Joint Embedding Predictive Architecture (JEPA) and extensions for different modalities

Gihan Jayatilaka, Siddhi Patil

Department of Computer Science, University of Maryland

CMSC848U, 2026 Spring, 2026 April 8

Please note that this PDF does not contain the JEPA extensions for different modalities section from Siddhi. Feel free to reach out to her if you are interested.

Outline

Outline

This talk will go through the following ideas.

1. Representation learning
2. JEPA [Assran et al., 2023] in the context of prior representation learning work
3. JEPA Extensions for Images, Audio, Video, Robotics

Few remarks

- We assume the class is generally aware of standard machine learning and deep learning terminology.
- We start with image modality for representation learning because it has been the driving force in literature.

Outline

This talk will go through the following ideas.

1. Representation learning
2. JEPA [Assran et al., 2023] in the context of prior representation learning work
3. JEPA Extensions for Images, Audio, Video, Robotics

Few remarks

- We assume the class is generally aware of standard machine learning and deep learning terminology.
- We start with image modality for representation learning because it has been the driving force in literature.

Outline

This talk will go through the following ideas.

1. Representation learning
2. JEPA [Assran et al., 2023] in the context of prior representation learning work
3. JEPA Extensions for Images, Audio, Video, Robotics

Few remarks

- We assume the class is generally aware of standard machine learning and deep learning terminology.
- We start with image modality for representation learning because it has been the driving force in literature.

Representation learning

Here, we try to learn a representation $z \in \mathbb{R}^k$ for every datapoint x by learning a function f such that,

$$f_\theta : x \rightarrow z$$

. For almost every technique, this function is parameterized by θ. θ is learn by minimizing a loss function over the full dataset.

Major schools of thought for different representation learning techniques:

- Supervised representation learning
- Unsupervised / Semi Supervised representation learning
  - Invariance based
  - Reconstruction based
  - Clustering based
  - Joint Embedding Predictive Architecture

Representation learning

Here, we try to learn a representation $z \in \mathbb{R}^k$ for every datapoint x by learning a function f such that,

$$f_\theta : x \rightarrow z$$

. For almost every technique, this function is parameterized by θ. θ is learn by minimizing a loss function over the full dataset.

Major schools of thought for different representation learning techniques:

- Supervised representation learning
- Unsupervised / Semi Supervised representation learning
  - Invariance based
  - Reconstruction based
  - Clustering based
  - Joint Embedding Predictive Architecture

Representation learning

Here, we try to learn a representation $z \in \mathbb{R}^k$ for every datapoint x by learning a function f such that,

$$f_\theta : x \to z$$

. For almost every technique, this function is parameterized by θ. θ is learn by minimizing a loss function over the full dataset.

Major schools of thought for different representation learning techniques:

- Supervised representation learning
- Unsupervised / Semi Supervised representation learning
  - Invariance based
  - Reconstruction based
  - Clustering based
  - Joint Embedding Predictive Architecture

Representation learning

Here, we try to learn a representation $z \in \mathbb{R}^k$ for every datapoint x by learning a function f such that,

$$f_\theta : x \rightarrow z$$

. For almost every technique, this function is parameterized by θ. θ is learn by minimizing a loss function over the full dataset.

Major schools of thought for different representation learning techniques:

- Supervised representation learning
- Unsupervised / Semi Supervised representation learning
  - Invariance based
  - Reconstruction based
  - Clustering based
  - Joint Embedding Predictive Architecture

Notation

- D is used as the dataset of images for every slide.
- ℓ is used as a distance metric.

Abuses of notation:

- ℓ is used as a distance metric (similarity materic) between images $\ell(x_1, x_2); x_i \in D$ or between representations $\ell(z_1, z_2); z_i = f_\theta(x_i)$. Please infer it from context.
- Some functions (g_ψ, $f_{r\phi}$), eventhough parametric, are witten without the parameter (g, f_r).

Notation

- D is used as the dataset of images for every slide.
- ℓ is used as a distance metric.

Abuses of notation:

- ℓ is used as a distance metric (similarity materic) between images $\ell(x_1, x_2); x_i \in D$ or between representations $\ell(z_1, z_2); z_i = f_\theta(x_i)$. Please infer it from context.
- Some functions (g_ψ, $f_{r\phi}$), eventhough parametric, are witten without the parameter (g, f_r).

Supervised representation learning

Intuition: The best representation would be the best way to represent datapoints for a downstream task. Start with a dataset

$D = (X, Y)$ with datapoints $(x, y) \in D$. Consider a task head g in addition to the f.

$$z = f_\theta(x) \qquad \hat{y} = g(z)$$

$$\theta = \mathit{argmin} \left(\sum_{x,y \in D} \ell(y, \hat{y}) \right)$$

Supervised representation learning

Intuition: The best representation would be the best way to represent datapoints for a downstream task. Start with a dataset $D = (X, Y)$ with datapoints $(x, y) \in D$. Consider a task head g in addition to the f.

$$z = f_\theta(x) \qquad \hat{y} = g(z)$$

$$\theta = \textit{argmin} \left(\sum_{x,y \in D} \ell(y, \hat{y}) \right)$$

Supervised representation learning

Intuition: The best representation would be the best way to represent datapoints for a downstream task. Start with a dataset

