Energy-Inspired Self-Supervised Pretraining for Vision Models

Ze Wang [1] **Jiang Wang** [2] **Zicheng Liu** [2] **Qiang Qiu** [1]

Abstract

Motivated by the fact that forward and backward passes of a deep network naturally form symmetric mappings between input and output representations, we introduce a simple yet effective self-supervised vision model pretraining framework inspired by energy-based models (EBMs). In the proposed framework, we model energy estimation and data restoration as the forward and backward passes of a single network without any auxiliary components, e.g., an extra decoder. For the forward pass, we fit a network to an energy function that assigns low energy scores to samples that belong to an unlabeled dataset, and high energy otherwise. For the backward pass, we restore data from corrupted versions iteratively using gradient-based optimization along the direction of energy minimization in as few as one step. Our framework accepts a wide range of pretext tasks with different data corruption methods, and permits models to be pretrained from masked image modeling and image restoration. We support our findings with extensive experiments, and show the proposed method delivers comparable and even better performance with remarkably fewer epochs of training compared to the state-of-the-art self-supervised vision model pretraining methods. Our findings shed light on further exploring self-supervised vision model pretraining pretext tasks beyond masked image modeling.

1. Introduction

The remarkable success of exploiting context information resides in the massive unlabeled data in natural language processing (NLP) stimulates the recent progress of self-supervised vision model through masked image modeling (MIM) (He et al., 2021; Wei et al., 2021; Xie et al., 2022). Such methods typically adopt an auto-encoder (AE) architecture, where the target vision model to be pretrained serves as an encoder to encode an image with incomplete pixel information to a latent representation. An auxiliary decoder is jointly trained to restore the missing information from the latent representation. On the other hand, contrastive self-supervised learning methods (Chen et al., 2020) usually require very large training batch sizes to provide sufficient negative samples, which lead to considerable hardware demand that is usually costly to afford. Recent contrastive self-supervised learning methods (Grill et al., 2020; Chen & He, 2021; Tian et al., 2021; He et al., 2020; Chen et al., 2021) alleviate the huge batch challenge at the cost of deploying an momentum copy of the target model to facilitate the training and prevent trivial solutions.

The sophisticated designs of self-supervised vision model training motivate us to consider the following question: *Can we train a standard deep network to do both representation encoding and masked prediction simultaneously, so that no auxiliary components, heavy data augmentations, or modifications to the network structure are demanded?*

Hinted by the fact that the forward and the backward passes of a deep network naturally form symmetric mappings between input and output representations, we extend the recent progress on energy-based models (EBMs) (Xie et al., 2016; Du & Mordatch, 2019; Du et al., 2020; Zhao et al., 2017) and introduce a model-agnostic self-supervised framework that pretrains any deep vision models to . Given an unlabeled dataset, we train the **forward pass** of the target vision model to perform discriminative recognition. Instead of instance-wise classification as in contrastive self-supervised learning, we train the target vision model to perform binary classification by fitting it to an energy function that assigns low energy values to positive samples from the dataset and high energy values otherwise. And we train the **backward pass** of the target vision model to perform conditional image restoration as in masked image modeling methods, by restoring positive image samples from their corrupted versions through conducting gradient-based updating iteratively along the direction of energy minimization. Such conditional sampling schemes can produce samples with satisfying quality using as few as one gradient step, thus pre-

[1]Purdue University [2]Microsoft Corporation. Correspondence to: Ze Wang, Qiang Qiu <{zewang, qqiu}@purdue.edu>, Jiang Wang, Zicheng Liu <{jiangwang, zliu}@microsoft.com>.

First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022.

