Enhancing JEPAs with Spatial Conditioning: Robust and Efficient Representation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Image-based Joint-Embedding Predictive Architecture (IJEPA) offers an attractive alternative to Masked Autoencoder (MAE) for representation learning using the Masked Image Modeling framework. IJEPA drives representations to capture useful semantic information by predicting in latent rather than input space. However, IJEPA relies on carefully designed context and target windows to avoid representational collapse. The encoder modules in IJEPA cannot adaptively modulate the type of predicted and/or target features based on the feasibility of the masked prediction task as they are not given sufficient information of both context and targets. Based on the intuition that in natural images, information has a strong spatial bias with spatially local regions being highly predictive of one another compared to distant ones. We condition the target encoder and context encoder modules in IJEPA with positions of context and target windows respectively. Our "conditional" encoders show performance gains on several image classification benchmark datasets, improved robustness to context window size and sample-efficiency during pretraining.

1 Introduction

Masked Image Modeling (MIM) offers a scalable framework to learn representations from unlabelled data in a self-supervised manner by learning to predict masked regions given unmasked ones as context [1–8]. A distinction can be drawn for models under this framework based on whether the targets are predicted in input space (pixels, words, sounds etc.) by MAEs [4] or in latent space by JEPAs [8, 9]. Recently, Littwin et al. [10] suggest that JEPAs have an implicit bias for learning "high-influence" features compared to Masked Autoencoders (MAEs) which could explain their empirical success compared to MAEs. However, JEPAs require careful selection of context and target windows (window size and distance of separation) to drive the representations to capture useful information (semantics) from input images for a variety of high-level downstream tasks like image classification as well as fine-grained tasks like object counting and depth prediction. Sub-optimal choice of context and target windows, i.e. pairs with low mutual information, potentially leads to representational collapse. Our work attempts to alleviate these limitations in JEPAs [8, 9] — improve representational quality to solve downstream tasks and robustness to masking hyperparameters for pretraining.

In natural images, it is intuitive to expect nearby regions to be highly predictive of one another (high mutual information) compared to distant ones. The feasibility of the masked prediction task in JEPAs is linked to the mutual information between context and target windows. Consider the scene in Figure 1 of the dog in the backyard, patches of grass co-occur with patches of flower pots but its plausible in other scenes for grass patches to co-occur with patches of sky, trees, water etc. Therefore grass alone is not a highly predictive contextual feature for flower pots. On the other hand, patches from the same object (eg. dog), are highly predictive of each other as they co-occur almost always. Good choices for context and target masks in MIM require a careful balance of the amount of mutual information between image regions in the context and target windows. When the mutual

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

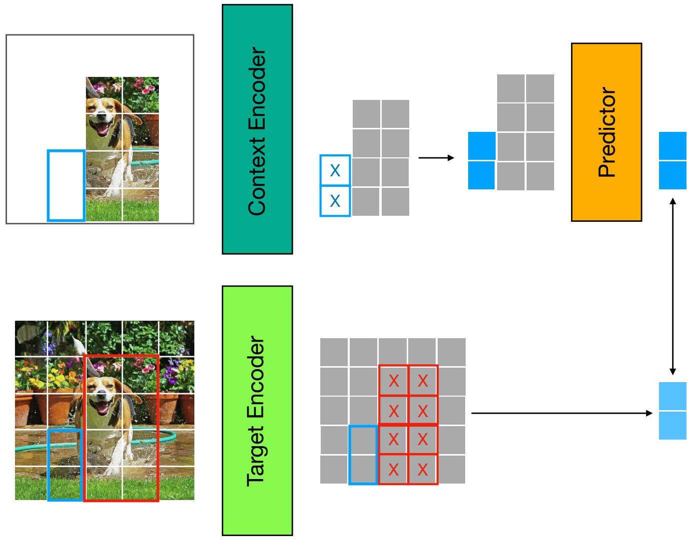

Figure 1: Conditioning the Context and Target Encoders in IJEPA with positions of the target (blue box) and context windows (red box) respectively. Patches marked with X indicate positional information while those with solid color fill indicate feature information is extracted at those locations.

information between image regions in the context and target windows is too low the prediction task is very challenging. This forces the encoders to extract only the most feasible set of features to predict from the target given a context leading to representational collapse in the limiting case. While if the mutual information is too high it becomes rather trivial resulting in the representations not capturing sufficiently abstract information from the input image.