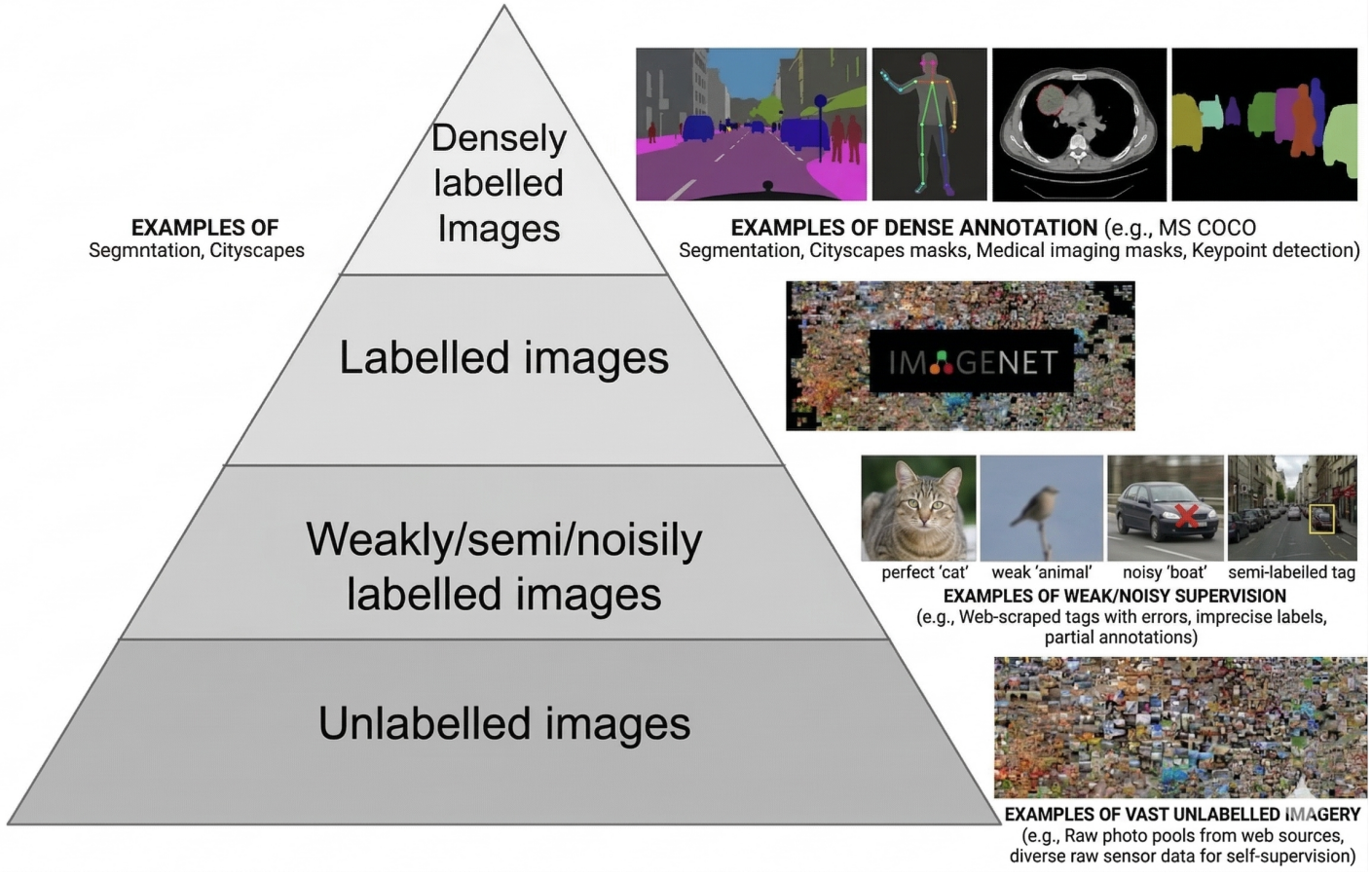
$D = (X, Y)$ with datapoints $(x, y) \in D$. Consider a task head g in addition to the f.

$$z = f_\theta(x) \qquad \hat{y} = g(z)$$

$$\theta = \text{argmin} \left(\sum_{x,y \in D} \ell(y, \hat{y}) \right)$$

$$\hat{y} = g(f_\theta(x))$$

Data pyramid



Invariance based representation learning

Intuition: The representation of an image should be invariant under certain changes to the image. Start with a dataset $x \in D$. Let $h^i : x \to x^i$ be a function that would make changes to the image appearance. Let H be a set of such functions.

$$\theta = argmin \left(\sum_{x \in D} \sum_{z_i} \ell(z_0, z_i) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \sum_{h \in H} \ell(f_\theta(x), f_\theta(h(x))) \right)$$

Representation collapse?

Invariance based representation learning

Intuition: The representation of an image should be invariant under certain changes to the image. Start with a dataset $x \in D$. Let $h^i : x \rightarrow x^i$ be a function that would make changes to the image appearance. Let H be a set of such functions.

$$\theta = argmin \left(\sum_{x \in D} \sum_{z_i} \ell(z_0, z_i) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \sum_{h \in H} \ell(f_\theta(x), f_\theta(h(x))) \right)$$

Representation collapse?

Invariance based representation learning

Intuition: The representation of an image should be invariant under certain changes to the image. Start with a dataset $x \in D$. Let $h^i : x \rightarrow x^i$ be a function that would make changes to the image appearance. Let H be a set of such functions.

$$\theta = argmin \left(\sum_{x \in D} \sum_{z_i} \ell(z_0, z_i) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \sum_{h \in H} \ell(f_\theta(x), f_\theta(h(x))) \right)$$

Representation collapse?

Invariance based representation learning

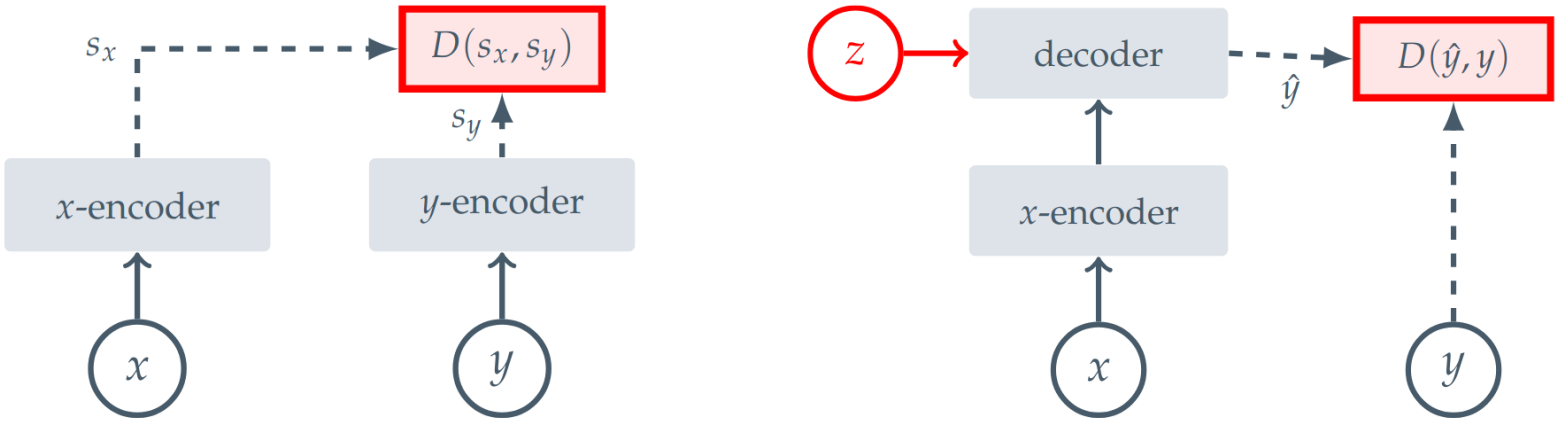
Intuition: The representation of an image should be invariant under certain changes to the image. Start with a dataset $x \in D$. Let $h^i : x \rightarrow x^i$ be a function that would make changes to the image appearance. Let H be a set of such functions.

$$\theta = argmin \left(\sum_{x \in D} \sum_{z_i} \ell(z_0, z_i) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \sum_{h \in H} \ell(f_\theta(x), f_\theta(h(x))) \right)$$

Representation collapse?