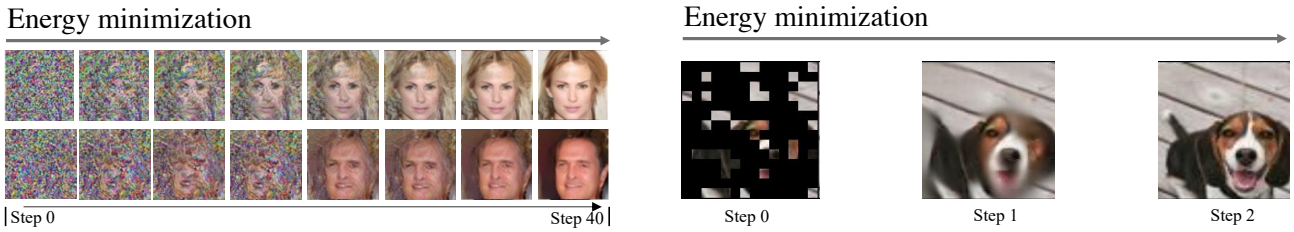

Figure 1. Typical EBM sampling demands long chains even with a mild resolution of 64×64 (left). Our conditional sampling with short chains obtain satisfactory results with as few as a single gradient step at a standard resolution of 224×224 (right).

vents the unaffordable cost of applying the standard implicit sampling of EBMs on high-dimensional data. In this way, we naturally fold the encoder-decoder architecture widely used in masked image modeling into the forward and backward passes of a single vision model, so that the structure tailored for discriminative tasks is fully preserved with *no auxiliary components or heavy data augmentation* needed. Therefore the obtained vision model can better preserve the representation discriminability and prevent knowledge loss or redundancy.

Moreover, after folding the corrupted data modeling (encoder) and the original data restoration (decoder) into a single network, the proposed framework now accepts a broader range of pretext tasks to be exploited. Specifically, we demonstrate that beyond typical masked image modeling, the proposed framework can be easily extended to learning from learning from image restoration, e.g., super-resolution and image colorization.

We demonstrate the effectiveness of the proposed method with extensive experiments on ImageNet-1K. It is easy to notice that almost every parameter trained from the self-supervised training stage will be effectively used in the downstream fine-tuning. And we show that competitive performance can be achieved even with only 100 epochs of pretraining on a single 8-GPU machine.

2. Method

In this section, we introduce in details the proposed framework of energy-inspired self-supervised vision model pretraining. We begin this section by briefly reviewing the backgrounds of energy-based model in Section 2.1. We present the general process of the proposed pretraining framework, with a straightforward example based on mask image modeling in Section 2.2. We then present how the proposed framework allows extensions to a wide range of variants adopting different pretext tasks

2.1. Backgrounds

EBM training is accomplished by learning an energy function that predicts the unnormalized density, named the energy score, for a given data sample. Specifically, given a data sample $\mathbf{x} \in \mathbb{R}^d$, the energy function $E_\theta(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$, with θ as the learnable parameters, maps the sample to its energy score, which is expected to be low for the in-distribution (positive) samples, and high for the out-of-distribution (negative) samples. The modeled data density $p_\theta(\mathbf{x})$ is then expressed as: $p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta}$, where $Z_\theta = \int_{\mathbf{x}} \exp(-E_\theta(x))$ is the partition function. Approximating a target data distribution $p_{\text{data}}(\mathbf{x})$ equals to minimizing the expected negative log-likelihood function over the data distribution, defined by the maximum likelihood loss function:

$$\mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[-\log p_\theta(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[E_\theta(\mathbf{x}) + \log Z_\theta]. \tag{1}$$

As the computation of $\mathcal{L}_{\text{ML}}$ involves the intractable Z_θ, the common practice is to represent the gradient of $\mathcal{L}_{\text{ML}}$ as,

$$\nabla_\theta \mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}^+\sim p_{\text{data}}(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^+)] - \mathbb{E}_{\mathbf{x}^-\sim p_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^-)]. \tag{2}$$

The objective in (2) trains the model $\mathbb{E}_\theta$ to effectively distinguish in-domain and out-of-domain samples by decreasing the predicted energy of positive data samples $\mathbf{x}^+$ from the true data distribution and increasing the energy of negative samples $\mathbf{x}^-$ obtained through sampling from the model p_θ.

Sampling from the modeled distribution equals to finding the samples with low energy scores $\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} E_\theta(\mathbf{x})$. In practice, directly evaluating the above $\arg\min$ can be intractable. Inspired by MCMC based sample techniques such Langevin dynamics (Welling & Teh, 2011), common practice (Du & Mordatch, 2019; Du et al., 2020) resorts to gradient-based optimization for implicit sampling. Specifically, by performing N gradient steps, the approximated optimum $\tilde{\mathbf{x}}^N$ can be obtained as

$$\tilde{\mathbf{x}}^n = \tilde{\mathbf{x}}^{n-1} - \alpha \nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{n-1}), n = 1, \dots N, \tag{3}$$

where α is the step size of the gradient-based optimization. $\tilde{\mathbf{x}}^0$ is usually obtained by sampling from a predefined prior distribution such as Uniform noise.

