Generative Pretraining from Pixels

Mark Chen [1] **Alec Radford** [1] **Rewon Child** [1] **Jeff Wu** [1] **Heewoo Jun** [1] **David Luan** [1] **Ilya Sutskever** [1]

Abstract

Inspired by progress in unsupervised representation learning for natural language, we examine whether similar models can learn useful representations for images. We train a sequence Transformer to auto-regressively predict pixels, without incorporating knowledge of the 2D input structure. Despite training on low-resolution ImageNet without labels, we find that a GPT-2 scale model learns strong image representations as measured by linear probing, fine-tuning, and low-data classification. On CIFAR-10, we achieve 96.3% accuracy with a linear probe, outperforming a supervised Wide ResNet, and 99.0% accuracy with full fine-tuning, matching the top supervised pre-trained models. We are also competitive with self-supervised benchmarks on ImageNet when substituting pixels for a VQVAE encoding, achieving 69.0% top-1 accuracy on a linear probe of our features.

1. Introduction

Unsupervised pre-training played a central role in the resurgence of deep learning. Starting in the mid 2000's, approaches such as the Deep Belief Network (Hinton et al., 2006) and Denoising Autoencoder (Vincent et al., 2008) were commonly used in neural networks for computer vision (Lee et al., 2009) and speech recognition (Mohamed et al., 2009). It was believed that a model which learned the data distribution $P(X)$ would also learn beneficial features for the subsequent supervised modeling of $P(Y|X)$ (Lasserre et al., 2006; Erhan et al., 2010). However, advancements such as piecewise linear activation functions (Nair & Hinton, 2010), improved initializations (Glorot & Bengio, 2010), and normalization strategies (Ioffe & Szegedy, 2015; Ba et al., 2016) removed the need for pre-training in order to achieve strong results. Other research cast doubt on the benefits of *deep* unsupervised representations and reported strong results using a single layer of learned features (Coates et al., 2011), or even random features (Huang et al., 2014; May et al., 2017). The approach fell out of favor as the state of the art increasingly relied on directly encoding prior structure into the model and utilizing abundant supervised data to directly learn representations (Krizhevsky et al., 2012; Graves & Jaitly, 2014). Retrospective study of unsupervised pre-training demonstrated that it could even hurt performance in modern settings (Paine et al., 2014).

Instead, unsupervised pre-training flourished in a different domain. After initial strong results for word vectors (Mikolov et al., 2013), it has pushed the state of the art forward in Natural Language Processing on most tasks (Dai & Le, 2015; Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018). Interestingly, the training objective of a dominant approach like BERT, the prediction of corrupted inputs, closely resembles that of the Denoising Autoencoder, which was originally developed for images.

As a higher dimensional, noisier, and more redundant modality than text, images are believed to be difficult for generative modeling. Here, self-supervised approaches designed to encourage the modeling of more global structure (Doersch et al., 2015) have shown significant promise. A combination of new training objectives (Oord et al., 2018), more recent architectures (Gomez et al., 2017), and increased model capacity (Kolesnikov et al., 2019) has allowed these methods to achieve state of the art performance in low data settings (Hénaff et al., 2019) and sometimes even outperform supervised representations in transfer learning settings (He et al., 2019; Misra & van der Maaten, 2019).

Given that it has been a decade since the original wave of generative pre-training methods for images and considering their substantial impact in NLP, this class of methods is due for a modern re-examination and comparison with the recent progress of self-supervised methods. We re-evaluate generative pre-training on images and demonstrate that when using a flexible architecture (Vaswani et al., 2017), a tractable and efficient likelihood based training objective (Larochelle & Murray, 2011; Oord et al., 2016), and significant compute resources (1024 TPU cores), generative pre-training is competitive with other self-supervised approaches and learns

[*]Equal contribution [1]OpenAI, San Francisco, CA, USA. Correspondence to: Mark Chen <mark@openai.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

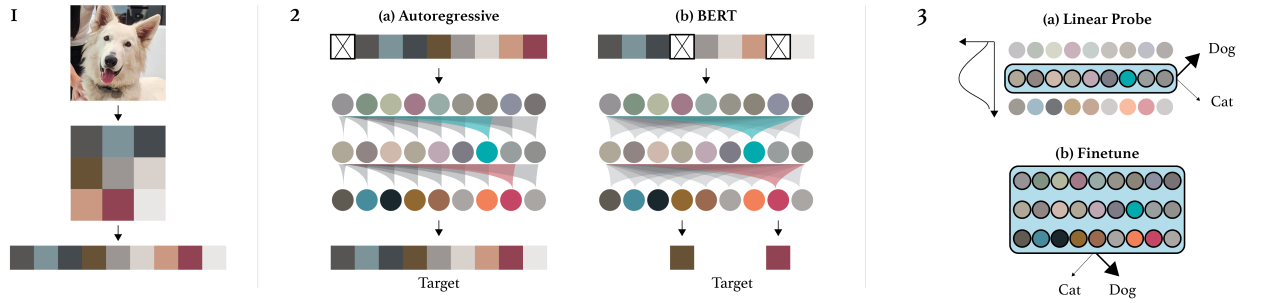

Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

representations that significantly improve the state of the art in low-resolution unsupervised representation learning settings.

This is especially promising as our architecture uses a dense connectivity pattern which does not encode the 2D spatial structure of images yet is able to match and even outperform approaches which do. We report a set of experiments characterizing the performance of our approach on many datasets and in several different evaluation settings (low data, linear evaluation, full fine-tuning). We also conduct several experiments designed to better understand the achieved performance of these models. We investigate how representations are computed inside our model via the performance of linear probes as a function of model depth as well as studying how scaling the resolution and parameter count of the approach affects performance.