Invariance based representation learning



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

Figure 1:

Reconstruction based representation learning

Intuition: Given a good part of an image, the masked/corrupted part of an image could be reconstructed.

Start with a dataset $x \in D$. Let $f_m : x \rightarrow \bar{x}$ be a function that would mask a portion of x. Let $f_r : \bar{z} \rightarrow \hat{x}$ be a function that would reconstruct the image.

$$\theta = argmin \left(\sum_{x \in D} \ell(x, \hat{x}) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

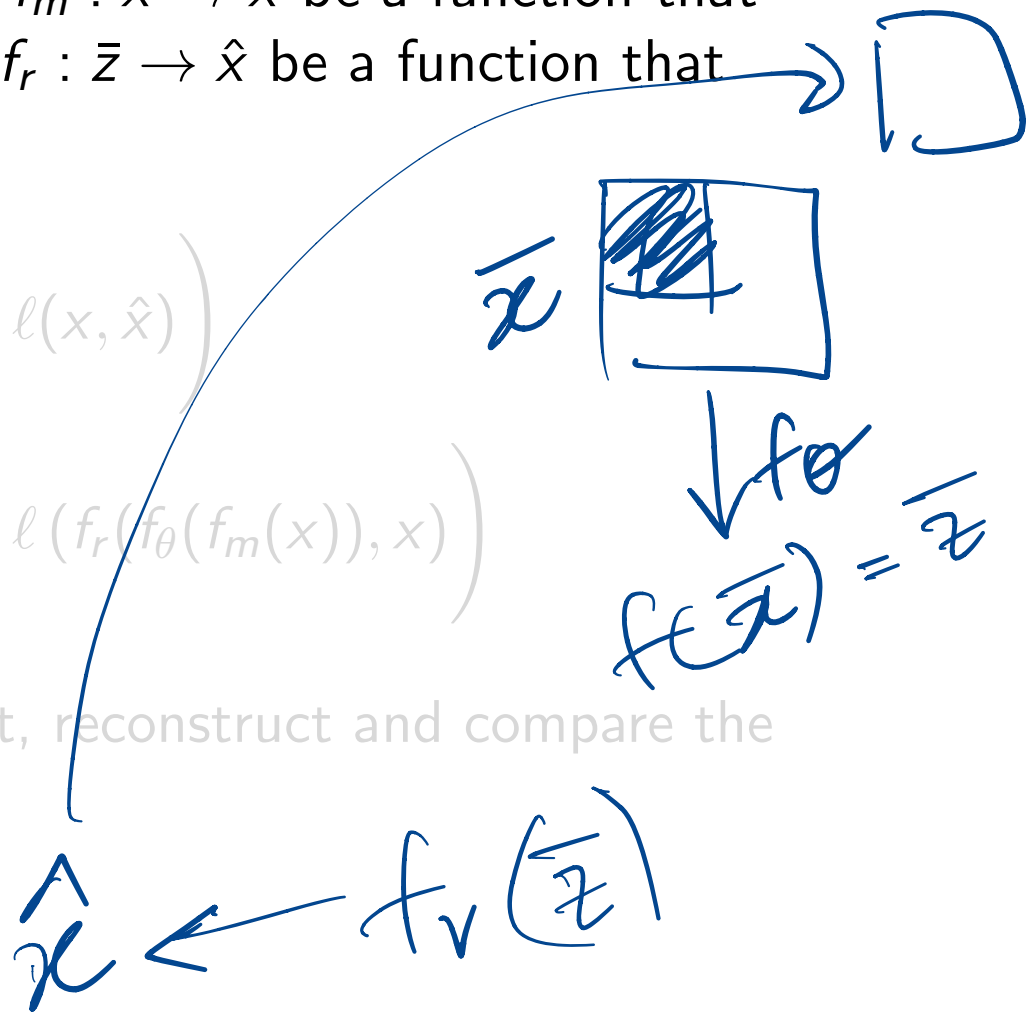
Reconstruction based representation learning

Intuition: Given a good part of an image, the masked/corrupted part of an image could be reconstructed.
Start with a dataset $x \in D$. Let $f_m : x \rightarrow \bar{x}$ be a function that would mask a portion of x. Let $f_r : \bar{z} \rightarrow \hat{x}$ be a function that would reconstruct the image.

$$\theta = argmin \left(\sum_{x \in D} \ell(x, \hat{x}) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \ell\left(f_r(f_\theta(f_m(x)), x\right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.



Reconstruction based representation learning

Intuition: Given a good part of an image, the masked/corrupted part of an image could be reconstructed.
Start with a dataset $x \in D$. Let $f_m : x \rightarrow \bar{x}$ be a function that would mask a portion of x. Let $f_r : \bar{z} \rightarrow \hat{x}$ be a function that would reconstruct the image.

$$\theta = argmin \left(\sum_{x \in D} \ell(x, \hat{x}) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Reconstruction based representation learning

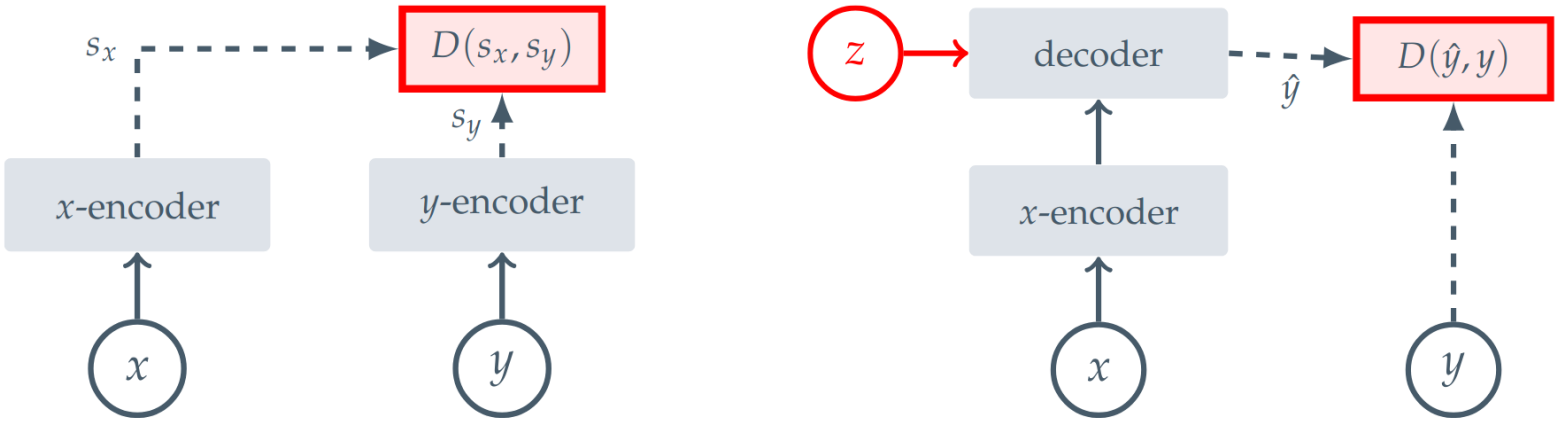
Intuition: Given a good part of an image, the masked/corrupted part of an image could be reconstructed.
Start with a dataset $x \in D$. Let $f_m : x \rightarrow \bar{x}$ be a function that would mask a portion of x. Let $f_r : \bar{z} \rightarrow \hat{x}$ be a function that would reconstruct the image.