2.2. Proposed Framework

We denote the deep vision model to be pretrained as ψ.An energy-based model can be constructed by simply append-

ing a linear head h with a single output dimension to the feature extractor, i.e., $E_\theta(\mathbf{x}) = h(\boldsymbol{\psi}(x))$ with θ collectively denoting the parameters of both $\boldsymbol{\psi}$ and h. In a typical setting, the linear head h contains only hundreds of parameters. After the pretraining, the obtained vision model can be directly used as an image recognition model by only replacing the linear head h. The full preservation of network architecture with no auxiliary network components, e.g., a decoder, to be removed, better maintains the network discriminability and prevents potential feature redundancy.

As illustrated in Figure 1, even using a low resolution, the typical implicit sampling of EBMs in (3) can take dozens or even hundreds of gradient steps to produce an image sample of satisfying quality (Du & Mordatch, 2019; Zhao et al., 2020). Applying the standard EBM training to self-supervised pretraining introduces unaffordable cost. In this paper, we forgo the from-scratch sampling and train the network to perform conditional sampling, so as to restore partially corrupted data with explicit supervision. As visualized in Figure 1, the costly noise-to-image sampling of EBMs is now replaced with conditional sampling, where a chain of sampled data moving towards the low-energy region are obtained for each corrupted sample rapidly. In our case of self-supervised learning, doing so has two major advantages. The proposed framework now allows the restoration of each sample to be completed with as few as two gradient optimization steps, and permits desirable speed for self-supervised training on large scale datasets. Moreover, such conditional sampling allows us to replace (2) designed for unconditional sampling by explicit supervision with pixel values as we will discuss later, and such strong supervision alleviates the unstable EBMs training according to our observations.

The proposed framework imposes little restrictions to the image sample corruption methods deployed and permits a wide range of pretext tasks to be exploited. For the sake of discussion, we present in details one straightforward variant with masked image modeling to walk through the training process, and illustrate other possible variants in later sections.

Masked image modeling. As visualized in Figure 2, given a batch of image samples $\{\mathbf{x}_i\}_{i=1,\dots,K}$, we first corrupt each image using a predefined function $\downarrow(\cdot)$. In this example, $\downarrow(\cdot)$ denotes random image masking. After image masking, $\downarrow(\mathbf{x}_i)$ can be seen as a sample that is out of the target data distribution p_{data} with the remaining pixels inferring the original contents of the image. With the target modeling a continuous energy function, we can perform online evaluation to the estimated energy function by examining how well moving the masked image in the modeled energy space along the energy minimization direction can restore the original data $\mathbf{x}_i$. Specifically, we resort to the gradient based optimization (3) and perform N-step image restoration with $\tilde{\mathbf{x}}_i^0 = \downarrow(\mathbf{x}_i)$. The loss of the restoration steps can then be expressed as:

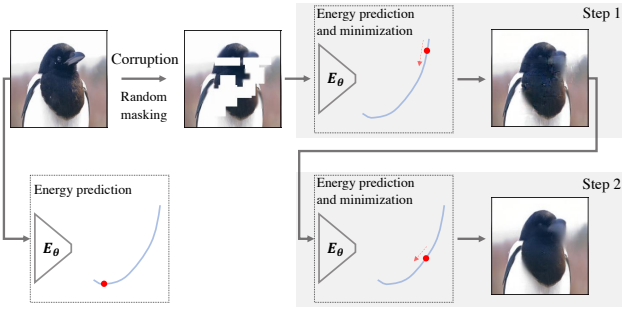

Figure 2. Applying the proposed framework to masked image modeling. The unlabeled image is corrupted with random patches, and the network is trained to recognize the corrupted sample as a negative one with high energy, and recover the original image by updating the image iteratively along the direction of energy minimization.

