure efficient batch processing, we restrict the size of all context masks on a co-located GPU to be identical. Similarly, we restrict the size of all target masks on a co-located GPU to be identical. The mask-sampler is efficiently implemented in only a few lines of code in PyTorch [56] using a batch-collator function, which runs in the data loader processes. In short, in each iteration, the data loader returns a mini-batch of images and a set of context and target masks for each image, identifying the patch indices to keep for the context and target views.

A.2. Downstream Tasks

Linear evaluation. When evaluating methods such as iBOT [79], DINO [18] or MAE [36], which leverage Vision Transformers [29] with an additional `[cls]` token, we use the default configurations of VISSL [34] to evaluate all the models on iNaturalist18 [65], CIFAR100 [45], Clevr/Count [42, 75], Clevr/Dist [42, 75], and Places205 [78]. We freeze the encoder and return the best number among the following representations: 1) the `[cls]` token representation of the last layer, 2) the concatenation of the last 4 layers of the `[cls]` token. For each representation, we try two different heads: 1) a linear head, or 2) a linear head preceded by a batch normalization, and return the best number. We use the default data augmentations of VISSL [34]: random resize cropping and horizontal flipping, with the exception of Clevr/Count and Clevr/Dist, where we only use center crop and horizontal flipping, as random cropping interferes with the capability of counting objects and estimating distance, removing critical objects from the scene. For CIFAR100, we resize the images to 224×224 pixels, so as to keep the number of patches equal to that used during pretraining.

Because our I-JEPA implementation uses Vision Transformer architectures without a `[cls]` token, we adapt the default VISSL evaluation recipe to utilize the average-pooled patch representation instead of the `[cls]` token. We therefore report the best linear evaluation number among the following representations: 1) the average-pooled patch representation of the last layer, 2) the concatenation of the last 4 layers of the average-pooled patch representations. We otherwise keep the linear-probing recipe identical.

ImageNet evaluations. To evaluate the I-JEPA on ImageNet [60], we adapt the VISSL recipe to use average pooled representations instead of the `[cls]` token. Following MAE [36], we use the LARS [72] optimizer with a batch-size of 16384, and train the linear probe for 50 epochs. We use a learning rate with a step-wise decay, dividing it by a factor of 10 every 15 epochs, and sweep three different reference learning rates $[0.01, 0.05, 0.001]$, and two weight decay values $[0.0005, 0.0]$.

Low-shot evaluation. To evaluate our model on the ImageNet-1% low-shot task, we adapt the fine-tuning protocol of MAE [36].We fine-tune our ViT-L/H models for 50 epochs on ImageNet-1% with the AdamW optimizer and a cosine learning rate scheduler. We use a batch size of 512, a learning rate layer decay of 0.75 and 0.1 label smoothing. We use the default randaugment data-augmentations as in MAE. In contrast to the fine-tuning done with MAE, we do not use mixup, cutmix, random erasing or drop path. For the I-JEPA, we use a learning rate /weight decay of $3e^{-5}/5e^{-2}$ for the ViT-L/16, $3e^{-5}/4e^{-1}$ for the ViT-H/14 and $3e^{-5}/4e^{-1}$ for the ViT-H/16_{448}. Similar fine-tuning strategy for low-shot learning has been explored by Semi-VIT in the context of semi-supervised learning [16].

B. Broader Related Work

Self-supervised learning of visual representations with joint-embedding architectures is an active line of research [3, 10, 12, 18, 23, 24, 35, 37, 54, 69, 79]. These approaches train a pair of encoders to output similar embeddings for two or more views of the same image. To avoid pathological solution, many popular joint-embedding approaches use explicit regularization [5, 10, 18, 20] or architectural constraints [24, 35]. Collapse-prevention based on architectural constraints leverage specific network design choices to avoid collapse, for example, by stopping the gradient flow in one of the joint-embedding branches [20], using a momentum encoder in one of the joint-embedding branches [35], or using an asymmetric prediction head [8, 20, 35]. Recent work [62] attempts to theoretically understand (in certain simplified settings) how joint-embedding methods with architectural constraints avoid representation collapse without explicit regularization.