$$\theta = argmin \left(\sum_{x \in D} \ell(x, \hat{x}) \right)$$

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Reconstruction based representation learning



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

Figure 2:

Clustering based representation learning

Intuition: Images of a similar group (cluster) should have similar representation. Start with a dataset $x \in D$. For every x, there is cluster information D_x^+ which is the cluster of data points in the same cluster as x, and $D_x^- = D - D_x^+$.

$$\theta = argmin \left(\mathbb{E}_{y \in D_x^+} [\ell(f_\theta(y), f_\theta(x))] \right)$$

$$AND$$

$$argmax \left(\mathbb{E}_{y \in D_x^-} [\ell(f_\theta(y), f_\theta(x))] \right)$$

Clustering based representation learning

Intuition: Images of a similar group (cluster) should have similar representation. Start with a dataset $x \in D$. For every x, there is

cluster information D_x^+ which is the cluster of data points in the same cluster as x, and $D_x^- = D - D_x^+$.

$$\theta = argmin \left(\mathbb{E}_{y \in D_x^+} [\ell(f_\theta(y), f_\theta(x))] \right)$$

$$AND$$

$$argmax \left(\mathbb{E}_{y \in D_x^-} [\ell(f_\theta(y), f_\theta(x))] \right)$$

Clustering based representation learning

Intuition: Images of a similar group (cluster) should have similar representation. Start with a dataset $x \in D$. For every x, there is

cluster information D_x^+ which is the cluster of data points in the same cluster as x, and $D_x^- = D - D_x^+$.

$$\theta = \mathit{argmin} \left(\mathbb{E}_{y \in D_x^+} [\ell(f_\theta(y), f_\theta(x))] \right)$$

$$\mathit{AND}$$

$$\mathit{argmax} \left(\mathbb{E}_{y \in D_x^-} [\ell(f_\theta(y), f_\theta(x))] \right)$$

What would we dislike in a representaion learning algorithm?

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Image reconstruction.
- Unreliable labels (clustering).

What would we like in a representation learning algorithm?

- ℓ, f_θ
- Masking seems more natural than augmentation.

Revisit reconstrction based representation learning with likes and dislikes

- ℓ, f_θ
- Masking seems more natural than augmentation.

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Full image reconstruction.
- Unreliable labels (clustering).

Reconstruction based representation learning:

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Revisit reconstrction based representation learning with likes and dislikes

- ℓ, f_θ
- Masking seems more natural than augmentation.

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Full image reconstruction.
- Unreliable labels (clustering).

Reconstruction based representation learning:

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Revisit reconstrction based representation learning with likes and dislikes

- ℓ, f_θ
- Masking seems more natural than augmentation.

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Full image reconstruction.
- Unreliable labels (clustering).

Reconstruction based representation learning:

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Revisit reconstrction based representation learning with likes and dislikes

- ℓ, f_θ
- Masking seems more natural than augmentation.

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Full image reconstruction.
- Unreliable labels (clustering).

Reconstruction based representation learning:

$$\theta = \mathit{argmin} \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

Here, we mask, embed/represent, reconstruct and compare the result to the original image.

Revisit reconstrction based representation learning with likes and dislikes

- ℓ, f_θ
- Masking seems more natural than augmentation.

- Supervisory labels.
- Preset augmentations.
- Representation collapse.
- Full image reconstruction.
- Unreliable labels (clustering).

Reconstruction based representation learning:

$$\theta = argmin \left(\sum_{x \in D} \ell \left(f_r(f_\theta(f_m(x)), x \right) \right)$$

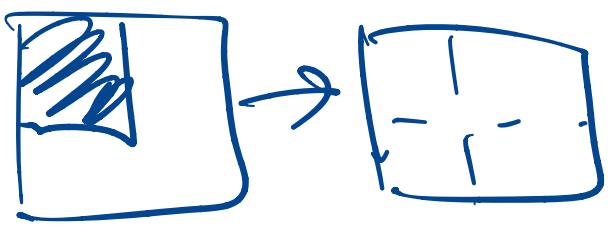
Here, we mask, embed/represent, reconstruct and compare the result to the original image.

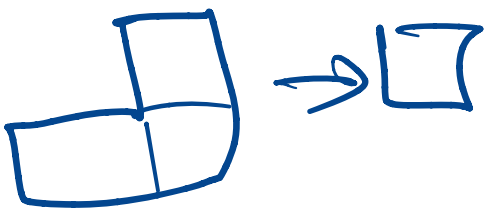
A representaion learning work we like!