2. Approach

Our approach consists of a pre-training stage followed by a fine-tuning stage. In pre-training, we explore both the auto-regressive and BERT objectives. We also apply the sequence Transformer architecture to predict pixels instead of language tokens.

One way to measure representation quality is to fine-tune for image classification. Fine-tuning adds a small classification head to the model, used to optimize a classification objective and adapts all weights. Pre-training can be viewed as a favorable initialization or as a regularizer when used in combination with early stopping (Erhan et al., 2010).

Another approach for measuring representation quality uses the pre-trained model as a feature extractor. In particular, given labeled examples (X, Y), the model is applied to X to produce features f_X. Then, a linear classifier is trained on (f_X, Y). Linear probing captures the intuition that good features should linearly separate the classes of transfer tasks. Furthermore, linear probes help disentangle feature quality from model architecture: in fine-tuning, one model may outperform another because its architecture is more suited for the downstream task rather than because of better pre-training.

We begin this section by defining the auto-regressive and BERT objectives in the context of images. Next, we outline implementation details for our transformer decoder. Finally, we describe how the transformer is used for fine-tuning and how features are extracted for linear probes.

2.1. Pre-training

Given an unlabeled dataset X consisting of high dimensional data $x = (x_1, ..., x_n)$, we can pick a permutation π of the set $[1, n]$ and model the density $p(x)$ auto-regressively as follows:

$$p(x) = \prod_{i=1}^{n} p(x_{\pi_i} | x_{\pi_1}, ..., x_{\pi_{i-1}}, \theta)$$

When working with images, we pick the identity permutation $\pi_i = i$ for $1 \leq i \leq n$, also known as raster order. We train our model by minimizing the negative log-likelihood of the data:

$$L_{AR} = \mathop{\mathbb{E}}_{x \sim X} [-\log p(x)]$$

We also consider the BERT objective, which samples a sub-sequence $M \subset [1, n]$ such that each index i independently has probability 0.15 of appearing in M. We call M the BERT mask, and we train our model by minimizing the negative log-likelihood of the "masked" elements x_M conditioned on the "unmasked" ones $x_{[1,n] \setminus M}$:

$$L_{BERT} = \mathop{\mathbb{E}}_{x \sim X} \mathop{\mathbb{E}}_{M} \sum_{i \in M} \left[-\log p \left(x_i | x_{[1,n] \setminus M} \right) \right]$$

In pre-training, we pick one of L_{AR} or L_{BERT} and minimize the loss over our pre-training dataset.

2.2. Architecture

The transformer decoder takes an input sequence $x_1, ..., x_n$ of discrete tokens and produces a d-dimensional embedding for each position. The decoder is realized as a stack of L blocks, the l-th of which produces an intermediate embedding $h_1^l, ..., h_n^l$ also of dimension d. We use the GPT-2

(Radford et al., 2019) formulation of the transformer decoder block, which acts on an input tensor h^l as follows:

$$n^l = \text{layer\_norm}(h^l)$$
$$a^l = h^l + \text{multihead\_attention}(n^l)$$
$$h^{l+1} = a^l + \text{mlp}(\text{layer\_norm}(a^l))$$

In particular, layer norms precede both the attention and mlp operations, and all operations lie strictly on residual paths. We find that such a formulation allows us to scale the transformer with ease.

The only mixing across sequence elements occurs in the attention operation, and to ensure proper conditioning when training the AR objective, we apply the standard upper triangular mask to the $n \times n$ matrix of attention logits. When using the BERT objective, no attention logit masking is required: after applying content embeddings to the input sequence, we zero out the positions in M.

Additionally, since we learn independent position embeddings for each sequence element, our BERT model has no positional inductive biases (i.e. it is permutation invariant). Put another way, any spatial relationships between positions must be learned by the model at train time. This is not entirely true for the AR model, as choosing the raster order also fixes a prespecified ordering of the conditionals. Nevertheless, permutation invariance is a property in strong contrast to convolutional neural networks, which incorporate the inductive bias that features should arise from spatially proximate elements.

Following the final transformer layer, we apply a layer norm $n^L = \text{layer\_norm}(h^L)$, and learn a projection from n^L to logits parameterizing the conditional distributions at each sequence element. When training BERT, we simply ignore the logits at unmasked positions.

2.3. Fine-tuning

When fine-tuning, we average pool n^L across the sequence dimension to extract a d-dimensional vector of features per example:

$$f^L = \langle n_i^L \rangle_i$$

We learn a projection from f^L to class logits, which we use to minimize a cross entropy loss L_{CLF}.

While fine-tuning on L_{CLF} yields reasonable downstream performance, we find empirically that the joint objective

$$L_{GEN} + L_{CLF}$$

$L_{GEN} \in \{L_{AR}, L_{BERT}\}$ works even better. Similar findings were reported by Radford et al. (2018).

2.4. Linear Probing

Extracting fixed features for linear probing follows a similar procedure to fine-tuning, except that average pooling is not always at the final layer:

$$f^l = \langle n_i^l \rangle_i$$

where $0 \leq l \leq L$. We will show in the experiments section that the best features often lie in the middle of the network. As in fine-tuning, we project these intermediate features to produce class logits. Because we view the features as fixed when linear probing, this projection contains the only trainable weights, so we can only optimize L_{CLF}.

3. Methodology

Although supervised pre-training is the dominant paradigm for image classification, curating large labeled image datasets is both expensive and time consuming. Instead of further scaling up labeling efforts, we can instead aspire to learn general purpose representations from the much larger set of available unlabeled images and fine-tune them for classification. We investigate this setting using ImageNet as a proxy for a large unlabeled corpus, and small classic labeled datasets (CIFAR-10, CIFAR-100, STL-10) as proxies for downstream tasks.