Typical regularization-based approaches to collapse prevention in joint-embedding architectures try to maximize the volume of space occupied by the representations. This is often motivated through the InfoMax [52] principle. Indeed, a long-standing conviction in unsupervised representation learning is that the resulting representations should be both maximally informative about the inputs, while also satisfying certain simplicity constraints [33,50]. The former objective is often referred to as the information-maximization principle (InfoMax), while the latter is sometimes referred to as the parsimony principle [52]. Such approaches to representation learning have been proposed for decades (e.g., [14]), where, historically, simplicity constraints were enforced by encouraging the learned representations to be sparse, low-dimensional, or disentangled, i.e., the individual dimensions of the representation vector should be statistically independent [33]. Modern approaches enforce the simplicity constraints coupled with InfoMax regularization through self-supervised loss terms [6, 40, 41, 44, 55, 64]. One example is the widespread view-invariance penalty [53], often coupled with with independence [10,74] or low-dimensionality constraints, e.g., by projecting representations on the unit hypersphere [20, 35, 37]. However, despite its proliferation, there have also been many criticisms of the InfoMax principle, especially since it is does not discriminate between different types of information (e.g, noise and semantics) [2]. Indeed, the sets of features we wish the model to capture are not always those with the highest marginal entropy (maximal information content).

Orthogonal to the contributions of invariance-based pretraining, another line of work attempts to learn representations by artificially masking parts of the input and training a network to reconstruct the hidden content [67]. Autoregressive models, and denoising autoencoders in particular, predict clean visual inputs from noisy views [8, 9, 19, 36, 67]. Typically, the goal is to predict missing inputs at a pixel level [29, 36, 70], or at a patch token-level, using a tokenizer [9, 68]. While these works demonstrate impressive scalability, they usually learn features at a low-level of semantic abstraction compared to joint-embedding approaches [4].

More recently, a set of approaches attempt to combine both joint-embedding architectures and reconstruction based approaches [30], wherein they combine an invariance pretraining loss with a patch-level reconstruction loss, as in the iBOT method [79]. Since view-invariance based approaches are typically biased towards learning global image representations, thereby limiting their applicability to other computer vision tasks, the idea is that adding local loss terms can improve performance on other popular tasks in computer vision [11, 26, 32]. The framework of contrastive predictive coding [55] is also closely related to this line of work on local loss terms. In the context of images [39], here the idea is to use a contrastive objective combined with a convolutional network to discriminate between overlapping image patch representations. Specifically, the goal is to encourage the representations of an image patch to be predictable of the image patches directly below it, while pushing away the representations of other patch views. In contrast to that work, the proposed I-JEPA method is non-contrastive and does not seek to discriminate between image patches. Rather, the goal is to predict the representations of various target blocks from a single context block. This is achieved with a Joint-Embedding Predictive Architecture, using a predictor network that is conditioned on positional embeddings corresponding to the location of the target block in the image. Qualitative experiments in Section 8 show that the predictor network in our architecture learns to correctly perform this local-to-local region feature mapping, and learns to correctly capture positional uncertainty in the image.

Targets		Context	
Scale	Freq.	Scale	**Top-1**
(0.075, 0.2)	4	(0.85, 1.0)	19.2
(0.1, 0.2)	4	(0.85, 1.0)	39.2
(0.125, 0.2)	4	(0.85, 1.0)	42.4
(0.15, 0.2)	4	(0.85, 1.0)	**54.2**
(0.2, 0.25)	4	(0.85, 1.0)	38.9
(0.2, 0.3)	4	(0.85, 1.0)	33.6

Table 8. **Ablation of the target block size for multi-block masking**. Linear evaluation on 1% ImageNet-1K (using only 1% of the available labels); ablating the multi-block target size during I-JEPA pretraining of a ViT-B/16 for 300 epochs. Predicting larger (semantic) blocks improves the low-shot accuracy as long as the context is sufficiently informative.

Targets		Context	
Scale	Freq.	Scale	**Top-1**
(0.15, 0.2)	4	(0.40, 1.0)	31.2
(0.15, 0.2)	4	(0.65, 1.0)	47.1
(0.15, 0.2)	4	(0.75, 1.0)	49.3
(0.15, 0.2)	4	(0.85, 1.0)	**54.2**

Table 9. **Ablation of the context size for multi-block masking**. Linear evaluation on 1% ImageNet-1K (using only 1% of the available labels); ablating the multi-block target size during I-JEPA pretraining of a ViT-B/16 for 300 epochs. Reducing the multi-block context size degrades the low-shot performance.