(Optimization problem)

$$\theta = \arg\min \sum_{x \in D} l\left(f_r\left(f_\theta\left(f_m(x)\right)\right), x\right)$$

$$l\left(f_r\left(f_\theta\left(f_m(x)\right)\right), f_\theta(x)\right)$$

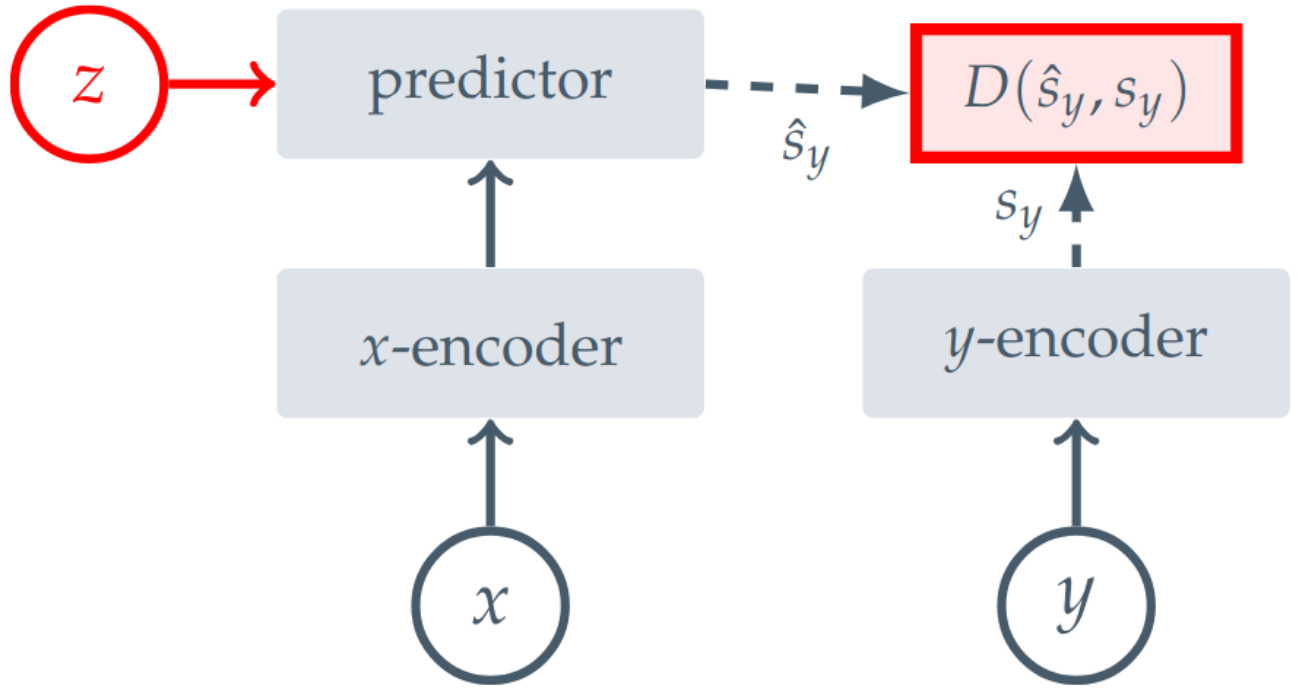




$$f_l\left(f_\theta\left(f_m(x)\right), f_\theta(x)\right)$$

What I made
What I want

I-JEPA Architecture

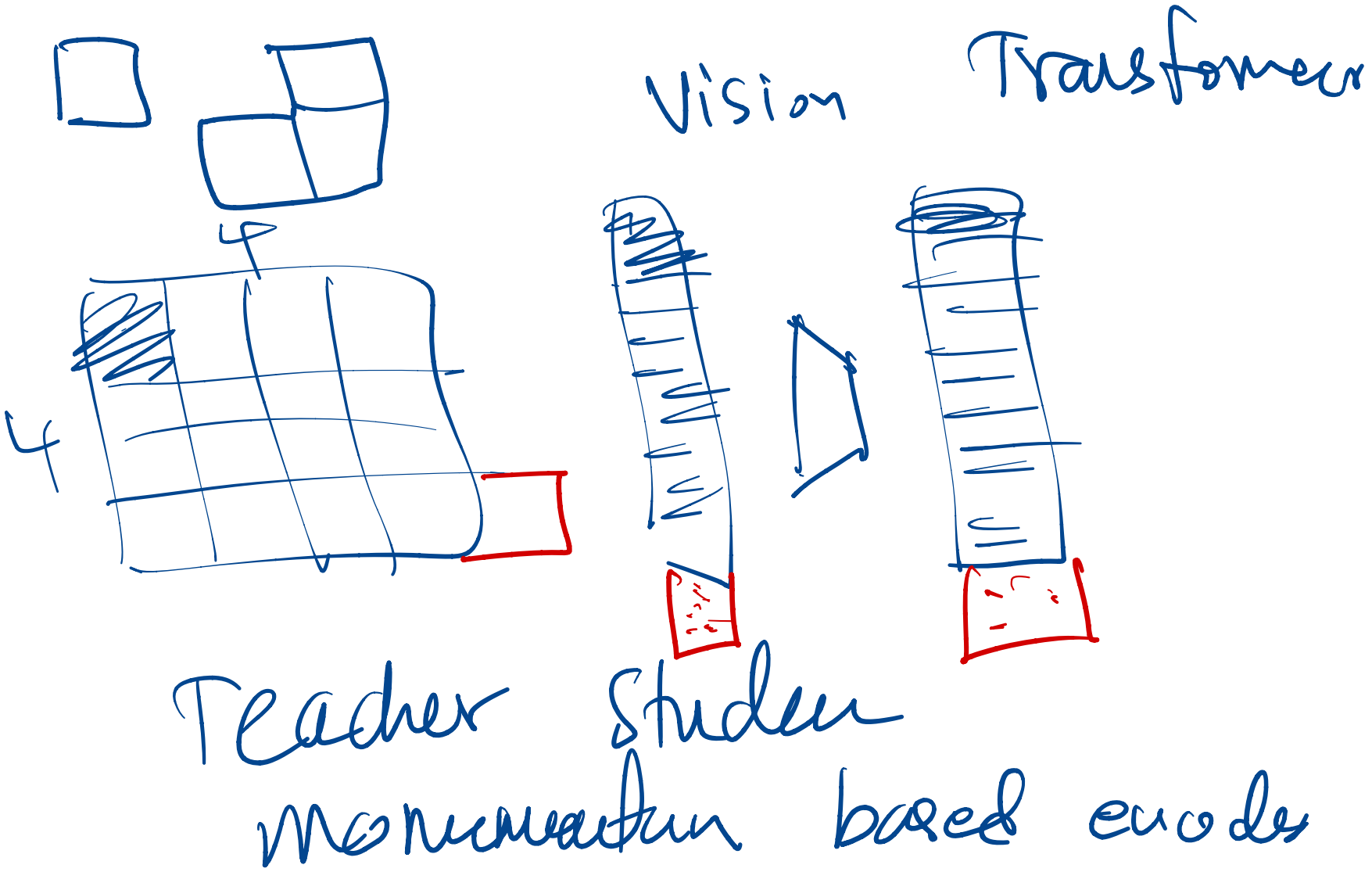


(c) **Joint-Embedding Predictive Architecture**

Figure 3:

A representaion learning framework we like!

