GIVT: Generative Infinite-Vocabulary Transformers

Michael Tschannen, Cian Eastwood*, and Fabian Mentzer°

Google DeepMind {tschannen, mentzer}@google.com

Abstract. We introduce Generative Infinite-Vocabulary Transformers (GIVT) which generate vector sequences with real-valued entries, instead of discrete tokens from a finite vocabulary. To this end, we propose two surprisingly simple modifications to decoder-only transformers: 1) at the input, we replace the finite-vocabulary lookup table with a linear projection of the input vectors; and 2) at the output, we replace the logits prediction (usually mapped to a categorical distribution) with the parameters of a multivariate Gaussian mixture model. Inspired by the image-generation paradigm of VQ-GAN and MaskGIT, where transformers are used to model the discrete latent sequences of a VQ-VAE, we use GIVT to model the unquantized real-valued latent sequences of a β-VAE. In class-conditional image generation GIVT outperforms VQ-GAN (and improved variants thereof) as well as MaskGIT, and achieves performance competitive with recent latent diffusion models. Finally, we obtain strong results outside of image generation when applying GIVT to panoptic segmentation and depth estimation with a VAE variant of the UViM framework.

Keywords: Image generation · Latent sequence modeling · Soft tokens

1 Introduction

After becoming the dominant architecture in natural language processing shortly after their introduction, Transformers [72] have also recently become very popular in computer vision [18, 40, 63]. Dosovitskiy *et al.* [18] showed that by splitting images into sequences of patches, linearly embedding those patches, and then feeding the resulting sequence of features to a transformer encoder leads to powerful image classifiers that outperform CNN-based architectures at large model and data scale. This strategy is now standard for many discriminative vision task including classification [18], detection [40], and segmentation [63]. It is less obvious how to apply generative transformer decoders to image *generation* since they were designed to consume and predict *discrete tokens from some fixed, finite vocabulary.* Such a structure naturally fits natural language, for which decoder-only models enable powerful sequential generative modeling and efficient training [52, 72].

*Work done as Student Researcher at GDM. °Significant technical contributions. Code and model checkpoints: https://github.com/google-research/big_vision.

Fig. 1: Selected 512×512 samples from GIVT-Causal-L for 10 ImageNet classes (130, 130, 138, 144, 933, 145, 360, 207, 829, 248).

To harness these capabilities for images, recent works [6,7,20,39,46,54] have employed a two-stage approach which first trains a Vector-Quantized Variational Autoencoder (VQ-VAE) [49] to map images to a sequence of discrete tokens, and then trains a transformer decoder to model the latent discrete-token distribution. An advantage of such a VQ-VAE-based image tokenization is that it enables interleaved multimodal generative models, simply by concatenating the vocabularies of the different modalities including text and images [1,2,29]. However, this approach also has several issues. First, the non-continuous nature of VQ requires differentiable approximations to enable stochastic gradient-based optimization [49]. Second, a VQ-VAE with a small vocabulary can make the latent modeling easy but also makes the latent code less informative, which prevents control of the low-level details in image generation, and impacts quality when using the tokens for dense prediction [33,42] or low-level discriminative tasks [1,29]. A large vocabulary, on the other hand, can lead to low vocabulary utilization [46] so that high-fidelity VQ-VAE setups typically rely on a range of advanced techniques, such as entropy losses [7] or codebook-splitting [33]. Furthermore, large vocabularies lead to correspondingly large embedding matrices and hence memory consumption, which can be an issue particularly in multimodal contexts.

In this work, we show—to our knowledge for the first time—**how to completely remove quantization** from generative transformers for visual data. Indeed, practitioners seem to agree that this would be hardly possible, since transformer decoders are strongly linked to discrete representations in many heads. **Surprisingly, we not only show that simple modifications enable transformer decoders to directly generate sequences of unquantized vectors, but also that this approach leads to better image generation quality and representation learning capabilities than VQ-based approaches.** We call such transformers *Generative Infinite-Vocabulary Transformer* (GIVT).[1] Concretely, we make two changes compared to the standard

[1] We discuss the relation between continuous latents and infinite vocabulary in Sec. A.