$$\mathcal{L} = \frac{1}{KN}\sum_{i=0}^{K}\sum_{j=0}^{N}\text{MSE}(\tilde{\mathbf{x}}_i^j, \mathbf{x}_i), \quad \text{where} \quad \tilde{\mathbf{x}}_i^j = \tilde{\mathbf{x}}_i^{j-1} - \alpha\nabla_{\mathbf{x}}E_\theta(\text{SG}(\tilde{\mathbf{x}}_i^{j-1})), \tag{4}$$

with SG denoting the *stop gradient* operation that blocks the gradient propagation across steps. We empirically observe that adding stop gradient operations between consecutive steps helps accelerate the training speed and convergence. The $\mathcal{L}_{\text{restore}}$ term here encourages original images to be restored from the negative images (corrupted versions and the sampled versions along the sampling chains of (4)) by gradient based updating along the direction of energy minimization, which equally encourages higher energy values for negative images, and can functionally replace the second term in (2).

Notably, as discussed in (Du & Mordatch, 2019), standard EBM training with (2) using arbitrary energy model can cause sharp changes in gradients, and the stable training requires heavy tuning to the hyperparameters and techniques like spectral normalization to constrain the Lipschitz constant of the network. While in our framework, unstable training caused by sharp gradients is naturally prevented by the explicit supervision in (4), as faithfully restoring the original data requires the gradient in (4) to be bounded within a certain range. We summarize the overall training steps of the proposed framework in Algorithm 1. We further provide PyTorch-style pseudo code in Appendix Section A.4 to facilitate reproducing our results.

Recent self-supervised vision model pretraining methods (Xie et al., 2022; He et al., 2021; Wei et al., 2021) invariably adopt masked image modeling as the pretext task. We argue that the encoder-decoder architectures used in these methods prevent them from being extended to other pretext

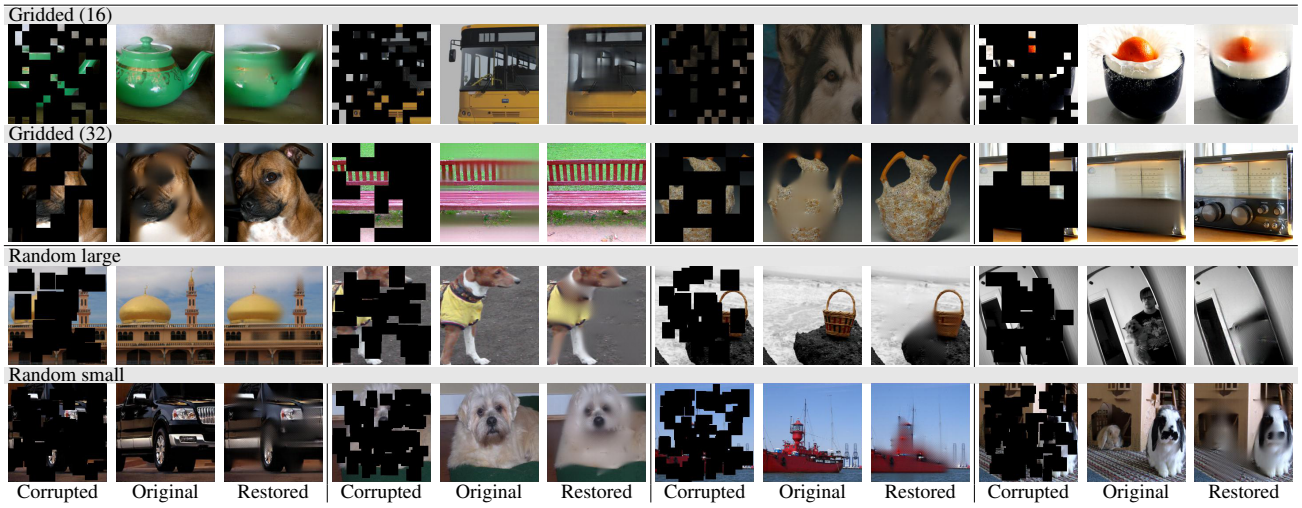

Figure 3. Qualitative results of learning from masked image modeling. We present results with different masking strategies.