In JEPAs (eg. IJEPA [8]), the context and target encoders are given insufficient information about the prediction task as they do not have access to both context and target windows. Therefore, the target encoder module cannot adaptively modulate target features (feedback signal) based on the feasibility of prediction to the context encoder. Without providing the context encoder and target encoder modules sufficient information of the masked prediction task, they can only extract highly predictable features from the context and target windows which could lead to representational collapse. Since predictability of information in natural images has a strong spatial bias as outlined above, providing information of sizes of context and target windows and the distance of separation could alleviate this issue.

We incorporate this intuition in IJEPA [8] by conditioning the context encoder with positions of the target window and conversely the target encoder with positions of the context window. Given this additional information of spatial locations of target patches allows the context encoder to modulate the type of features to capture (low-level features $\rightarrow$ color, texture, shape or higher-level features $\rightarrow$ object categories) from the input image. Conversely, the target encoder can use the positional information of the context window to adaptively modulate the type of target features that are feasible to predict for the context encoder module. Our proposed conditioning allows the context and target encoders to adapt the set of predictive features based on the size of context or target windows and/or their distance of separation. Such "conditional" encoders, we term **E**ncoder **C**onditioned JEPAs (EC-JEPAs), when used as a drop-in replacement in IJEPA [8] lead to — i) improved representational quality measured by rank-based metrics (*LiDAR* [11] and *RankMe* [12]) as well as classification performance on benchmark datasets such as ImageNet [13] (see Table 1), out-of-distribution datasets such as CIFAR10, CIFAR100, Food101 etc. ii) improved robustness to context window hyperparameters during pretraining (see Figure 2) crucial to prevent representational collapse during pretraining iii) improved sample-efficiency in pretraining measured by classification performance on ImageNet [13] (see Figure 3).

2 Method

We first review the IJEPA model [14] followed by our proposed modification to the same.

IJEPA Let $x \in \mathbb{R}^{T \times d}$ and $p \in \mathbb{R}^{T \times d}$ denote the tokenized input image and position embeddings respectively, where T is the number of tokens, and d the token dimension (we assume position embeddings p are added to the image tokens to produce x). Let c denote a set of indices corresponding to the context tokens, such that $x_c = \{x_j\}_{j \in c}$. Likewise, let $t^1, ..., t^k$ denote k sets of indices with cardinality $m = |t_1| = |t_2| = ...|t_k|$ corresponding to the target token blocks (we use $k = 4$ in our experiments following IJEPA [14]). In the IJEPA formulation, an encoder function encodes the context tokens into latent representations $z_c = f(x_c; \theta)$ where θ are the encoder weights, which are then used to predict the target representations $z_{t^j} = f(x; \tilde{\theta})_{t^j}$ for $j = \{1, ..., k\}$, where $\tilde{\theta}$ are an exponential moving average of the weights θ, with the aid of a predictor function g. The predictor function takes as input the context representations z_c, the target positions p_{t^j}, and predicts the targets representations $\hat{z}_{t_j} = g(z_c, p_{t^j}; \psi)$ for $j = \{1, ..., k\}$ where ψ are the predictor weights.

EC-IJEPA In our approach, we use the context and target positions to condition the encoders for pretraining. Namely $z_c^{t^1,...,t^k} = f(x_c, p_{t^1}, ..., p_{t^k}; \theta)$, and similarly $z_{t^j}^c = f(x, p_c; \tilde{\theta})_{t^j}$ for $j = \{1, ..., k\}$. At inference, we simply condition the encoder on all position embeddings p. In practice, the functions f and g are instantiated as Vision Transformers (ViTs) [15], and are conditioned by appending the positions as additional tokens in the input sequence processed by the Transformer modules. This increase in sequence length however, could incur a non-negligible cost in memory and compute resources, especially during inference which now processes twice as many tokens as the baseline IJEPA. To reduce this computational and memory overhead, we introduce an aggregation step prior to conditioning. At both training and inference, we first reduce the conditioning position tokens to a smaller set, which are used as the conditioning tokens instead of the full sequence. Concretely, we use 1D average pooling on $p_c, p_{t_1}, ..., p_{t_k}$ with a kernel and step size of $m//2$[1]. During inference, we use 2D average pooling on all positions p with a kernel and stride size of $[4, 4]$. This incurs an additional $T//16$ tokens to be processed at inference. Finally, we note that we use 1D, rather than 2D average pooling in training due to efficiency and implementation considerations, resulting in approximately 3% increase in FLOPs for training.

3 Results

We evaluate the baseline IJEPA and our proposed encoder conditioned variant EC-IJEPA on several visual benchmarks consistent with prior work [14, 16]. We follow the setup from Assran et al. [14] to pretrain the baseline IJEPA and our proposed EC-IJEPA on the ImageNet-1k (IN-1k) dataset [13] (see Appendix A for more details). The pretrained encoders are then used to extract representations, by average pooling the output sequence of patch-level tokens from the encoder. We evaluate these representations on various downstream benchmark datasets using the linear probing protocol adopted by prior work [14, 17] (see Appendix A for more details).