Even in cases where labels are available, unsupervised or self-supervised pre-training can still provide benefits in data efficiency or on fine-tuning speed. We investigate this setting by pre-training on ImageNet without labels and then fine-tuning or linear probing with labels.

3.1. Dataset and Data Augmentation

We use the ImageNet ILSVRC 2012 training dataset, splitting off 4% as our experimental validation set and report results on the ILSVRC 2012 validation set as our test set. For CIFAR-10, CIFAR-100 and STL-10, we split off 10% of the provided training set instead. We ignore the provided unlabeled examples in STL-10, which constitute a subset of ImageNet.

When pre-training or fine-tuning on ImageNet, we make use of lightweight data augmentation. First, we randomly resize the image such that the shorter sidelength is in the range $[256, 384]$. Next, we take a random 224×224 crop. When evaluating on ImageNet, we resize the image such that the shorter sidelength is 224, and use the single 224×224 center crop.

When full-network fine-tuning on CIFAR-10 and CIFAR-100, we use the augmentation popularized by Wide Residual Networks: 4 pixels are reflection padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip (Zagoruyko & Komodakis, 2016).

Once optimal hyperparameters are found, we fold our experimental validation set back into the training set, retrain the model, and report numbers on the respective test set.

3.2. Context Reduction

Because the memory requirements of the transformer decoder scale quadratically with context length when using dense attention, we must employ further techniques to reduce context length. If we naively trained a transformer on a sequence of length $224^2 \times 3$, our attention logits would be tens of thousands of times larger than those used in language models and even a single layer would not fit on a GPU. To deal with this, we first resize our image to a lower resolution, which we call the input resolution (IR). Our models have an IR of $32^2 \times 3$, $48^2 \times 3$, $96^2 \times 3$, or $192^2 \times 3$.

An IR of $32^2 \times 3$ is still quite computationally intensive. While even lower IRs are tempting, prior work has demonstrated human performance on image classification begins to drop rapidly below this size (Torralba et al., 2008). When using an IR of $32^2 \times 3$ or $48^2 \times 3$, we instead further reduce context size by a factor of 3 by clustering (R, G, B) pixel values using k-means with $k = 512$. A similar approach was applied to spatial patches by Ranzato et al. (2014). We call the resulting context length (32^2 or 48^2) the model resolution (MR). Note that this reduction breaks permutation invariance of the color channels, but keeps the model spatially invariant.

To push performance on ImageNet linear probes, we also work with IRs of $96^2 \times 3$ or $192^2 \times 3$. Here, only clustering pixels produces a context that is still too large. Using a VQ-VAE (van den Oord et al., 2017) with a latent grid size of 48^2, we can downsample our images and stay at a MR of 48^2. We choose a latent vocabulary size of 4096, the lowest size at which we do not observe reconstruction artifacts. Our clustering of (R, G, B) values can also be interpreted as the training of a VQ-VAE with an identity encoder and decoder.

For the VQ-VAE architecture, we choose a small encoder and decoder pair ($<$ 1M parameters) to aid the sequence Transformer in modeling latent codes. Although downsampling with VQ-VAE destroys spatial permutation invariance, the receptive field for a latent code is only 16×16 for an IR of $96^2 \times 3$ and 34×34 for an IR of $192^2 \times 3$. Because the encoder is so small, information stays relatively local.

3.3. Model

Our largest model, iGPT-L, is essentially identical to GPT-2. Both models contain $L = 48$ layers, but we use an embedding size of $d = 1536$ (vs 1600), resulting in a slightly reduced parameter count (1.4B vs 1.5B). We use the same model code as GPT-2, except that we initialize weights in the layer-dependent fashion as in Sparse Transformer (Child et al., 2019) and zero-initialize all projections producing logits.

We also train iGPT-M, a 455M parameter model with $L = 36$ and $d = 1024$ and iGPT-S, a 76M parameter model with $L = 24$ and $d = 512$ to study the effect of model capacity on representation quality in a generative model.

3.4. Training

When pre-training, we use a batch size of 128 and train for 1000000 iterations using Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We sequentially try the learning rates 0.01, 0.003, 0.001, 0.0003, ..., stopping at the first local minimum. The learning rate is warmed up for one epoch, and then decays to 0 following a cosine schedule. No dropout is used.

When fine-tuning, we use the same batch size and Adam hyperparameters. Here, we do not employ a cosine schedule, and early stop once we reach the maximum validation accuracy. Again, no dropout is used.

When running a linear probe on ImageNet, we follow recent literature and use SGD with momentum 0.9 and a high learning rate (we try the values 30, 10, 3, ... in the manner described above) (He et al., 2019). We train for 1000000 iterations with a cosine learning rate schedule. Finally, when running a linear probe on CIFAR-10, CIFAR-100, or STL-10, we use the L-BFGS algorithm for consistency with prior results (Pedregosa et al., 2011).

4. Experiments and Results

We begin with experiments and results from the autoregressive formulation of iGPT. Comparisons with the BERT formulation appear in Section 4.6.

4.1. What Representation Works Best in a Generative Model Without Latent Variables?

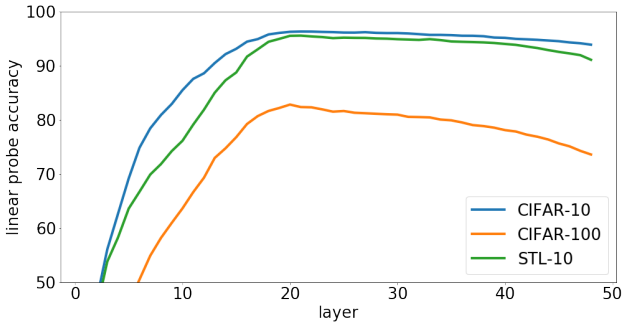

Figure 2. Representation quality heavily depends on the layer from which we extract features. In contrast with supervised models, the best representations for these generative models lie in the middle of the network. We plot this unimodal dependence on depth showing linear probes on CIFAR-10, CIFAR-100, and STL-10.