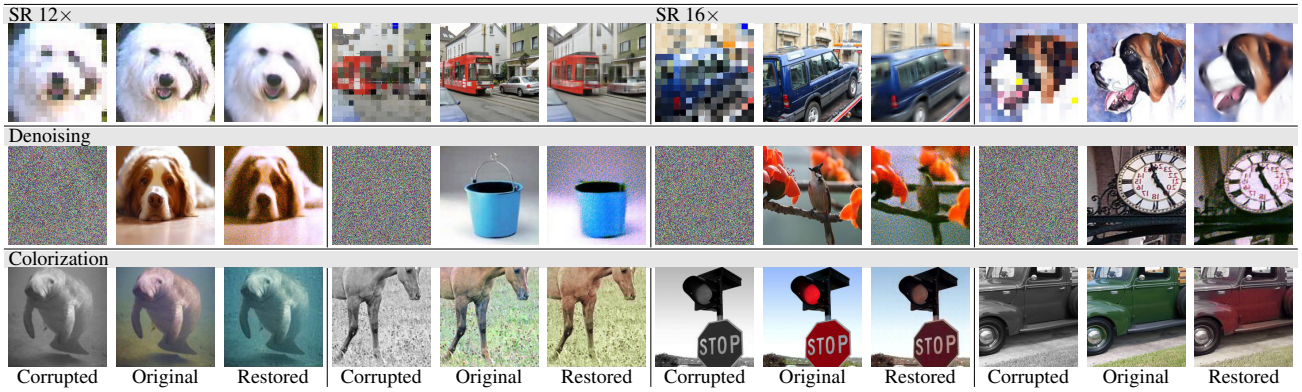

Figure 4. Qualitative results of the experiments with learning from image restoration. The proposed framework accepts a broader range of pretext tasks.

tasks. In the auto-encoder based methods, the vision model to be pretrained serves as the encoder, and is only exposed with the corrupted images during pretraining. Therefore, it is important to present part of the original image patches to the encoder, so that the encoder can learn from those intact patches network weights that transfer well in downstream finetuning. While in the proposed pretraining framework, both corrupted samples and original samples are exposed to the target vision model, in the forms of input and supervision, respectively. By simply replacing the corruption function $\downarrow(\cdot)$, we can establish variants that learn vision models from super-resolution, denoising, and image colorization. Further details and results will be discussed in Section 3.1. With certain degrees of global image corruption, the network can be trained to infer possible content given the incomplete pixel information, and restore the missing information, such as detailed textures or color, by the patterns learned from the true data and stored in the network weights. With the restriction to the image corruption methods being lifted, the proposed framework stimulates further discussions on the pretext tasks of vision model pretraining.

3. Experiments

3.1. Self Comparisons

The proposed framework accepts a wide range of variants with different pretext tasks. To illustrate the flexibility, we present results with different variants including learning from masked image modeling and image restoration.All results in this section are obtained by pretraining and finetuning a ViT-S for 100 epochs on the ImageNet-1K (Deng et al., 2009) dataset.

Learning from masked image modeling. A straightforward way of implementing the proposed framework is to train the network to perform masked image modeling given incomplete pixel information. We present results obtained with different masking strategies and ratios of masking in Table 1. Detail regarding the masking strategy is in Appendix Section A.3. Our method achieves good performance across

Table 1. Masked image modeling with different patterns and ratios of image masking. The result of MAE (He et al., 2021) with 400 epochs is based on our reimplementation. The results of our methods are obtained by 100 epochs of pretraining. All results are obtained with 100 epochs of finetuning. Baseline results are in gray.

Masking strategies	Accuracy				
From scratch	76.6				
Random large	79.7				
Random small	79.3				
% of masking	10%	30%	50%	70%	90%
MAE (He et al., 2021)	-	-	-	78.3	-
Gridded (16)	76.7	78.3	78.7	79.0	78.8
Gridded (24)	76.8	78.2	78.7	79.2	78.8
Gridded (32)	77.1	78.4	78.6	79.0	78.7

Table 2. Results obtained by different pretext tasks of learning from image restoration. Baseline results are in gray.

Methods	Accuracy	Methods	Accuracy
From scratch	76.6		
AE + SR 16 ×	77.1	AE + denoising	76.8
SR 14×	78.2	SR 16×	79.6
SR 24×	78.4	SR 32×	76.3
Colorization	78.2	Denoising	79.2

different masking schemes.