Targets		Context	
Scale	Freq.	Scale	**Top-1**
(0.15, 0.2)	1	(0.85, 1.0)	9.0
(0.15, 0.2)	2	(0.85, 1.0)	22.0
(0.15, 0.2)	3	(0.85, 1.0)	48.5
(0.15, 0.2)	4	(0.85, 1.0)	**54.2**

Table 10. **Ablation of the targets number for multi-block masking**. Linear evaluation on 1% ImageNet-1K (using only 1% of the available labels); ablating the multi-block number of targets during I-JEPA pretraining of a ViT-B/16 for 300 epochs. Increasing the number of target blocks improve the low-shot accuracy.

C. Additional Ablations

This section follows the same experimental protocol as Section 9. We report the result of a linear probe with a frozen backbone, trained on the low-shot 1% ImageNet-1K benchmark.

Multiblock masking strategy. We present an extended ablation of the multiblock masking strategy where we change the targets block scale (Table 8), the context scale (Table 9) and the number of target blocks (Table 10). We train a ViT-B/16 for 300 epochs using I-JEPA with various `multi-block` settings and compare performance on the 1% ImageNet-1K benchmark using a linear probe. In short, we find that it is important to predict several relatively large (semantic) target blocks, and to use a sufficiently informative (spatially distributed) context block.

Masking at the output of the target-encoder. An important important design choice in I-JEPA is that the target blocks are obtained by masking the *output* of the target-encoder, not the input. Table 11 shows the effect of this design choice on the semantic level of the learned representations when pretraining a ViT-H/16 using I-JEPA for 300 epochs. In the case where masking is applied to the input, we forward-propagate through the target-encoder once for each target region. Masking the output of the target-encoder during pretraining results in more semantic prediction targets and improves linear probing performance.

Target Masking	Arch.	Epochs	Top-1
Output	ViT-H/16	300	**67.3**
Input	ViT-H/16	300	56.1

Table 11. **Ablating masking output of target encoder**. Linear evaluation on ImageNet-1K using only 1% of the available labels; ablating the effect of masking the target-encoder output during I-JEPA pretraining of a ViT-H/16 for 300 epochs. Masking the output of the target-encoder during pretraining significantly improves the linear probing performance of the pretrained representations.

Predictor depth. We examine the impact of the predictor depth on the downstream low-shot performance in Table 12. We pretrain a ViT-L/16 for 500 epochs using either a 6-layer predictor network or a 12-layer predictor network. The model pretrained using a deeper predictor shows a significant improvement in downstream low-shot performance compared to the model pretrained with a shallower predictor.

Predictor Depth	Arch.	Epochs	Top-1
6	ViT-L/16	500	64.0
12	ViT-L/16	500	**66.9**

Table 12. **Ablating the predictor depth**. Linear evaluation on ImageNet-1K using only 1% of the available labels; ablating the effect of masking the predictor depth for a ViT-L/16 pretrained for 500 epochs. Increasing the predictor depth leads to significant improvement of the linear probe performance of the pretrained representations.

Weight decay. In Table 13, we evaluate the impact of weight-decay during pretraining. We explore two weight decay strategies: linearly increase the weight-decay from 0.04 to 0.4 or use a fix weight-decay of 0.05. Using a smaller weight decay during pretraining improves the downstream performance on ImageNet-1% when fine-tuning. However, this also leads to a degradation of performance in linear evaluation. In the main paper, we use the first weight decay strategy as it improves the performances in linear evaluation downstream tasks.

Weight Decay	Arch.	Epochs	ImageNet-1%	ImageNet Linear-Eval
$0.04 \rightarrow 0.4$	ViT-L/16	600	69.4	**77.8**
0.05	ViT-L/16	600	**70.7**	76.4

Table 13. **Ablating the pretraining weight-decay**. We compare our default pretraining weight decay strategy where we linearly increase the weight-decay from 0.04 to 0.4 to using a fix weight decay of 0.05. Using a smaller weight-decay during pretraining can improve the fine-tuning performance on ImageNet-1%, However, it also leads to a drop of performance in linear evaluation.