(Architecture decisions and Representation collapse)



I-JEPA Architecture

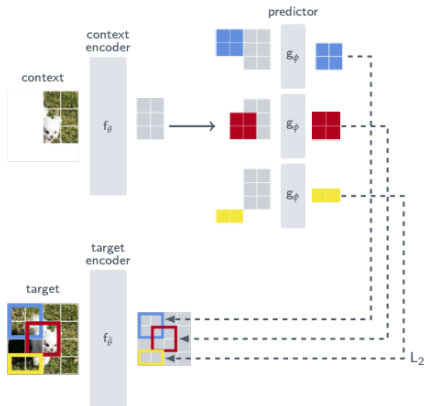
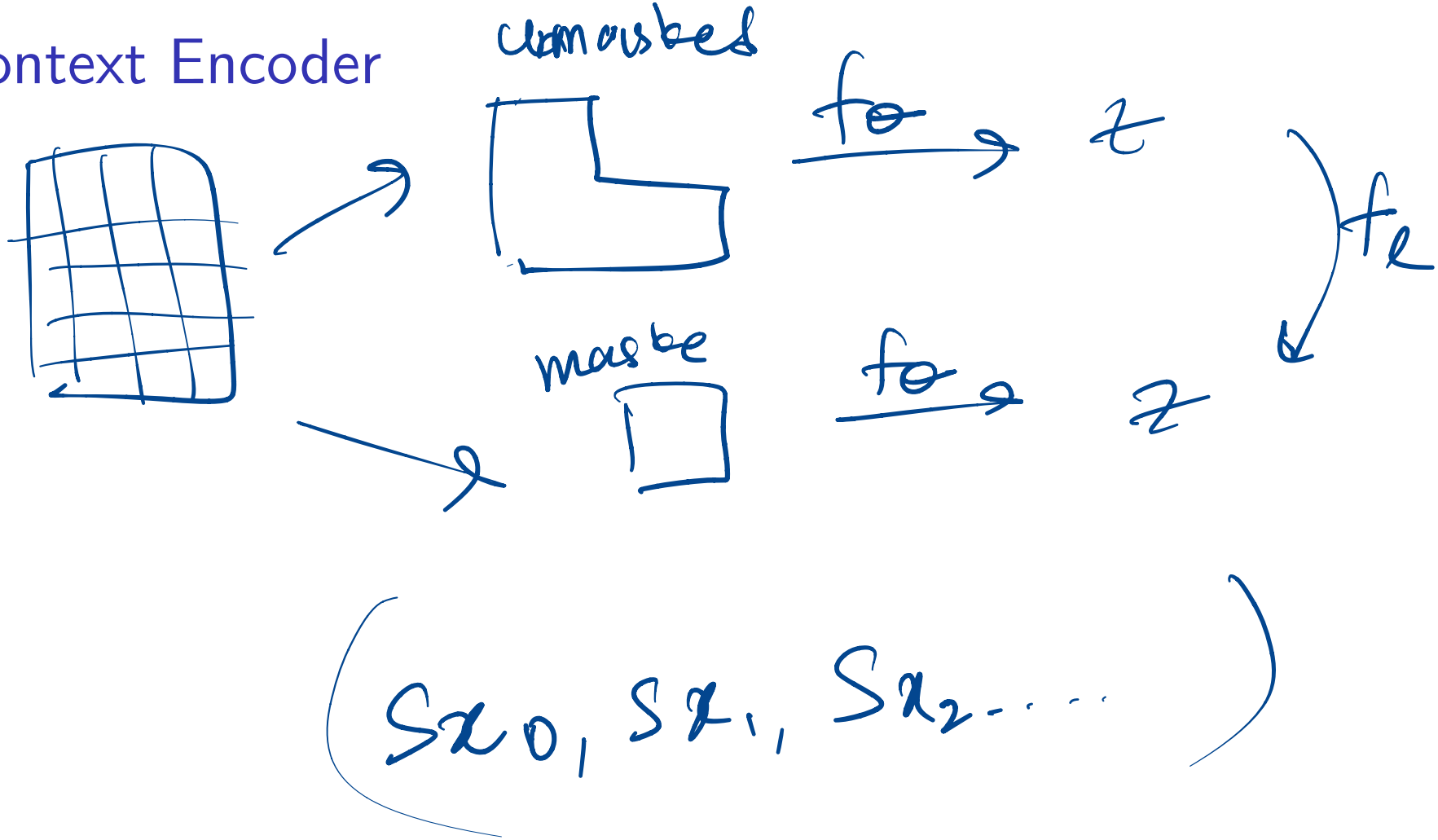


