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.

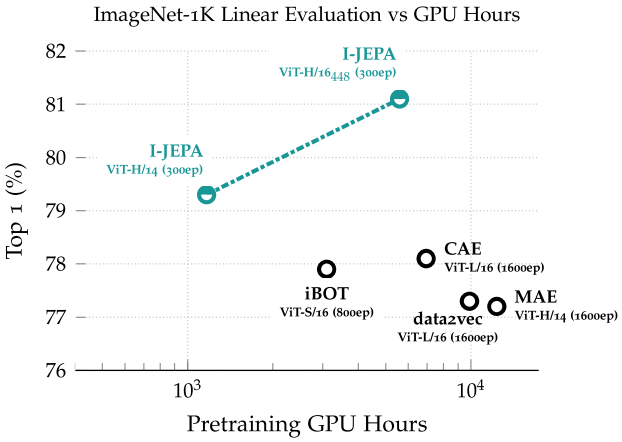


Figure 1. **ImageNet Linear Evaluation**. The I-JEPA method learns semantic image representations without using any view data augmentations during pretraining. By predicting in representation space, I-JEPA produces semantic representations while using less compute than previous methods.

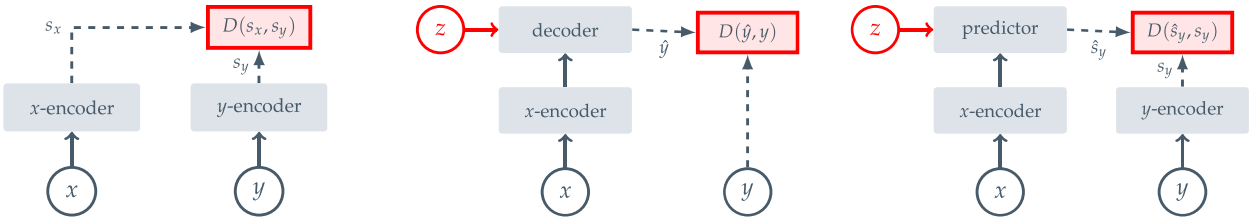
1. Introduction

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

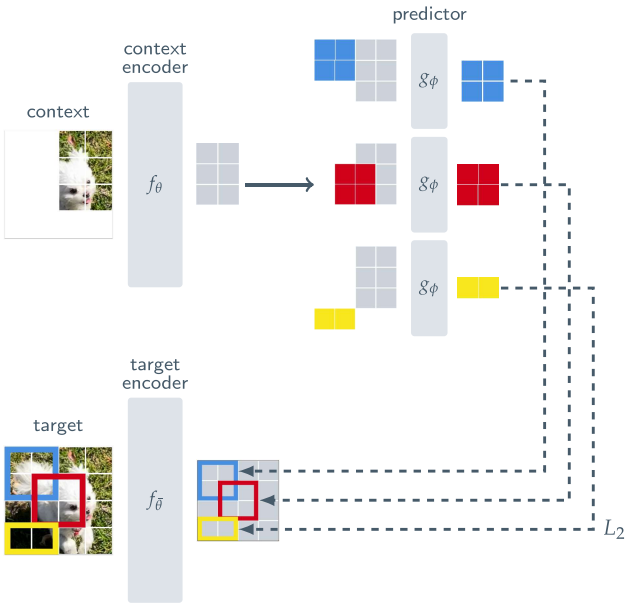


Figure 3. **I-JEPA.** The Image-based Joint-Embedding Predictive Architecture uses a single context block to predict the representations of various target blocks originating from the same image. The context encoder is a Vision Transformer (ViT), which only processes the visible context patches. The predictor 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

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 $5\times$ 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).

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

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.

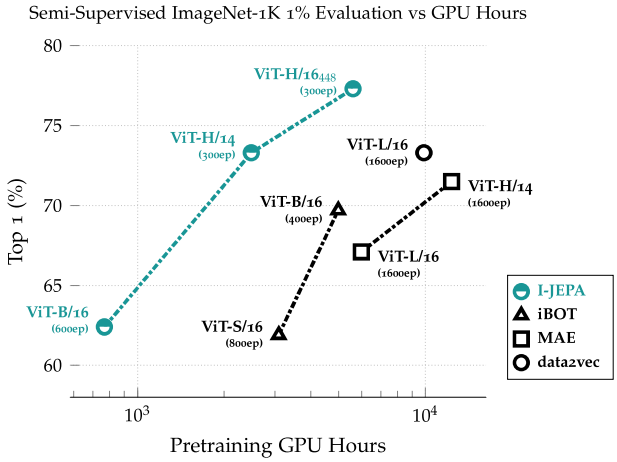


Figure 5. **Scaling**. Semi-supervised evaluation on ImageNet-1K 1% as a function of pretraining GPU hours. I-JEPA requires less compute than previous methods to achieve strong performance. Compared to MAE and data2vec, I-JEPA obtains a significant speedup by requiring fewer pretraining epochs. Compared 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

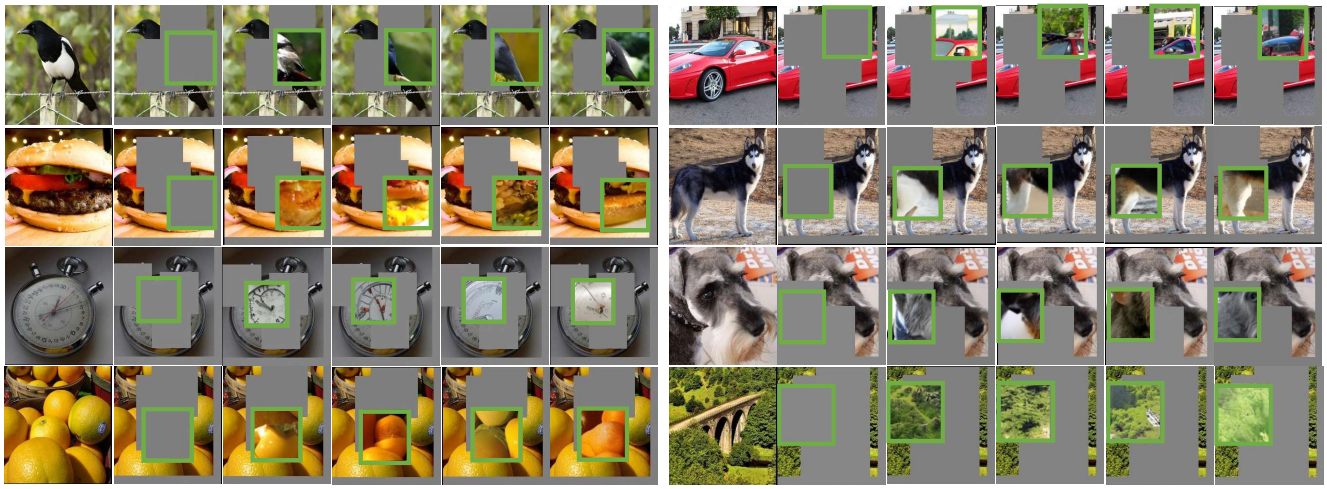


Figure 6. **Visualization of I-JEPA predictor representations**. For each image: first column contains the original image; second column contains the context image, which is processed by a pretrained I-JEPA ViT-H/14 encoder. Green bounding boxes in subsequent columns contain samples from a generative 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.