Predictor width. We explore the impact of the predictor width in Table 14. We compare I-JEPA using a ViT-L encoder and a predictor with 386 channels to a similar model using a predictor with 1024 channels. Note that the ViT-L encoder has 1024 channels. Using a bottleneck in the predictor width improves the downstream performance on ImageNet 1%.

Predictor Width	Arch.	Epochs	Top-1
384	ViT-L/16	600	**70.7**
1024	ViT-L/16	600	68.4

Table 14. **Ablating the predictor width.** We reports results on ImageNet-1K 1% using fine-tuning. We compare two predictors having a width of either 384 or 1024. Note the I-JEPA encoder is a ViT-L with 1024 channels. Having a width bottleneck in the predictor improves the downstream performances.

D. Finetuning on the full ImageNet

In this section, we report performance on I-JEPA when fine-tuning on the full ImageNet dataset. We focus on the ViT-H/16_{448} as this architecture achieves state-of-art performance with MAE [36].

We use a fine-tuning protocol similar to MAE. Specifically, we fine-tune our model for 50 epochs using AdamW and a cosine learning rate schedule. The base learning rate is set to 10^{-4} and the batch size to 528. We train using mixup [76] set to 0.8, cutmix [73] set to 1.0, a drop path probability of 0.25 and a weight decay set to 0.04. We also use a layer decay of 0.75. Finally, we use the same rand-augment data-augmentations as MAE,

Table 15 reports the fine-tuning results. I-JEPA achieves 87.1 top-1 accuracy. Its performance is less than 1% away from the best MAE model despite I-JEPA being trained for 5.3 times less epochs than MAE. This result demonstrates that I-JEPA is competitive when fine-tuning on the full ImageNet dataset.

Method	Arch.	Epochs	Top-1
MAE [36]	ViT-H/14_{448}	1600	**87.8**
I-JEPA	ViT-H/16_{448}	300	87.1

Table 15. **Finetuning on the full ImageNet dataset**. I-JEPA achieves competitive performance. I-JEPA is close to MAE approach despite I-JEPA being trained for 5.3 times less epochs than MAE.

E. RCDM Visualizations

To visualize the representations of a pretrained neural network in pixel space, we use the RCDM framework [13]. The RCDM framework trains a decoder network h_ω, comprising a generative diffusion model, to reconstruct an image $\boldsymbol{x}$ from the representation vector of that image $\boldsymbol{s}_x$ and a noisy version of that image $\hat{\boldsymbol{x}} := \boldsymbol{x} + \epsilon$, where ϵ is an additive noise vector. Concretely, the decoder objective is to minimize the loss function $\|h_\omega(\hat{\boldsymbol{x}}, \boldsymbol{s}_x) - \epsilon\|$. We train each RCDM network for 300,000 iterations using the default hyperparameters [13]. After training the decoder, one can subsequently feed the representation vector of an unseen test image $\boldsymbol{s}_y$ into the decoder along with various random noise vectors to generate several pixel-level visualizations of the representation, thus providing insight into the features captured in the representations of the pretrained network. Qualities that are common across samples represent information that is contained in the representation. On the other hand, qualities that vary across samples represent information that is not contained in the representations

In Figure 6, the visualizations are obtained by feeding the average-pooled output of the predictor, conditioned on a specific target region, into the decoder network, along with various random noise vectors. In Figures 7 and 8, the visualizations are obtained by feeding the average-pooled output of the target-encoder into the decoder network, along with various random noise vectors.

E.1. Encoder Visualization

In Figure 7, we visualize the average-pooled I-JEPA representations at the output of our ViT-H/14 target-encoder. The first column contains the original image, while subsequent columns contain synthetic samples obtained by feeding the average-pooled representation of the image into the decoder along with various random noise vectors. Figure 7 suggests that the I-JEPA target-encoder is able to correctly capture the high-level information regarding objects and their poses, while discarding low-level image details and background information.

Figure 8 shows similar visualizations, but when using an MSN [4] pretrained ViT-L/7 target-encoder to compute the image representations. The MSN method trains a context- and target-encoder using a Joint-Embedding Architecture to enforce invariance of global image representations to various hand crafted data augmentations and missing patches. While the MSN pretrained network is able to capture high level semantic information about the image in the first column, it also exhibits higher variability in the generated samples, e.g., variability in the object pose, object scale, and number of instances. In short, the MSN pretrained discards much of the local structure in the image, which is in stark contrast to I-JEPA, which retains information about much of the local structure in the input image.