Figure 4: I-JEPA Architecture

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

Context Encoder

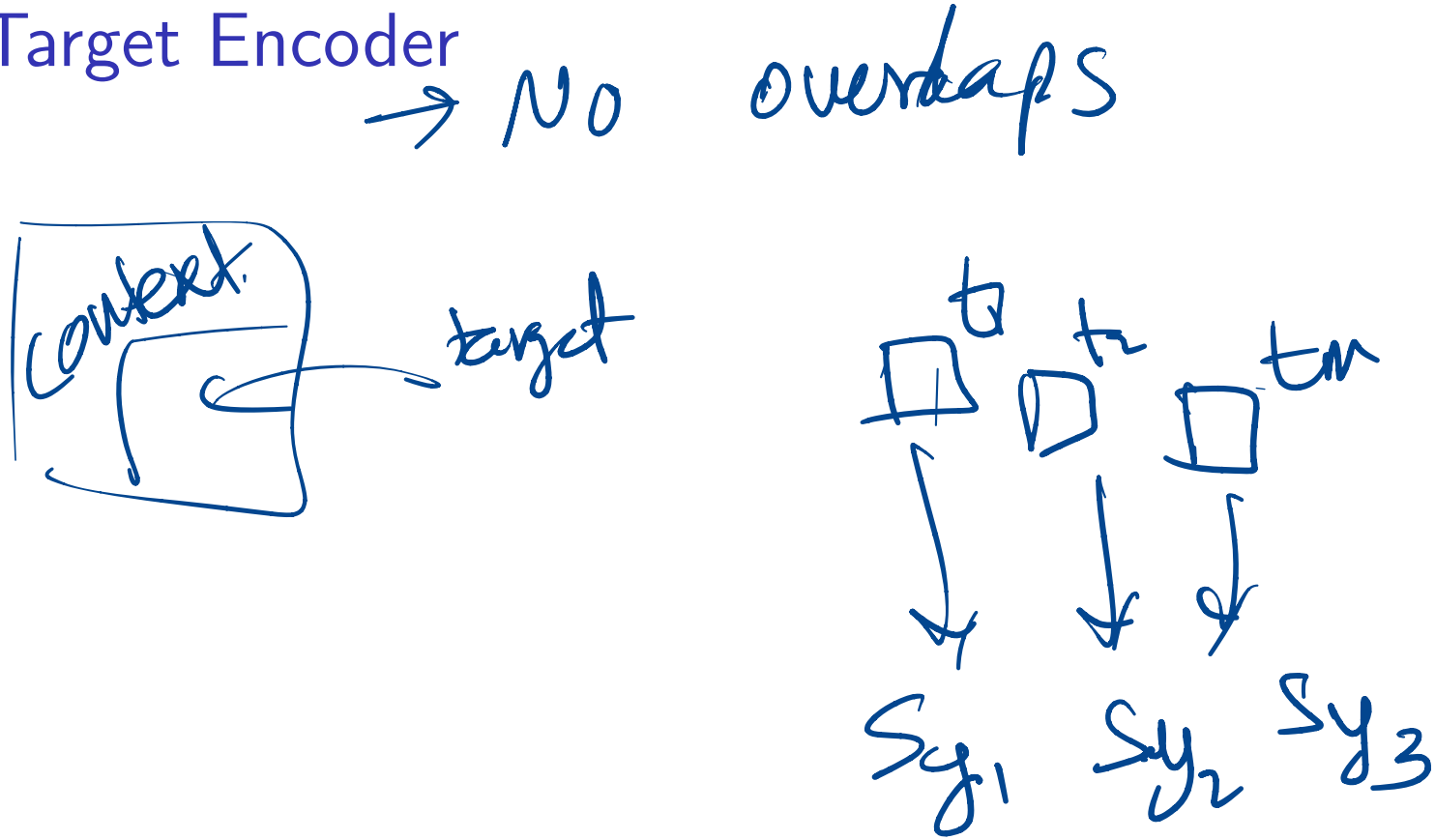


To obtain the context, a single block x with scale $(0.85, 1.0)$ and mask B_x is processed by the context encoder f_θ to produce the representation $s_x = \{s_{x_j}\}_{j \in B_x}$.

Context Encoder

To obtain the context, a single block x with scale $(0.85, 1.0)$ and mask B_x is processed by the context encoder f_θ to produce the representation $s_x = \{s_{x_j}\}_{j \in B_x}$.

Target Encoder

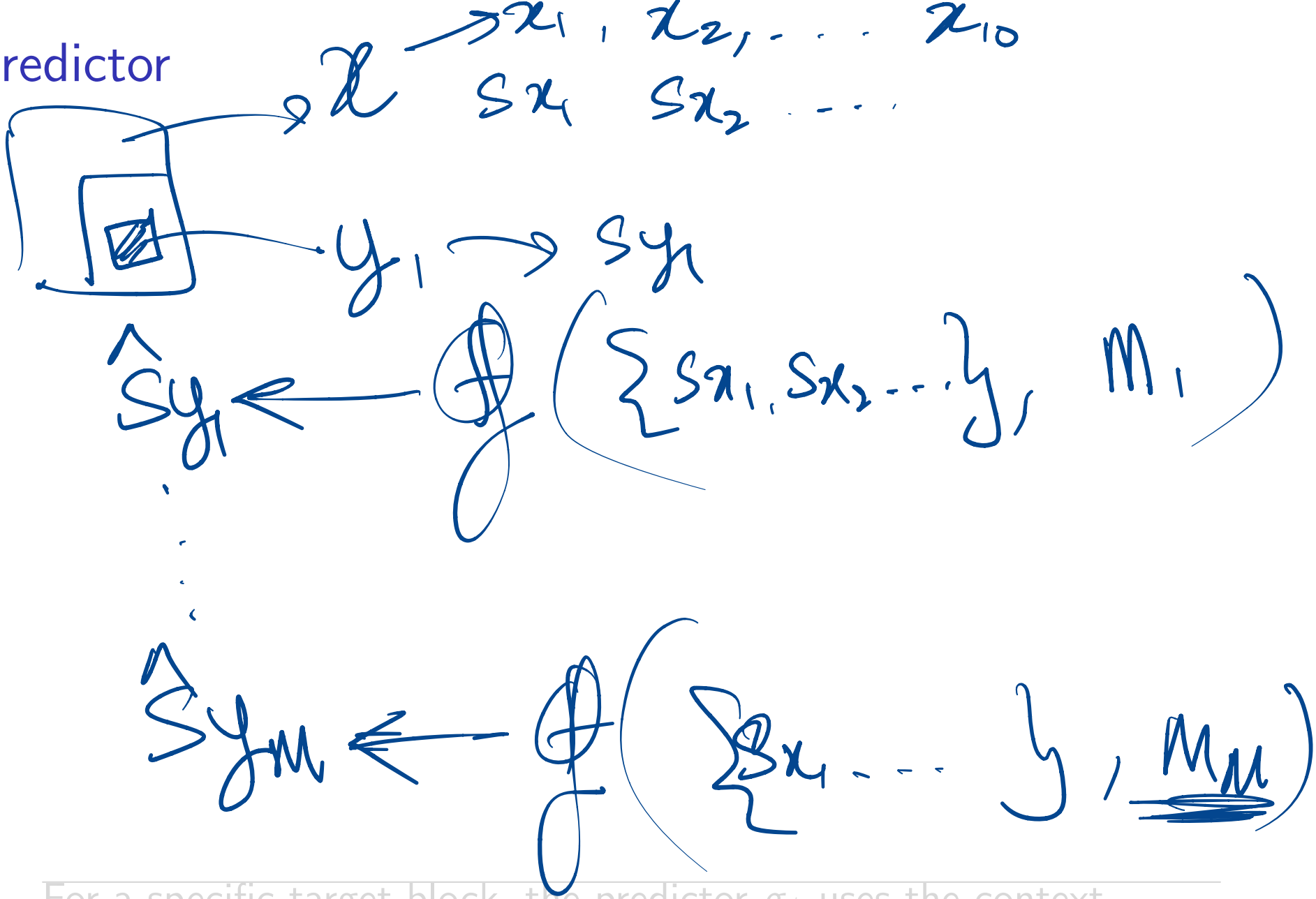


To produce targets in the I-JEPA framework, an input image y is converted into N non-overlapping patches and passed through a target-encoder $f_{\bar{\theta}}$ to obtain patch-level representations $s_y = \{s_{y_1}, \ldots, s_{y_N}\}$, from which M blocks are sampled such that the i^{th} block's representation is defined by the mask B_i as $s_y(i) = \{s_{y_j}\}_{j \in B_i}$.