Table 1: Classification performance comparison on IN-1k dataset.

Model	Accuracy
IJEPA (ViT-L/16)	74.8
EC-IJEPA (ViT-L/16)	**76.7**
IJEPA (ViT-H/14)	77.4
EC-IJEPA (ViT-H/14)	**78.1**

Table 1 shows the performance of IJEPA and EC-IJEPA on the IN-1k classification benchmark. We see that EC-IJEPA outperforms the baseline IJEPA with different encoder sizes.

Prior works [11, 12] introduced metrics for measuring representational quality that correlate with downstream task performance without the need for a downstream task. *RankMe* [12], is one such metric that measures the soft effective rank of embeddings. *LiDAR* [11] is another that builds on *RankMe* by defining a surrogate task to estimate the effective rank of a Linear Discriminant Analysis matrix. Both *RankMe* and *LiDAR* metrics empirically show that they serve as useful proxies of representational quality. Higher scores of these metrics are positively correlated and serve as a necessary condition for

Table 2: RankMe and LiDAR scores for models pretrained on IN-1k. ViT-L/16 and ViT-H/14 encoders have embedding sizes 1024 and 1280 respectively.

Architecture	RankMe ↑	LiDAR ↑
IJEPA (ViT-L/16)	488.6	385.2
EC-IJEPA (ViT-L/16)	**533.0**	**486.5**
IJEPA (ViT-H/14)	540.8	437.2
EC-IJEPA (ViT-H/14)	**567.3**	**547.0**

[1]Note that the target cardinality m is sampled out of a range as in IJEPA

improved downstream performance for a given encoder architecture. We follow the setup from Garrido et al. [12] and Thilak et al. [11] including dataset size and construction to compute these metrics. Table 2 shows the *RankMe* and *LiDAR* metrics for IJEPA and EC-IJEPA pretrained on IN-1k. We see that EC-IJEPA shows higher scores for *RankMe* and *LiDAR* metrics compared to IJEPA which support the improvements in downstream task performance shown in Table 1.

Further, we measure the robustness of the baseline IJEPA and our variant EC-IJEPA to varying sizes for the context window. Figure 2 compares the classification scores of the baseline and our variant on IN-1k when pretrained for masked prediction task using a wider range of context window sizes using a ViT-L/16 encoder. We see that the quality of representations learned by the baseline IJEPA is very sensitive to this hyperparameter. In contrast, our variant EC-IJEPA is more robust to a wider range of context window sizes used for masking during pretraining. This suggests that our simple positional conditioning alleviates representational collapse in the encoders.

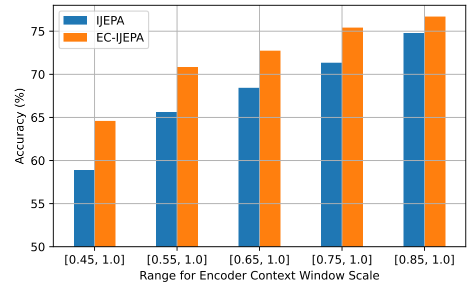

Figure 2: Ablation on ranges of context window scale used for pretraining.

Figure 3 shows the classification accuracy obtained by the baseline IJEPA and our variant EC-IJEPA on IN-1k over the pretraining cycle. We see that our EC-IJEPA is more sample-efficient for representation learning as it obtains consistently higher classification accuracy throughout the pretraining cycle.

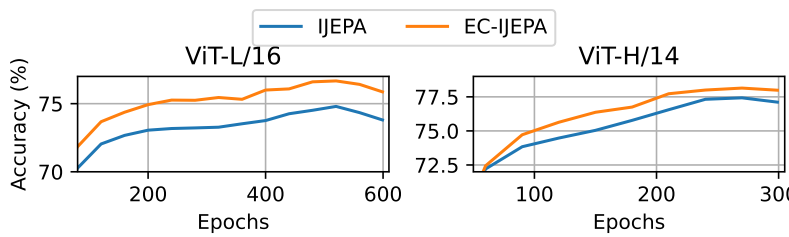

Figure 3: Classification performance on ImageNet-1k measured during pretraining cycle in IJEPA (blue) and EC-IJEPA (orange) at two encoder sizes (left: ViT-L/16 and right: ViT-H/14).