In supervised pre-training, representation quality tends to increase monotonically with depth, such that the best representations lie at the penultimate layer (Zeiler & Fergus, 2014). Indeed, since a linear layer produces logits from pre-logits, a high performing classifier necessarily achieves high

accuracy on a linear probe of its pre-logits. If a downstream task also involves classification, it is empirically validated that penultimate features perform well.

With generative pre-training, it is not obvious whether a task like pixel prediction is relevant to image classification. This suggests that the penultimate layer of a model trained for pixel prediction might not produce the most useful representations for classification. Latent variable models such as VAEs can avoid this issue by explicitly learning a representation of the input data, but deep autoregressive generative models have the same width and connectivity pattern at every layer. Our first experiment studies how representation quality varies over one set of candidate representations: different layers of a generative model. We observe a very different behavior from supervised learning: representations first improve as a function of depth, and then, starting around the middle layer, begin to deteriorate until the penultimate layer (Figure 2).

This behavior potentially suggests that these generative models operate in two phases. In the first phase, each position gathers information from its surrounding context in order to build a more global image representation. In the second phase, this contextualized input is used to solve the conditional next pixel prediction task. This could resemble the behavior of encoder-decoder architectures common across deep learning, but learned within a monolithic architecture via a pre-training objective.

Consequently, when evaluating a generative model with a linear probe, it is important to search for the best layer. Taking the final layer on CIFAR-10 decreases performance by 2.4%, the difference between a baseline and a state-of-the-art result. In all settings, we find that the dependence of representation quality on depth is strongly unimodal.

4.2. Better Generative Models Learn Better Representations

Using the linear probe as a tool for measuring representation quality, we investigate whether better generative models (as measured by log-prob on held-out data) also learn better representations.

In Figure 3, we see that as validation loss on the autoregressive objective decreases throughout training, linear probe accuracy increases as well. This trend holds across several model capacities, with higher capacity models achieving better validation losses. This highlights the importance of scale for our approach. Note that for a given validation loss value, bigger models also perform better.

4.3. Linear Probes on CIFAR and STL-10

In addition to CIFAR-10, we also evaluate linear probes on CIFAR-100 and STL-10 (Figure 2) to check whether the learned representations are useful across multiple datasets. For this evaluation setting, we achieve state-of-the-art across the entire spectrum of pre-training approaches (Table 1). For example, on CIFAR-10, our model achieves 96.3%, outperforming both AMDIM-L (pre-trained on ImageNet without labels) and a ResNet-152 (pre-trained on ImageNet with labels). In fact, on all three datasets a linear classifier fit to the representations of iGPT-L outperforms the end-to-end supervised training of a WideResNet baseline.

Note that our model is trained at the same input resolution (IR) as CIFAR, whereas models trained at the standard ImageNet IR may experience distribution shock upon linear evaluation. As a counterpoint, though STL-10 has an IR of $96^2 \times 3$, we still outperform AMDIM-L when we downsample to $32^2 \times 3$ before linear probing. We also note that fine-tuning should allow models trained at high IR to adjust

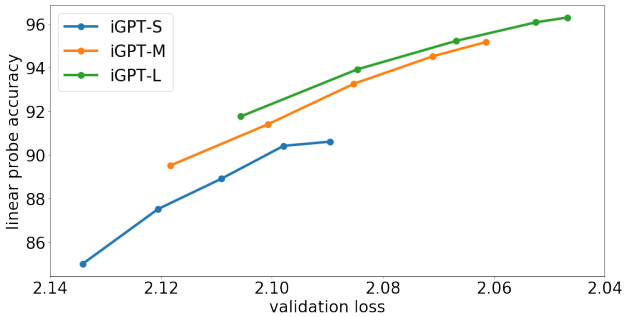

Figure 3. Plot of representation quality as a function of validation generative loss. Each line tracks a model throughout generative pre-training: the dotted markers denote checkpoints at steps 65K, 131K, 262K, 524K, and 1000K. The positive slope suggests a link between improved generative performance and improved representation quality. Larger models produce better representations than smaller ones both at the end of training and at the same value of validation loss.

Table 1. Comparing linear probe accuracies between our models and state-of-the-art models utilizing unsupervised ImageNet transfer or supervised ImageNet transfer.

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
AMDIM-L	91.2	√	
ResNet-152	94		√
iGPT-L	96.3	√	
CIFAR-100			
AMDIM-L	70.2	√	
ResNet-152	78		√
iGPT-L	82.8	√	
STL-10			
AMDIM-L	94.2	√	
iGPT-L (IR $32^2 \cdot 3$)	95.5	√	
iGPT-L (IR $96^2 \cdot 3$)	97.1	√	

Table 2. Comparing linear probe accuracies between our models and state-of-the-art self-supervised models. A blank input resolution (IR) corresponds to a model working at standard ImageNet resolution. We report the best performing configuration for each contrastive method, finding that our models achieve comparable performance.