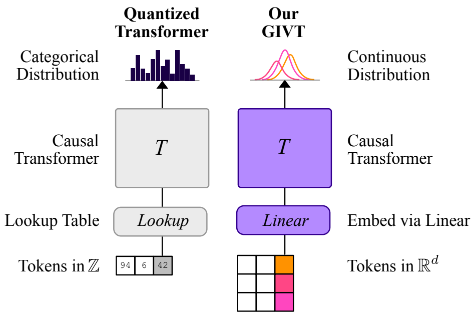

Fig. 2: We compare the standard discrete-token generative transformer (left) to our continuous, infinite-vocabulary variant (GIVT, right), using the same decoder-only architecture. At the input, GIVT linearly embeds a sequence of *real-valued vectors* instead of discrete tokens via lookup. At the output, GIVT predicts the parameters of a multivariate, continuous distribution rather than a categorical distribution.

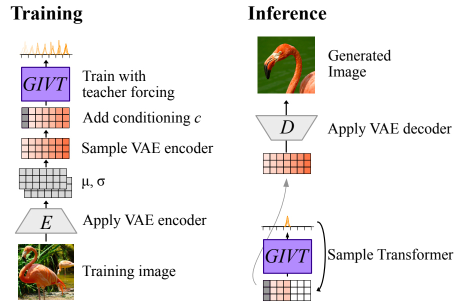

Fig. 3: GIVT-Causal training and inference. *Left:* During training, we sample a sequence of real-valued latent vectors from the VAE encoder, and train GIVT via teacher forcing. *Right:* During inference, we sample a sequence of vectors (left-to-right) and feed it to the VAE decoder. We note that we also explore MaskGIT-like GIVT models not shown here. *No component uses a quantizer.*

transformer decoder architecture [52, 72], see Fig. 2: 1) at the input, rather than using a sequence of discrete tokens to look up a finite vocabulary of embeddings, GIVT linearly embeds a sequence of real-valued vectors; and 2) at the output, rather than predicting a categorical distribution over a finite vocabulary, GIVT predicts the parameters of a d-variate Gaussian Mixture Model (GMM). We train GIVT in the same way as standard transformer decoders: with a causal attention mask and teacher forcing [72], and alternatively also explore fast progressive masked-bidirectional-modelling as in MaskGIT [6, 7, 13].

Similar to the two-stage approach with VQ-VAEs and analogous the two-stage approach of latent-diffusion models [51,55], we first learn a lower-dimensional latent space with a Gaussian-prior β-VAE [24,30], and then model it with GIVT. We emphasize that training both β-VAE and GIVT only relies on standard techniques from the deep-learning toolbox, and not the advanced training techniques of the VQ-VAE literature like auxiliary losses [7, 49] on the latent representation, codebook reinitialization [37], or dedicated optimization algorithms [27, 33].
Our main contributions can be summarized as follows:

1. We show that GIVT outperforms VQGAN [55] (and follow-up variants) and MaskGIT [7] in class-conditional image generation, often by a large margin and/or at significantly lower computational cost. GIVT is also competitive with strong latent diffusion baselines, particularly at high resolution.
2. We derive variants of standard sampling techniques for the continuous case, such as temperature sampling, beam search, and classifier-free guidance (CFG) [25], and showcase their effectiveness.

3. We demonstrate that GIVT matches or outperforms prior sequential image generation models in representation learning at significantly lower computational cost.
4. GIVT achieves comparable performance with the VQ-based UViM approach [33] in dense prediction tasks like semantic segmentation and monocular depth estimation.

We emphasize that advances in transformer decoder-based models for visual data generation as GIVT directly benefit form advances in scaling and inference efficiency for large language models. Conversely, and unlike for diffusion models, improvements in models as ours are straight-forward to transfer to multimodal interleaved modeling [1, 2, 29] which is becoming increasingly popular.