Table 3 shows the classification performance of IJEPA and EC-IJEPA on various out-of-distribution datasets such as CIFAR10, CIFAR100, EuroSat, Food101 and SUN397. We see that EC-IJEPA consistently outperforms IJEPA which highlights the superior representational quality of the former. Table 3 also compares performance of models on tasks which require local information such as object counting (CLEVR/Count) and depth prediction (CLEVR/Dist) [18, 19] where the two models are comparable with one exception (ViT-L/16 encoder on CLEVR/Dist).

Table 3: Classification performance on out-of-distribution datasets using two encoder sizes.

Model	CIFAR10	CIFAR100	EuroSat	Food101	SUN397	CLEVR/Count	CLEVR/Dist
IJEPA (ViT-L/16)	92.5	75.0	**96.7**	75.3	69.5	74.5	**65.3**
EC-IJEPA (ViT-L/16)	**93.4**	**76.7**	95.7	**76.5**	**71.2**	**75.2**	60.0
IJEPA (ViT-H/14)	94.5	78.9	**96.5**	78.4	71.5	79.3	64.8
EC-IJEPA (ViT-H/14)	**96.0**	**81.8**	96.0	**78.7**	**73.5**	79.4	64.6

4 Conclusion

Predictability of patch-level features in natural images has a strong spatial bias. We introduce a simple modification to the sequence of input tokens given to the encoder modules in JEPAs, we concatenate positions of target and context windows to the context and target encoders respectively. Using our "conditional" encoders as a drop-in replacement in IJEPA [14] shows improved representational quality for downstream image classification tasks and rank-based metrics (*RankMe* and *LiDAR*). Conditional encoders alleviate representational collapse across larger ranges of context window sizes and improve sample-efficiency during pretraining.

References

[1] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[2] 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9643–9653, 2021.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[5] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[6] 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. In *International Conference on Machine Learning*, 2022.

[7] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, 2022.

[8] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[9] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

[10] Etai Littwin, Omid Saremi, Madhu Advani, Vimal Thilak, Preetum Nakkiran, Chen Huang, and Joshua Susskind. How jepa avoids noisy features: The implicit bias of deep linear self distillation networks. *arXiv preprint arXiv:2407.03475*, 2024.

[11] Vimal Thilak, Chen Huang, Omid Saremi, Laurent Dinh, Hanlin Goh, Preetum Nakkiran, Joshua M. Susskind, and Etai Littwin. LiDAR: Sensing linear probing performance in joint embedding SSL architectures. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[14] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[16] Amir Bar, Florian Bordes, Assaf Shocher, Mahmoud Assran, Pascal Vincent, Nicolas Ballas, Trevor Darrell, Amir Globerson, and Yann LeCun. Stochastic positional embeddings improve masked image modeling. In *International Conference on Machine Learning*, 2023.

[17] 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.

[18] 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.

[19] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[21] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv: 1708.03888*, 2017.

[22] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, University of Toronto, ON, Canada, 2009.

[23] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2019.

[24] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461, 2014.

[25] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

A Experimental Details

Architecture Details. We instantiate the context, target and predictor modules in both IJEPA and EC-IJEPA models as Vision Transformers (ViTs) [15]. We experiment with two different model sizes for the encoder modules, i.e. ViT-Large and ViT-Huge, and a lower capacity ViT Predictor following IJEPA [14]. Table 4 and Table 5 respectively show the relevant architecture hyperparameters for the ViT-based encoders and predictors.

Table 4: Encoder architecture using ViT-based models. The value after "/" indicates the patch size.

Architecture	Depth	Hidden Dimension	Number of Heads
ViT-L/16	24	1024	16
ViT-H/14	32	1280	16

Table 5: Predictor architecture using ViT-based models. Number of heads is set to match that of the encoder.

Architecture	Depth	Hidden Dimension	Number of Heads
ViT-Predictor	12	384	16

Pretraining Details. We use the AdamW optimizer [20] [2] to train IJEPA and EC-IJEPA in all our experiments. Table 6 and Table 7 show the hyperparameters used to pretrain all models in this work. We follow the pretraining configuration from IJEPA [14]. We follow masking hyperparameters used to create context and target masks from IJEPA [14].