Method	IR	Params (M)	Features	Acc
Rotation	orig.	86	8192	55.4
iGPT-L	$32^2 \cdot 3$	1362	1536	60.3
BigBiGAN	orig.	86	8192	61.3
iGPT-L	$48^2 \cdot 3$	1362	1536	65.2
AMDIM	orig.	626		68.1
MoCo	orig.	375	8192	68.6
iGPT-L	$192^2 \cdot 3$	1362	16896	69.0
CPC v2	orig.	303	8192	71.5

Table 3. Comparing fine-tuning performance between our models and state-of-the-art models utilizing supervised ImageNet transfer. We also include AutoAugment, the best performing model trained end-to-end on CIFAR. Table results: AutoAugment (Cubuk et al., 2019), GPipe (Huang et al., 2019), EfficentNet (Tan & Le, 2019)

Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
AutoAugment	98.5		
GPipe	99.0		√
iGPT-L	99.0	√	
CIFAR-100			
iGPT-L	88.5	√	
AutoAugment	89.3		
EfficientNet	91.7		√

to low resolution input.

4.4. Linear Probes on ImageNet

Recently, there has been a resurgence of interest in unsupervised and self-supervised learning on ImageNet, evaluated using linear probes on ImageNet. This is a particularly difficult setting for us, since we cannot efficiently train at the standard ImageNet input resolution (IR). Indeed, with a model resolution (MR) of 32^2, we achieve only 60.3% best-layer linear probe accuracy. As with CIFAR-10, scale is critical to our approach: iGPT-M achieves 54.5% accuracy and iGPT-S achieves 41.9% accuracy.

The first obvious optimization is to increase MR while staying within accelerator memory limits. With a MR of 48^2, we achieve a best-layer accuracy of 65.2% using 1536 features. However, since contrastive methods report their best results on 8192 features, we would ideally evaluate iGPT with an embedding dimension 8192 for comparison. Training such a model is prohibitively expensive, so we instead concatenate features from multiple layers as an approximation. Our features tend to be correlated across layers, so we find that we need more of them to be competitive. If we concatenate features from 11 layers centered at the best single layer, we achieve an accuracy of 67.3% using 16896 features. Note that we achieve this accuracy both working at low resolution and without 2D structure.

To push performance even further, we use the VQ-VAE data preprocessing step described in section 3.2, sacrificing local spatial invariance. Interestingly, when training on an IR of $192^2 \times 3$ and a MR of 48^2, the best-layer accuracy remains unchanged at 65.3%. However, the benefit of working with a higher IR is realized when we concatenate 11 layers centered at the best single layer, giving us an accuracy of 69.0%, competitive with recent contrastive learning approaches (Table 2).

Because best-layer accuracy is insensitive to IR given a fixed MR, a finding also observed by Sandler et al. (2019), we conjecture that training on longer contexts (larger MRs) will yield the largest improvements in linear probe accuracy. We also suspect that features from wider models will outperform concatenated layerwise features, which tend to be correlated in residual networks (Kornblith et al., 2019).

4.5. Full Fine-tuning

To achieve even higher accuracy on downstream tasks, we adapt the entire model for classification through fine-tuning. Building off of the previous analysis, we tried attaching the classification head to the layer with the best representations. Though this setup trains faster than one with the head attached at the end, the latter is able to leverage greater model depth and eventually outperforms.

On CIFAR-10, we achieve 99.0% accuracy and on CIFAR-100, we achieve 88.5% accuracy after fine-tuning. We outperform AutoAugment, the best supervised model on these datasets, though we do not use sophisticated data augmentation techniques. In fact, 99.0% ties GPipe, the best model which pre-trains using ImageNet labels.

On ImageNet, we achieve 66.3% accuracy after fine-tuning at MR 32^2, a bump of 6% over linear probing. When fine-tuning at MR 48^2, we achieve 72.6% accuracy, with a similar 7% bump over linear probing. However, our models still slightly underperform Isometric Neural Nets (Sandler et al., 2019), which achieves 70.2% at an IR of $28^2 \times 3$.

Finally, as a baseline for ImageNet fine-tuning, we train the classification objective from a random initialization. At MR 48^2, a model with tuned learning rate and dropout achieves 53.2% after 18 epochs, 19.4% worse than the pre-trained model. Comparatively, the pre-trained model is much quicker to fine-tune, achieving the same 53.2% loss in roughly a single epoch.

When fine-tuning, it is important to search over learning rates again, as the optimal learning rate on the joint training objective is often an order of magnitude smaller than that for pre-training. We also tried regularizing with dropout, though we did not observe any clear benefits. It is easy to overfit the classification objective on small datasets, so we employ early stopping based on validation accuracy.

4.6. BERT

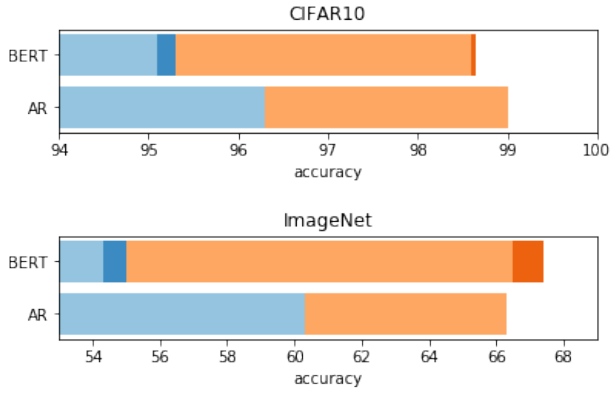

Figure 4. Comparison of auto-regressive pre-training with BERT pre-training using iGPT-L at an input resolution of $32^2 \times 3$. Blue bars display linear probe accuracy and orange bars display fine-tune accuracy. Bold colors show the performance boost from ensembling BERT masks. We see that auto-regressive models produce much better features than BERT models after pre-training, but BERT models catch up after fine-tuning.

Given the success of BERT in language, we train iGPT-L at an input resolution of $32^2 \times 3$ and a model resolution of 32^2 (Figure 4). On CIFAR-10, we observe that linear probe accuracy at every layer is worse than that of the auto-regressive model, with best-layer performance more than 1% lower. Best-layer accuracy on ImageNet is 6% lower.