2 Related work

VQ-VAE for visual data tokenization Following the success of pixel-space autoregressive modeling [8, 43, 50, 59, 71] for image generation, moving the autorgressive modeling to the latent space of VQ-VAEs [49,54] emerged as a more efficient alternative. The use of GANs and perceptual losses for VQ-VAE training as well as modern causal [20, 73, 77] and masked [6, 7, 39] transformers for latent modeling led to substantial quality improvements. Another active area leveraging VQ-VAEs is interleaved multimodal generative modeling of images and text [1,2, 29]. Further, VQ-VAEs are a popular choice to tokenize the label space of dense prediction vision tasks [33, 42]. Finally, some language-inspired techniques for self-supervised learning from images rely on VQ-VAE representations [3, 39, 75].

Discretized mixtures of distributions replace the dense prediction of the logits of a categorical distribution with a continuous mixture model which is subsequently discretized. This approach was proposed in [59] for pixel-space autoregressive modeling, to reduce the number of model parameters and to improve learning efficiency, and is also popular in neural compression [10, 44, 45].

Continuous outputs in NLP A popular approach to handle large vocabularies in machine translation is to predict language tokens via their word embeddings with a continuous distribution, instead of token IDs with a categorical distribution [34, 35, 38, 64, 65]. Decoding is usually done in greedy fashion with embedding lookup and hence does not produce diverse samples. Further, the models consume and predict word embeddings form a fixed, finite set.

VAEs with learned priors A rich body of literature studies improving VAEs with learned priors: Inverse autoregressive flows emerged as a popular choice [9, 31]. Other approaches use normalizing flows [70] or a mixture of variational posteriors with pseudo-inputs [66]. For VAEs with discrete (non-VQ) latents, learned priors based on Restricted Boltzmann Machines were studied [57, 69].

Time-series modeling with Transformers A variety of works has recently explored transformers for time-series modeling/forecasting. Those works either use a regression loss [11, 22, 36, 48, 79], quantile forecasting [19, 41], or resort to discretizing/binning the data [53]. Somewhat related, [47, 74] regress continuous speech features from discrete tokens. None of these models predict a continuous distribution like GIVT that allows for autoregressive generation.

3 Generative infinite-vocabulary transformers

As mentioned in Sec. 1, our method is conceptually similar to recent works that train decoder-only transformer models on the discrete codes of VQ-VAEs [6, 7, 20, 76], with the crucial difference being that we do not quantize (*i.e.*, do not use VQ). We now describe the components of our method.

3.1 VAE training

We first train a *continuous-latent* β-VAE [24] with Gaussian encoder and prior as originally proposed by [30]. Given an input image x, the encoder E predicts mean μ, and covariance σ of a multivariate normal distribution with diagonal covariance matrix, and samples a representation z from $\mathcal{N}(\mu, \sigma)$ using the reparametrization trick [30]. The VAE decoder then maps the latent sequence back to an image. Since we use a Gaussian encoder distribution, the KL-term in the evidence lower bound (ELBO) [30] can be computed in closed form as described in [30, Sec. F.1]. As for the reconstruction/likelihood term in the ELBO, we rely on a mixture of MSE, perceptual loss and GAN loss for image generation following [7, 20], or the categorical cross-entropy for dense prediction tasks [33]. Our encoder spatially-downsamples x, whereby we obtain z with spatial dimensions $h \times w$ and feature dimension d, with $h{=}\lceil H/16 \rceil, w{=}\lceil W/16 \rceil$, given a $H{\times}W$ input x. To compute the KL-term, the associated μ and σ with shapes $w \times h \times d$ are flattened into whd vectors.

The hyperparameter β multiplying the KL-term controls how strongly z is regularized. As we shall see in Sec. 5, this regularization of the VAE is important to be able to model the resulting (true) latent distribution $p(z)$ well.