Learning from image restoration. Our framework enjoys higher flexibility as the pretrained vision model is exposed with both true samples and artificial negative ones, thus even when the input images are corrupted globally, our framework can still learn good models. To show this, we present in Table 2 results obtained with learning from image restoration. Specifically, we train the network to learn from image super-resolution, denoising, and image colorization, where every pixel is corrupted with a predefined function. Table 2, *SR* denotes super-resolution. *AE + SR 16* denotes a baseline experiment with a auto-encoder architecture as in (He et al., 2021). Detail regarding the corruption methods is in Appendix Section A.2.

As shown in the quantitative results in Table 2 and visualization in Appendix Figure 4, with proper degrees of corruption, restoring the original images may require the network to infer the general content given the corrupted pixels, and recover the details using the knowledge learned from the true samples and stored in the network weights. For example, in the image colorization experiments, the pretrained vision model learns the common colors of different objects from the massive unlabeled data in a self-supervised way. As visualized in Appendix Figure 4, the vision model learns from the unlabeled data common knowledge such as stop signs are usually red, and the background of a horse is usually green while manatees are marine mammals therefore the background is usually blue.

Table 3. Quantitative comparisons against the recent self-supervised model pretraining methods. $*$ denotes results produced by our re-implementation. PT and FT denote pretraining and finetuning, respectively. † denotes the training involves external dataset other than ImageNet-1K. For our results, we set $e = 100$ for ViT-B and $e = 200$ for ViT-S.

Methods	(PT + FT)	ViT-S	ViT-B
From scratch	300	79.6*	82.3
DINO (Caron et al., 2021)	-	-	82.8
MoCo-V3 (Chen et al., 2021)	300+150	-	83.2
BEiT† (Bao et al., 2021)	800+100	-	83.2
MaskFeat (Wei et al., 2021)	300+100	-	83.6
iBOT (Zhou et al., 2022)	600 + 200	81.4	-
iBOT (Zhou et al., 2022)	1600 + 100	-	83.8
MAE (He et al., 2021)	400 + 100	78.3*	83.1*
MAE (He et al., 2021)	1600 + 100	-	83.6
Ours Mixed	200 + e	81.2	83.1
Ours Mixed	800 + e	81.9	83.3

3.2. Quantitative Comparisons Against Recent Methods

In this section, we present quantitative comparisons against the recent self-supervised model pretraining methods. We train our method using a mixture of pretext tasks that are uniformly sampled from image masking, super-resolution, denoising, and colorization. All ImageNet results are evaluated on the validation set with a single center crop of 224×224 for each image. In Table 3, we compare our method against DINO (Caron et al., 2021), MoCo-V3 (Chen et al., 2021), MaskFeat (Wei et al., 2021), BEiT (Bao et al., 2021), iBOT (Zhou et al., 2022), and MAE (He et al., 2021). With only 200 epochs of pretraining, the proposed framework can achieve comparable or even better performance with the state-of-the-art self-supervised pretraining methods, some of which adopt much more epochs and leverage external data for training.

4. Conclusion

We presented energy-inspired self-supervised vision model pretraining. We accelerated EBM training and trained the vision model to perform conditional sampling initialized from corrupted sampled by moving them along the direction of energy minimization. The bi-directional mappings between images and latent representations are modeled naturally by the forward and backward passes of a network, which fully preserve the discriminative structure of the target vision model and avoid auxiliary network components and sophisticated data augmentation to facilitate pretraining. The proposed framework accepts a wider range of pretext tasks with different data corruption methods. We hope our findings can shed light on further exploring the pretext tasks of self-supervised vision model pretraining.

References

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

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

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *CVPR*, 2021.

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

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. *ICLR*, 2020.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *NeurIPS*, 2019.

Du, Y., Li, S., Tenenbaum, J., and Mordatch, I. Improved contrastive divergence training of energy based models. *ICML*, 2020.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *NeurUPS*, 33:6840–6851, 2020.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *ECCV*, 2016.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *ICLR*, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *ICLR*, 2021.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Tian, Y., Chen, X., and Ganguli, S. Understanding self-supervised learning dynamics without contrastive pairs. *ICML*, 2021.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*. Citeseer, 2011.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *ICML*, 2016.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. *CVPR*, 2022.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *ICLR*, 2018.