However, during fine-tuning, BERT makes up much of this gap. A fully fine-tuned CIFAR-10 model achieves 98.6% accuracy, only 0.4% behind its auto-regressive counterpart, while a fully fine-tuned ImageNet model achieves 66.5%, slightly surpassing auto-regressive performance.

Finally, because inputs to the BERT model are masked at training time, we must also mask them at evaluation time to keep inputs in-distribution. This masking corruption may hinder the BERT model's ability to correctly predict image classes. Therefore, we also try an evaluation scheme where we sample 5 independent masks for each input and take the modal prediction, breaking ties at random. In this setting, CIFAR-10 results are largely unchanged, but on ImageNet, we gain almost 1% on our linear probes and fine-tunes.

Table 4. Comparing performance on low-data CIFAR-10. By leveraging many unlabeled ImageNet images, iGPT-L is able to outperform methods such as Mean Teacher (Tarvainen & Valpola, 2017) and MixMatch (Berthelot et al., 2019) but still underperforms the state of the art methods (Xie et al., 2019; Sohn et al., 2020). Our approach to semi-supervised learning is very simple since we only fit a logistic regression classifier on iGPT-L's features without any data augmentation or fine-tuning - a significant difference from specially designed semi-supervised approaches. Other results reported from FixMatch (Sohn et al., 2020).

Model	40 labels	250 labels	4000 labels
Mean Teacher		32.3 ± 2.3	9.2 ± 0.2
MixMatch	47.5 ± 11.5	11.0 ± 0.9	6.4 ± 0.1
iGPT-L	26.8 ± 1.5	12.4 ± 0.6	5.7 ± 0.1
UDA	29.0 ± 5.9	8.8 ± 1.1	4.9 ± 0.2
FixMatch RA	13.8 ± 3.4	5.1 ± 0.7	4.3 ± 0.1
FixMatch CTA	11.4 ± 3.4	5.1 ± 0.3	4.3 ± 0.2

4.7. Low-Data CIFAR-10 Classification

Evaluations of unsupervised representations often reuse supervised learning datasets which have thousands to millions of labeled examples. However, a representation which has robustly encoded a semantic concept should be exceedingly data efficient. As inspiration, we note that humans are able to reliably recognize even novel concepts with a single example (Carey and Bartlett 1978). This motivates evaluating performance in a low-data regime as well. It is also a more realistic evaluation setting for the potential practical usefulness of an approach since it better matches the common real-world scenario of an abundance of raw data but a lack of labels.

In contrast with recent approaches for low-data classification, we do not make use of pseudo-labeling or data augmentation. Instead, we work directly on a subset of the raw supervised dataset, extracting features using our pre-trained model, and training a linear classifier on those features.

As is standard in the low-data setting, we sample 5 random subsets and report mean and standard deviation accuracies (Table 4). On CIFAR-10, we find that with 4 labels per class, we achieve 73.2% accuracy outperforming MixMatch with much lower variance between runs and with 25 labels per class, we achieve 87.6% accuracy, though still significantly lower than the state of the art, FixMatch.

Although we have established that large models are necessary for producing good representations, large models are also difficult to fine-tune in the ultra-low data regime. Indeed, we find that iGPT-L quickly memorizes a 40-example training set and fails to generalize well, achieving only 42.1% accuracy. We expect adapting recent approaches to semi-supervised learning will help in this setting.

5. Related Work

Many generative models have been developed and evaluated for their representation learning capabilities. Notably, GANs (Goodfellow et al., 2014; Radford et al., 2015; Donahue et al., 2016) and VAEs (Kingma & Welling, 2013; Kingma et al., 2014; Higgins et al., 2017) have been well-studied.

As of yet, most generative model based approaches have not been competitive with supervised and self-supervised methods in the image domain. A notable exception is BigBiGAN (Donahue & Simonyan, 2019) which first demonstrated that sufficiently high fidelity generative models learn image representations which are competitive with other self-supervised methods.

Many self-supervised approaches focus on designing auxiliary objectives which support the learning of useful representations without attempting to directly model the input data. Examples include surrogate classification (Dosovitskiy et al., 2015), jigsaw puzzle solving (Noroozi & Favaro, 2016), and rotation prediction (Gidaris et al., 2018). A cluster of similar approaches based on contrastive losses comparing various views and transformations of input images have recently driven significant progress in self-supervised learning (Hjelm et al., 2018; Bachman et al., 2019; Tian et al., 2019).

Among contrastive approaches, our work is most similar to Contrast Predictive Coding (Oord et al., 2018) which also utilizes a autoregressive prediction objective, but in a learned latent space, and to Selfie (Trinh et al., 2019) which trains a bidirectional self-attention architecture on top of a standard convolutional network to differentiate correct vs wrong patches.

Our work is directly inspired by the success of generative pre-training methods developed for Natural Language Processing. These methods predict some parts of a piece of text conditioned on other parts. Our work explores two training objectives in this framework, autoregressive prediction as originally explored for modern neural sequence models by Dai & Le (2015), and a denoising objective, similar to BERT (Devlin et al., 2018). The context in-painting approach of Pathak et al. (2016) also explores pre-training by predicting corruptions but predicts large regions of high-resolution images.

Kolesnikov et al. (2019); Goyal et al. (2019) conducted rigorous investigations of existing self-supervised methods. Several of our findings are consistent with their results, including the benefits of scale and the non-monotonic performance of representations with depth in certain architectures.

Expressive autoregressive models tractably optimizing likelihood were first applied to images by Uria et al. (2013) and popularized by Oord et al. (2016) serving for the basis of several papers similarly adapting transformers to the problem of generative image modeling (Parmar et al., 2018; Child et al., 2019).