3.2 GIVT training

We next train a GIVT to predict $p(z)$ or $p(z|c)$ (when a conditioning signal c is available, *e.g.*, in class-conditional generation). The representation z is reshaped into a hw-length sequence of d-dimensional *real-valued* vectors (or "soft tokens"). Note how this differs from the standard VQ-VAE-based setup, where the latent transformer decoder models a hw-length sequence of *integers* denoting codebook indices. To accommodate this difference, we make two small changes to the standard transformer decoder-only architecture (see Fig. 2): We replace the embedding lookup tables at the input with a single linear layer to project from d to the transformer hidden dimension. At the output, we do not predict a categorical distribution, and instead let the transformer predict the parameters of a continuous distribution. Assuming channel-wise independence of the mixture components, we model this continuous distribution with a k-mixture GMM. The

Table 1: Results on class-conditional 256×256 ImageNet, where GIVT-Causal models outperform their quantization-based counterparts at much smaller model size (VQ-GAN) or substantially shorter sequence length (ViT-VQGAN). We report FID as well as precision and recall (where available). We use the standard ADM evaluation suite, where FID is calculated w.r.t. the training set. *+A*: GIVT variants with adapter, *CG*: Classifier guidance acceptance rate or scale, *CFG* $= w$: Classifier-free guidance with weight w [25], *DB-CFG* $= w$: Our distribution based CFG variant (Sec. 3.4), *Top-k*: Top-k sampling [21] ("mixed" refers to multiple k), t: Temperature sampling by scaling the predicted σ of our models with t, t_C: Choice temperature for MaskGIT. *Steps* number of inference steps. Additional comments: †Numbers obtained by us from public code, ⋆Inference uses activation caching.

	Model	Inference	Steps	FID↓	Precision↑	Recall↑
GANs	BigGAN-deep [5]			6.95	**0.87**	0.28
	StyleGAN-XL [60]			**2.30**	0.78	**0.53**
Diffusion Models	ADM [14]		250	10.94	0.69	0.63
	ADM-G [14]	CG = 1.0	250	4.59	0.82	0.52
	LDM-4 [55]		250	10.56	0.71	0.62
	LDM-4-G [55]	CFG = 1.5	250	3.60	**0.87**	0.48
	DiT-XL/2 [51]		250	9.62	0.67	**0.67**
	DiT-XL/2-G [51]	CFG = 1.5	250	**2.27**	0.83	0.57
Masked Modeling	MaskGIT [7]	$t_C = 4.5$	16	4.92†	0.84†	**0.49†**
	GIVT-MaskGIT *(Ours)*	$t_C = 35$	16	4.64	0.85	**0.49**
	GIVT-MaskGIT *(Ours)*	$t_C = 60$, DB-CFG = 0.1	16	**4.53**	**0.87**	0.47
Sequence Models	VQGAN [20]	Top-k = Mixed	256⋆	17.04		
	VQGAN [20]	Top-k = 600, CG = 0.05	256⋆	5.20		
	ViT-VQGAN-L [76]		1024⋆	4.17		
	ViT-VQGAN-L [76]	CG = 0.5	1024⋆	3.04		
	GIVT-Causal *(Ours)*	$t = 0.9$	256⋆	5.67	0.75	0.59
	GIVT-Causal *(Ours)*	$t = 0.95$, DB-CFG = 0.4	256⋆	3.35	**0.84**	0.53
	GIVT-Causal-L+A *(Ours)*	$t = 0.9$	256⋆	3.46	0.77	**0.61**
	GIVT-Causal-L+A *(Ours)*	$t = 0.95$, DB-CFG = 0.4	256⋆	**2.59**	0.81	0.57

GIVT model hence predicts $2kd + k$ parameters per soft token (kd mean and kd variance parameters for the mixture components, and k mixture probabilities). Experimentally, we found it beneficial to normalize the mixture probabilities with a softmax activation, and the variance parameters with softplus.