Table 6: Pretraining hyperparameters used for ViT-L/16

Hyperparameter	Value
Optimizer	AdamW
Epochs	600
Max learning rate	0.001
LR Warmup type	Linear
LR Decay type	Cosine
Warmup epochs	15
Batch size	2048
Weight decay scheduler	Cosine
Weight decay (start, end)	[0.04, 0.4]
EMA momentum scheduler	Linear
EMA momentum (start, end)	[0.996 1.0]

Evaluation on ImageNet-1k We evaluate the pretrained encoders described above using linear probing on ImageNet-1k dataset [13]. We adapt the evaluation protocol from IJEPA [14] wherein the pretrained model weights are frozen and are used to extract a feature vector by average pooling (across the sequence length) the output tokens from the last layer of the encoder. A linear probe that consists of a batch normalization layer with non-learnable affine parameters followed by a linear layer is used to map this feature vector to the set of classification logits on ImageNet-1k dataset. The parameters of the linear probe are trained with the LARS [21] optimizer using a learning rate of 0.05, no weight decay and with a batch size of 16384 for 50 epochs.

[2] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

Table 7: Hyperparameter configuration used to pretrain ViT-H/14

Hyperparameter	Value
Optimizer	AdamW
Epochs	300
Max learning rate	0.001
LR Warmup type	Linear
LR Decay type	Cosine
Warmup epochs	40
Batch size	2048
Weight decay scheduler	Cosine
Weight decay (start, end)	[0.04, 0.4]
EMA momentum scheduler	Linear
EMA momentum (start, end)	[0.996 1.0]

Evaluation on out-of-distribution (OOD) datasets We use CIFAR10, CIFAR100 [22], EuroSAT [23], Food101 [24], SUN397 [25], CLEVR/Count and CLEVR/Dist [18, 19] as unseen or OOD datasets w.r.t the pretraining dataset (ImageNet-1k). We again adopt the evaluation protocol of linear probing with a frozen backbone. We follow the evaluation protocol used in VISSL [17] also used in prior works [14, 16] to train and evaluate a linear probe for the OOD datasets. Table 8 lists the relevant hyperparameter configurations used in our experiments.

Table 8: Hyperparameters used for linear evaluation on OOD datasets.

Dataset	Optimizer	Momentum	Weight decay	Learning rate (LR)	Epochs
CIFAR10	SGD with Nesterov	0.9	0.0005	0.01	28
CIFAR100	SGD with Nesterov	0.9	0.0005	0.01	28
EuroSAT	SGD with Nesterov	0.9	0.0005	0.01	28
Food101	SGD with Nesterov	0.9	0.0005	0.01	28
SUN397	SGD with Nesterov	0.9	0.0005	0.01	28
CLEVR/Count	SGD with Nesterov	0.9	0.0005	0.01	50
CLEVR/Dist	SGD with Nesterov	0.9	0.0005	0.01	50

B Additional Results

Average Pooling Ablation. EC-IJEPA uses average pooling with a kernel size and stride of $[4, 4]$ respectively at inference time to create conditioning position tokens as described in Section 2. We perform an ablation experiment to measure the impact of kernel size and stride on downstream classification accuracy on ImageNet-1k [13] by varying these hyperparameters. Figure 4 shows the maximum classification accuracy achieved on ImageNet-1k validation as a function of kernel size and stride. We observe from Figure 4 that the highest accuracy is achieved with a kernel size of 4 and stride of 4. Furthermore, we observe that there is a drop off in accuracy for kernel size of 1 and stride of 1. These observations suggest that the values for these hyperparameters used in Section 2 are reasonable to extract representations from EC-IJEPA for classification tasks.

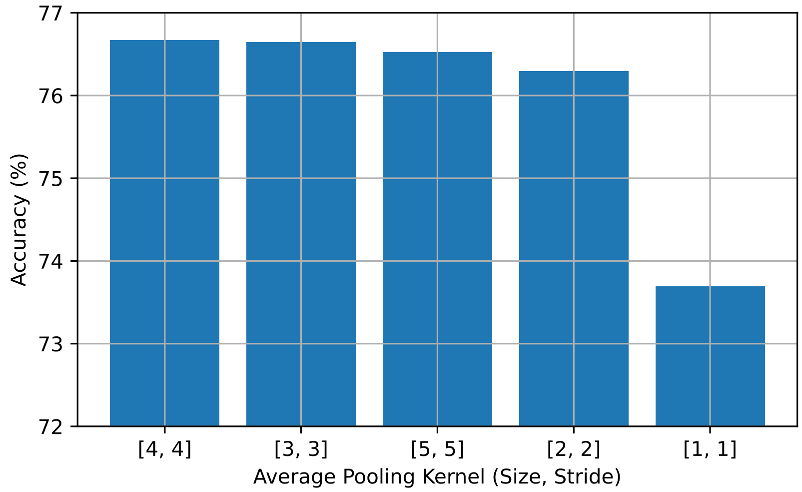

Figure 4: Linear probing accuracy on Imagenet-1k dataset w.r.t kernel size and stride.