Ke et al. (2018) introduced the pixel-by-pixel CIFAR10 task and first benchmarked the performance of a 1D sequence transformer on a competitive image classification dataset. Rives et al. (2019) similarly investigates whether the recent success of unsupervised pre-training in NLP applies to other domains, observing promising results on protein sequence data.

6. Discussion and Conclusion

Our results suggest that generative image modeling continues to be a promising route to learn high-quality unsupervised image representations. Simply predicting pixels learns state of the art representations for low resolution datasets. In high resolution settings, our approach is also competitive with other self-supervised results on ImageNet.

However, our experiments also demonstrate several areas for improvement. We currently model low resolution inputs with self-attention. By comparison, most other self-supervised results use CNN based encoders that easily work with high resolution images. It is not immediately obvious how to best bridge the gap between performant autoregressive and discriminative models. Additionally, we observed that our approach requires large models in order to learn high quality representations. iGPT-L has 2 to 3 times as many parameters as similarly performing models on ImageNet and uses more compute.

Although dense self-attention was a deliberate choice for this work due to it being domain agnostic and widely used in NLP, it becomes very memory and computationally expensive due to its quadratic scaling with sequence length. We mitigated this via the context reduction techniques discussed in section 3.2 but it is still a significant limitation. Future work could instead address this via architectural changes by exploring more efficient self-attention approaches. Several promising techniques have recently been developed such as local 2D relative attention (Bello et al., 2019; Ramachandran et al., 2019), sparse attention patterns (Child et al., 2019), locality sensitive hashing (Kitaev et al., 2020), and multiscale modeling (Menick & Kalchbrenner, 2018).

Finally, our results, considered together with Donahue & Simonyan (2019), suggest revisiting the representation learning capabilities of other families of generative models such as flows (Dinh et al., 2014; Kingma & Dhariwal, 2018) and VAEs in order to study whether they show similarly competitive representation learning capabilities.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pp. 15509–15519, 2019.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3286–3295, 2019.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Cubuk, E., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data, 2019.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pp. 3079–3087, 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pp. 10541–10551, 2019.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Dosovitskiy, A., Fischer, P., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why does unsupervised pretraining help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pp. 2214–2224, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6391–6400, 2019.

Graves, A. and Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pp. 1764–1772, 2014.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2017.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18 (7):1527–1554, 2006.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Huang, P.-S., Avron, H., Sainath, T. N., Sindhwani, V., and Ramabhadran, B. Kernel methods match deep neural networks on timit. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 205–209. IEEE, 2014.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, pp. 103–112, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Ke, N. R., GOYAL, A. G. A. P., Bilaniuk, O., Binas, J., Mozer, M. C., Pal, C., and Bengio, Y. Sparse attentive backtracking: Temporal credit assignment through reminding. In *Advances in neural information processing systems*, pp. 7640–7651, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1920–1929, 2019.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*, 2019.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 29–37, 2011.

Lasserre, J. A., Bishop, C. M., and Minka, T. P. Principled hybrids of generative and discriminative models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pp. 87–94. IEEE, 2006.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pp. 609–616, 2009.

May, A., Garakani, A. B., Lu, Z., Guo, D., Liu, K., Bellet, A., Fan, L., Collins, M., Hsu, D., Kingsbury, B., et al. Kernel approximation methods for speech recognition. *arXiv preprint arXiv:1701.03577*, 2017.

Menick, J. and Kalchbrenner, N. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. *arXiv preprint arXiv:1812.01608*, 2018.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

Mohamed, A.-r., Dahl, G., and Hinton, G. Deep belief networks for phone recognition. 2009.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paine, T. L., Khorrami, P., Han, W., and Huang, T. S. An analysis of unsupervised pre-training in light of recent advances. *arXiv preprint arXiv:1412.6597*, 2014.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., and Tran, D. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, pp. 622803, 2019.

Sandler, M., Baccash, J., Zhmoginov, A., and Howard, A. Non-discriminative data or weak model? on the relative importance of data and model resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204, 2017.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

Trinh, T. H., Luong, M.-T., and Le, Q. V. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*, 2019.

Uria, B., Murray, I., and Larochelle, H. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pp. 2175–2183, 2013.

van den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Table 5. Learning rates used for each model, objective, and input resolution (IR) combination.

Model	Objective	IR	Learning Rate
iGPT-S	auto-regressive	$32^2 \times 3$	0.003
iGPT-M	auto-regressive	$32^2 \times 3$	0.003
iGPT-L	auto-regressive	$32^2 \times 3$	0.001
iGPT-L	auto-regressive	$48^2 \times 3$	0.01
iGPT-L	auto-regressive	$96^2 \times 3$	0.003
iGPT-L	auto-regressive	$192^2 \times 3$	0.01
iGPT-S	BERT	$32^2 \times 3$	0.01
iGPT-M	BERT	$32^2 \times 3$	0.003
iGPT-L	BERT	$32^2 \times 3$	0.001

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

A. Experimental details

A.1. Hyperparameters

In Table 5, we present the learning rates used to train each model in the paper. When using too high a learning rate, we observe an irrecoverable loss spike early on in training. Conversely, with too low a learning rate, training is stable but loss improves slowly and eventually underperforms. As we increase model size, the irrecoverable loss spike occurs at even lower learning rates. This motivates our procedure of sequentially searching learning rates from large to small and explains why larger models use lower learning rates than smaller models at fixed input resolution.

We used an Adam β_2 of 0.95 instead of the default 0.999 because the latter causes loss spikes during training. We did not use weight decay because applying a small weight decay of 0.01 did not change representation quality.