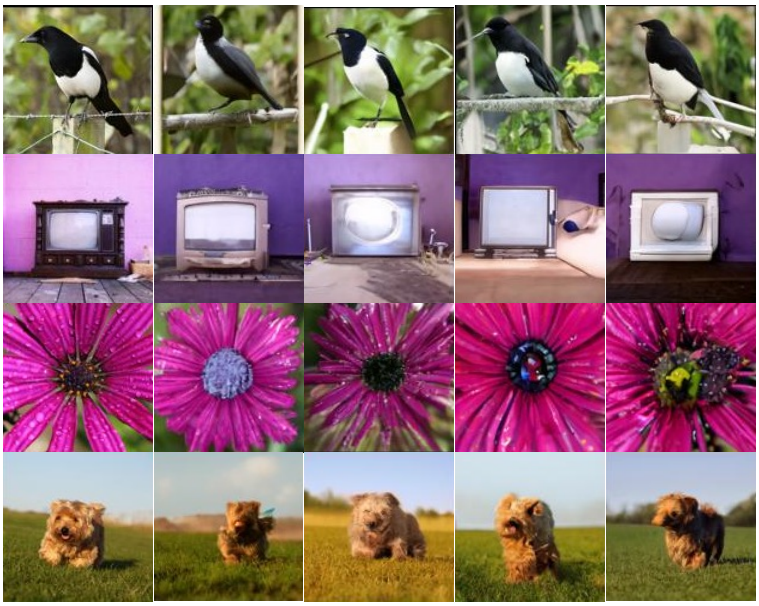


Figure 7. **Visualization of I-JEPA target-encoder representations**. For each image: first column contains the original image; subsequent columns contain samples from a generative model decoding the average-pooled output of a pretrained I-JEPA target-encoder. Qualities that are common across samples represent information that contained is in the I-JEPA representation. I-JEPA is able to correctly capture the high-level information regarding objects and their poses. Qualities that vary across samples represent information that is not contained in the representation. I-JEPA encoder discards the precise low-level details as well as background information.

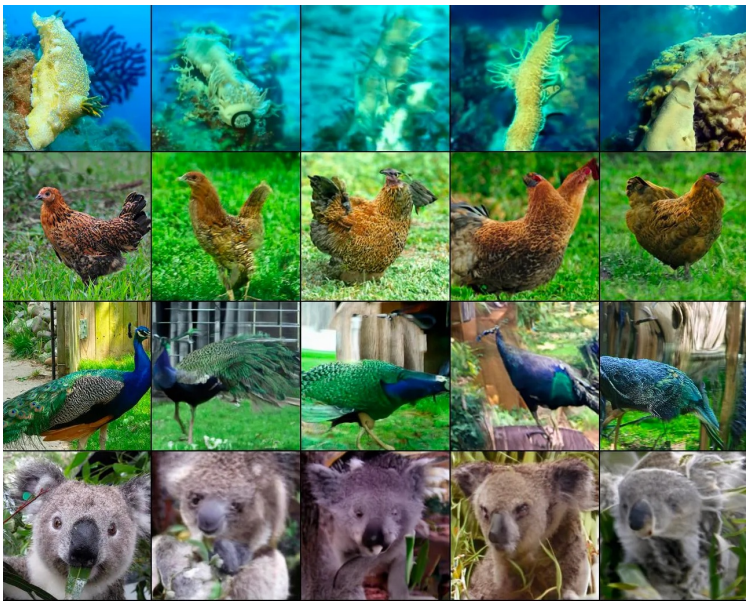


Figure 8. **Visualization of MSN target-encoder representations**. For each image: first column contains the original image; subsequent columns contain samples from a generative model decoding the output of a frozen MSN encoder [4]. Qualities that are common across samples represent information that is contained in the representation. Qualities that vary across samples represent information that is not captured by MSN. Compared to I-JEPA, MSN samples show higher variability. MSN retains less information from the input. In particular, it discards global structure information such as the object pose or even number of instances.