We use the standard cross-entropy loss (which is equivalent to the negative log-likelihood) on the distribution $\tilde{p}$ predicted by GIVT, and minimize $\mathcal{L}_\mathrm{T} = \sum_c \mathbb{E}_z \left[-\log \tilde{p}(z|c)\right]$, assuming the the classes or conditioning signal c uniformly distributed (see App. C.1 for details on the loss). We train two types of GIVT models, as described next.

GIVT-Causal Here, GIVT is trained to predict every d-dimensional vector in the hw sequence of latents conditioned on all previous vectors. Thereby, the self-attention layers are masked to be temporally causal [20, 72] (which enables sequential generation at inference time and is unrelated to causal inference). This training strategy is also called teacher forcing and is analogous to the latent modeling in VQ-GAN [20]. For class-conditional image generation we prepend a `[CLS]` vector to the input sequence, *i.e.*, a learned vector for each class c.

GIVT-MaskGIT As in MaskGIT [7], we mask a subset of the input sequence randomly during training and then gradually uncover the masked tokens during inference. The only changes compared to [7] are related to our real-valued tokens: since we have infinitely many tokens, there is no obvious choice to define a special mask token (when using VQ, one can just extend the vocabulary to contain special tokens, such as `[MASK]`). Instead, given z and a mask M indicating for every location whether it is masked, we first replace the locations in z corresponding to M with zeros (to remove information), and then embed it with a single dense layer, as above. Additionally, we *concatenate* one of two learned special vectors in the feature dimension, a `[MASK]` vector for masked locations, and a `[UNMASK]` vector otherwise (we half the dimension of the embedded inputs and special tokens s.t. the final hidden dimension remains unchanged).

3.3 Towards end-to-end training: Adapters

An interesting consequence of using an unquantized VAE and modeling the resulting latent sequence with a continuous rather than a categorical distribution is that the VAE and GIVT can be jointly trained or fine-tuned end-to-end (using the reparametrization trick [30]). However, this setup comes with its own set of challenges (*e.g.*, it encompasses multiple losses which have to be balanced appropriately) and we leave it for future work. Instead, we explore a simple alternative to better match the latent distributions of the VAE and the one predicted by GIVT: We use a small invertible flow model [15, 16], or "adapter", to map the VAE latent sequences to a new latent space of identical dimensions. We rely on a "volume preserving" additive coupling layer-based model which has a diagonal Jacobian [15]. GIVT is then trained jointly with the adapter to predict the sequences in this transformed latent space induced by the adapter (using the same loss). At inference time, samples drawn from GIVT are first processed by the inverted adapter, and then decoded to an image with the VAE decoder. Note that the adapter does not require additional losses thanks to invertibility and adds a negligible compute and model parameter overhead (less than 0.1%) compared to the GIVT model (see Sec. 4 and App. B for details).

3.4 Inference

Given a VAE and GIVT trained as above, during inference we sample form GIVT either sequentially (see Fig. 3) or as in MaskGIT [7] and decode the sampled sequence into an image. We now investigate the various inference schemes for discrete transformers, and derive their continuous counterparts.

Temperature Sampling, Nucleus Sampling, Beam Search In sequence models for text (see [26] for an overview and discussion) and VQ-GAN-based approaches, it is common to adapt and tune the sampling algorithm. We start with temperature sampling, which for discrete models adapts the softmax temperature of the categorical distributions predicted at each decoding step. For

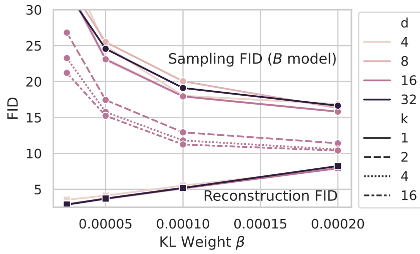

Fig. 4: β-VAE ablation: Interplay of KL weight β, number of channels d, and number of mixtures k when training the VAE. Round markers show the sampling FIDs obtained with a Base-size GIVT-Causal. As β and k increase, the sampling FID improves, but the reconstruction FID also increases, limiting the best possible sampling FID.