Zhao, J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial network. *ICLR*, 2017.

Zhao, Y., Xie, J., and Li, P. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *ICLR*, 2020.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *CVPR*, pp. 633–641, 2017.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. Ibot: Image bert pre-training with online tokenizer. *ICLR*, 2022.

Appendix

A. Implementation Details

A.1. Details on Training

We use AdamW (Loshchilov & Hutter, 2019) as the optimizer for both self-supervised training and tuning. For all the self-supervised pretraining experiments, we adopt only random cropping and random horizontal flipping as the data augmentation. We present comprehensive training details in Appendix Section A.1 Table A. Most of the experimental settings follow (He et al., 2021). Unlike recent methods (Zhou et al., 2022; He et al., 2021), we *do not* perform exhaustive searches for the optimal hyperparameters such as learning rates. Training energy functions introduces a new hyperparameter α, which is the step size of the gradient optimization to the corrupted data. Thanks to the explicit supervision available in the proposed framework, we can set α to be learnable, and jointly train it with the network without the concern of training stability as in standard EBM training. If not otherwise specified, we adopt $N = 2$, i.e., two steps of gradient-based energy minimization in the pretraining stage for the best performance-efficiency trade-off.

We present the training details for both self-supervised training and finetuning in Table A. All experiments are implemented using PyTorch (Paszke et al., 2019). We use the default API for automatic mixed-precision training.

Configurations	Pretraining	Finetuning
optimizer	AdamW	AdamW
base learning rate	1e-4	1e-3
learning rate schedular	Cosine decay	Cosine decay
weight decay	0.05	0.05
momentum of AdamW	$\beta_1 = 0.9, \beta_2 = 0.95$	$\beta_1 = 0.9, \beta_2 = 0.999$
layer-wise learning rate decay (Clark et al., 2020)	-	0.75
batch size	256	1024
drop path (Huang et al., 2016)	-	0.1
augmentation	RandomResizedCrop	RandAug (9, 0.5) (Cubuk et al., 2020)
label smoothing (Szegedy et al., 2016)	-	0.1
mixup (Zhang et al., 2018)	-	0.8
cutmix (Yun et al., 2019)	-	1.0
Mix-precision training	✓	✓

Table A. Training details for both self-supervised pretraining and finetuning.

A.2. Learning from Image Restoration

In the s-time super-resolution (denoted as *SR* $s\times$), the image are first downsampled using bicubic interpolation for s times, and resized back to the original size using nearest-neighbor interpolation. In the denoising experiments, we take a noise scheme inspired by diffusion models (Song et al., 2021; Ho et al., 2020) with $\downarrow(\mathbf{x}) = \sqrt{\gamma}\mathbf{x} + \sqrt{1-\gamma}\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$ and γ uniformly sampled as $\gamma \sim \mathcal{U}(0, 1)$.

A.3. Learning from Masked Image Modeling.

As visualized in Figure 3, in the experiments with gridded mask, we evenly divide an image into squared patches with the same size, and randomly mask out a portion of the patches. Note that in the *Gridded (16)* experiments, the patch partition in the image masking matches exactly with the patch partition in the ViT networks, therefore it is a fair comparison against MAE (He et al., 2021). For the random masking experiments, we randomly place blank patches with the size and aspect ratio sampled from a particular range to each image. In the *Random small* experiments, we randomly place 75 blank patches with normalized sizes sampled from a Uniform distribution of $\mathcal{U}(0.01, 0.025)$. In the *Random large* experiments, we randomly place 25 blank patches with normalized sizes sampled from $\mathcal{U}(0.02, 0.05)$. For both experiments, the aspect ratio of each patch is sampled from $\mathcal{U}(0.5, 2.0)$.

A.4. Algorithm and Pseudo Code in PyTorch Style

Algorithm 1 Energy-based self-supervised vision model pretraining.