Target Encoder

To produce targets in the I-JEPA framework, an input image y is converted into N non-overlapping patches and passed through a target-encoder $f_{\bar{\theta}}$ to obtain patch-level representations $s_y = \{s_{y_1}, \ldots, s_{y_N}\}$, from which M blocks are sampled such that the i^{th} block's representation is defined by the mask B_i as $s_y(i) = \{s_{y_j}\}_{j \in B_i}$.

Predictor



For a specific target block, the predictor g_ϕ uses the context representation s_x and mask tokens $\{m_j\}$ to generate the patch-level prediction $\hat{s}_y = \{\hat{s}_{y_j}\} = g_\phi(s_x, \{m_j\})$.

Predictor

For a specific target block, the predictor g_ϕ uses the context representation s_x and mask tokens $\{m_j\}$ to generate the patch-level prediction $\hat{s}_y = \{\hat{s}_{y_j}\} = g_\phi(s_x, \{m_j\})$.

Putting them all together

$$x \to x_1 \cdots x_n \xrightarrow{f_a} s_{x_i}$$

$$y \to y_1 \cdots y_k \xrightarrow{f_a} s_{y_j}$$

$$\hat{s}_{y_j} = g(\{s_{x_i}\}, m_j)$$

$$\sum_{x \in D} \sum_{i=1}^{M} l(\hat{s}_{y_i}, s_{y_i})$$

JEPA Loss

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.

JEPA Main Results (Image Classification)

Method	Arch.	Epochs	Top-1
Methods without view data augmentations			
data2vec [7]	ViT-L/16	1600	77.3
MAE [35]	ViT-B/16	1600	68.0
	ViT-L/16	1600	76.0
	ViT-H/14	1600	77.2
CAE [21]	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 [20]	RN152 (2×)	800	79.1
DINO [17]	ViT-B/8	300	80.1
iBOT [75]	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 [7]	ViT-L/16	1600	73.3
MAE [35]	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 [75]	ViT-B/16	400	69.7
DINO [17]	ViT-B/8	300	70.0
SimCLR v2 [34]	RN151 (2×)	800	70.2
BYOL [34]	RN200 (2×)	800	71.2
MSN [3]	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.

JEPA Other Results

Method	Arch.	CIFAR100	Places205	iNat18
Methods without view data augmentations				
data2vec [7]	ViT-L/16	81.6	54.6	28.1
MAE [35]	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 [17]	ViT-B/8	84.9	57.9	55.9
iBOT [75]	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.

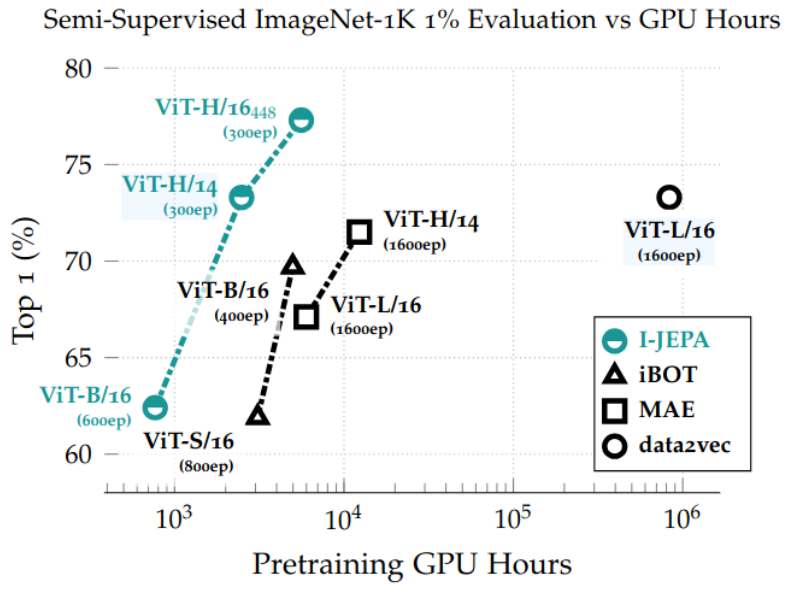


Figure 5. **Scaling**. Semi-supervised evaluation on ImageNet-1K 1% as a function of pretraining GPU hours. I-JEPA requires less compute than previous methods to achieve strong performance. Compared to MAE and data2vec, I-JEPA obtains a significant speedup by requiring fewer pretraining epochs. Compared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

JEPA Other Results

Method	Arch.	CIFAR100	Places205	iNat18
Methods without view data augmentations				
data2vec [7]	ViT-L/16	81.6	54.6	28.1
MAE [35]	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 [17]	ViT-B/8	84.9	57.9	55.9
iBOT [75]	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.

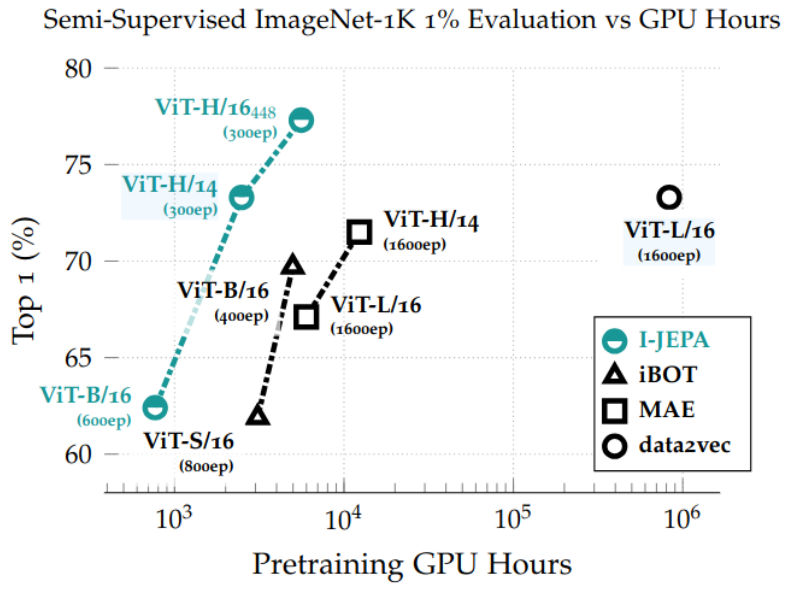


Figure 5. **Scaling**. Semi-supervised evaluation on ImageNet-1K 1% as a function of pretraining GPU hours. I-JEPA requires less compute than previous methods to achieve strong performance. Compared to MAE and data2vec, I-JEPA obtains a significant speedup by requiring fewer pretraining epochs. Compared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

References I

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. (2023).

Self-supervised learning from images with a joint-embedding predictive architecture.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629.