On iGPT-S, we found small gains in representation quality from using float32 instead of float16, from untying the token embedding matrix and the matrix producing token logits, and from zero initializing the matrices producing token and class logits. We applied these settings to all models.

When training BERT models, one additional hyperparameter is the masking probability, set to 15% in Devlin et al. (2018). We also tried higher masking rates of 20%, 25%, 30%, and 35%, finding that 20% matched the performance of 15%, though higher probabilities decreased performance.

A.2. VQ-VAE

Our VQ-VAE models largely follow the original approach in (van den Oord et al., 2017). Each encoder block consists of a downsampling convolution, ReLU, and a residual network. The decoder block mirrors this with a residual network, ReLU, and an upsampling transposed convolution. All resampling convolutions use stride 2×2 and kernel size 4×4. To get 48^2 MR from $96^2 \times 3$ IR, the encoder and decoder each use one of the blocks described above. For compressing $192^2 \times 3$ IR to 48^2 MR, two stacked blocks are needed. Our residual networks have the same architecture as the one in (van den Oord et al., 2017), but use 32 hidden units in the residual branch. For resampling convolutions and the VQ codebook, we use 64 channels.

While our autoencoders are tiny (fewer than 200K parameters), foreground reconstruction quality from our models is similar to that of much larger (40M parameter) autoencoders as long as a large codebook is used. In fact, we found that increasing the autoencoder size results in codes that are harder for the prior to compress. We use a vocab size of 4096, which puts more parameters (262K) in the VQ embeddings than the autoencoders themselves.

We experimented with L_1 and L_2 reconstruction losses, and found that L_2 reconstructs textures marginally better as shown in Figure 5. After rescaling reconstruction and commitment losses to unit variance, we chose a commitment cost coefficient $\beta_2 = 0.02$ based on visual inspection of reconstructed images. We used Adam (Kingma & Ba, 2014) with a learning rate of 0.0001 to learn the autoencoder weights.

Following (van den Oord et al., 2017), we updated the codebook using an exponential moving average (EMA). While EMA is fairly robust, we still observed a small degree of codebook collapse with a large vocabulary size. To combat this, we considered a VQ code dead if its usage fell below 10% of its expected usage ($(\text{batch size}) \times (\text{MR})/(\text{codebook size}) = 128 \times 48^2/4096 = 72$), and revived it to take on a value near a live code.

B. Samples

Although our goal is not explicitly to produce high quality samples, training an auto-regressive objective gives us this capability. Note that we cannot use class conditioning to improve sample quality since we do not have access to labels during pre-training. Below, we show class-unconditional samples from iGPT-L with IR $32^2 \times 3$ (Figure 6) and with IR $96^2 \times 3$ (Figure 7).

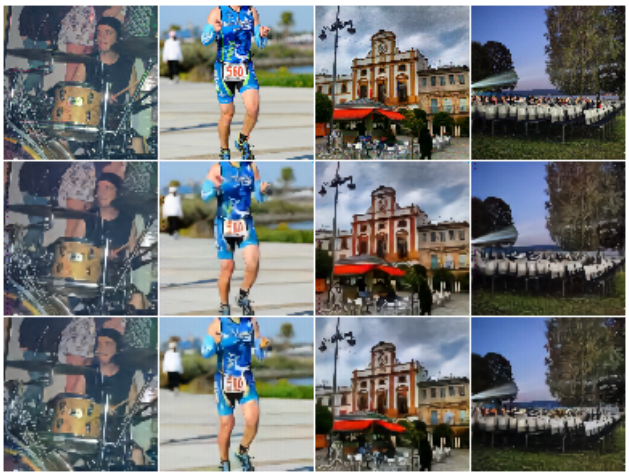

Figure 5. We compressed $96^2 \times 3$ IR to 32^2 MR using VQ-VAE with L_1 (middle) and L_2 (bottom) losses. Both reconstructions are generally almost as good as the groundtruth (top), but L_1 tends to produce slightly more diffuse images.

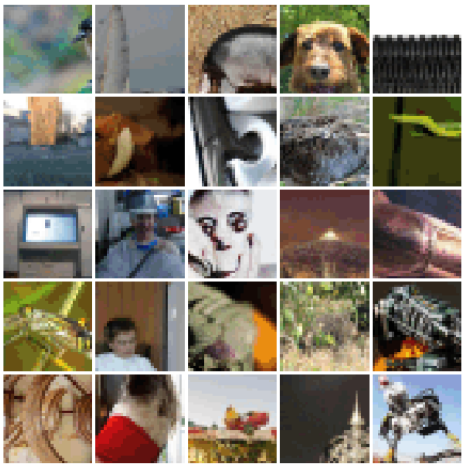

Figure 6. Class-unconditional samples at temperature 1.0 from iGPT-L trained on input images of resolution $32^2 \times 3$.

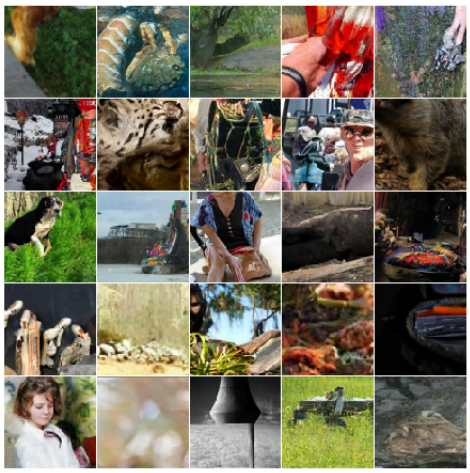

Figure 7. Class-unconditional samples at temperature 0.98 from iGPT-L trained on input images of resolution $96^2 \times 3$.