Given: A target network ψ to be pretrained, a large-scale unlabeled dataset $\{\mathbf{x}_i\}$, and an image sample corruption function $\downarrow(\cdot)$.
Given: Step size α and number of steps N for the gradient update of corrupted samples.
Initialize the target network ψ and the linear head h.
repeat
  Sample a batch of images from the unlabeled dataset.
  Corrupt each sample and initialize the conditional sampling chains as $\tilde{\mathbf{x}}_i^0 = \downarrow(\mathbf{x}_i)$.
  for Step $n = 1 : N$ **do**
    Stop gradient $\tilde{\mathbf{x}}_i^{n-1} = \text{SG}(\tilde{\mathbf{x}}_i^{n-1})$.
    Perform gradient update to the corrupted samples as in (4).
  end for
  Compute the restoration error of each step using (4), and update ψ and h with gradient optimization.
until Converge
Return ψ.

```
model = VisionModel()
# initialize deep vision model with any architectures
head = Linear(in_channels=model.dim, out_channels=1, bias=False)
# initialize a simple linear head for energy score prediction

criterion = SmoothL1Loss(beta=1.0)
# define loss function for image reconstruction

optimizer = AdamW(model.parameters() + head.parameters())
# initialize parameter optimizer

# training loop
for images in image_loader:
    # images with shape [n, c, h, w]
    corrupted_images = corruption_method(images)

    loss = 0

    for _ in num_steps:
        corrupted_images = corrupted_images.detach()
        # stop gradients between inner-loop steps.
        energy_score = head(model(corrupted_images))
        # energy score with shape [n, 1]

        im_grad = autograd(energy_score.sum(), corrupted_images)
        # compute the gradient of input pixels along the direction
        # of energy maximization
        corrupted_images = corrupted_images - alpha * im_grad
        # gradient descent along the direction of energy minimization

        loss += criterion(corrupted_images, images)

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

```

Listing 1. PyTorch-style pseudo code of the proposed pretraining framework.

B. Additional Analysis

B.1. Performance with Different N.

We present performance obtained with different N steps of gradient update to the corrected samples. We use $N = 2$ for the best performance-efficiency trade-off and the proposed framework can perform fairly well with as few as a single step of

gradient update to each corrupted sample.

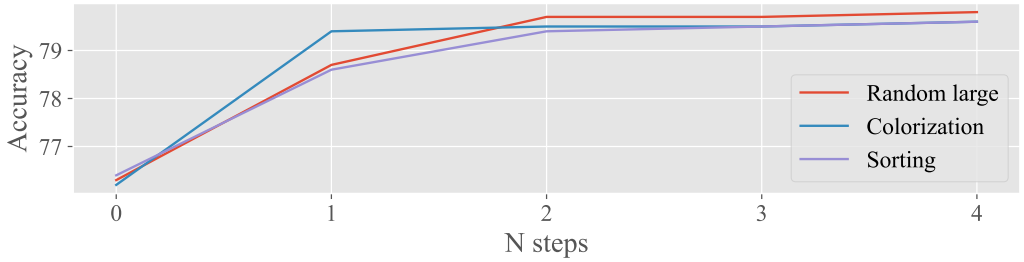

Figure A. Performance with different N. $N = 0$ corresponds to using corrupted images as negative.

C. Experiments

C.1. Other Network Architectures and Downstream Transfer

Different from models like MAE (He et al., 2021) and SimMIM (Xie et al., 2022) that are specifically tailored for particular network architectures, our framework can be seamlessly applied to any deep vision models without any customization or auxiliary network components beside the simple linear head h. To show this, we present results with convolution-based ConvNeXts (Liu et al., 2022) and Swin-Transformer (Liu et al., 2021) in Table B. And to validate the effectiveness to the downstream transfer, we finetune the pretrained network on the ADE20K (Zhou et al., 2017) semantic segmentation dataset, and present the results in Table C.

Table B. The proposed framework can be seamlessly applied to any deep vision models. FS, PT, and FT denote from-scratch training, pretraining, and finetuning, respectively.

Networks	FS 300E	PT 200E + FT 100E
ConvNeXt-T	82.1	82.7
Swin-T	81.3	82.2

Table C. IoU results with ADE20K semantic segmentation finetuning.

method	data	ViT-B
ViT-L	supervised ImageNet	47.4
MoCo-v3	IN1K	47.3
BEiT	ImageNet+DALL-E	47.1
MAE	ImageNet	48.1
Ours	ImageNet	47.6