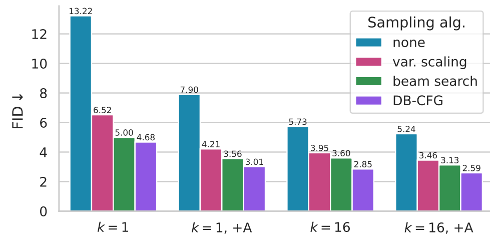

Fig. 5: Effect of different sampling strategies and model variants (GIVT-Causal-L) on sample quality. Increasing the number of mixtures k and adding an adapter (+A) lead to compounding improvements. DB-CFG is the most effective sampling strategy for all model configurations.

GIVT, we instead scale the covariance matrices of the predicted Gaussian distributions and call this strategy "variance scaling". As we will see in Sec. 4, this simple change can have a significant impact on sample quality.

Nucleus sampling [26] proposes to collect the largest logits such that its cumulative probability after normalization exceeds a threshold (for example 0.8), and to sample from this reduced-support distribution. In GIVT, when predicting a single mixture, this can be approximated by truncating the predicted distributions per dimension (thereby choosing a higher-density support). This has a similar effect to variance scaling and therefore do not pursue this strategy.

We also consider beam search, which is the same for GIVT as it is for discrete transformer decoders. For every sample, we maintain B beams, and at every step we sample a number of candidates for every beam (we call these "fans" here). We then compute the cumulative log probability for all beams and fans up to the current sampling step, and select the B beams with the highest cumulative log probability. Finally, there is no analogous concept for top-k sampling [21] in GIVT, because it predicts continuous distributions.

Distribution-Based Classifier-Free Guidance In the diffusion literature, classifier-free guidance (CFG) [25] has been employed with great success. Concretely, conditional diffusion models are trained with an additional null class $\emptyset$ to learn the unconditional data distribution. Then, during inference, the conditional log density is "moved away" from the unconditional one: given a guidance weight w, the updated (diffusion) score estimate is is obtained as

$$\tilde{\epsilon}(z, c) = (1 + w)\epsilon(z, c) - w\epsilon(z, \emptyset), \tag{1}$$

where ϵ estimates the gradient of the log density of the data distribution, $\epsilon(z, c) \propto \nabla_z \log \tilde{p}(z|c)$ (see [25, Sec. 2]). From this, we now derive a CFG variant for our

GIVT, since *we directly predict a density*. We term this approach "Density-Based CFG" (DB-CFG). Eq. 1 can be written as

$$\begin{aligned}\tilde{\epsilon}(z,c) &\propto (1+w)\nabla_z \log \tilde{p}(z|c) - w\nabla_z \log \tilde{p}(z|\emptyset) \\ &\propto \nabla_z \log\left(\tilde{p}(z|c)^{1+w}\tilde{p}(z|\emptyset)^{-w}\right),\end{aligned}$$

i.e., $\tilde{\epsilon}$ estimates the log of the density $p_{\text{CFG}}(z|c) \propto \tilde{p}(z|c)^{1+w}\tilde{p}(z|\emptyset)^{-w}$ (see Fig. 6). Thus, we want to adapt our models to sample from p_{CFG}. We follow [25] and train GIVT with an additional null class $\emptyset$. During inference, we evaluate GIVT twice at every step, once conditional on the actual label c and once conditional on $\emptyset$. To implement classifier-free guidance, we then have to sample from an unnormalized version of $p_{\text{CFG}}(z)$ derived from the two GIVT predictions. To this end, we turn to rejection sampling, which requires: 1) an unnormalized density; 2) a good proposal distribution p', that is close to the true target distribution; and 3) a scaling factor K to bound the likelihood ratio between p' and the unnormalized target density.

The distributions we mix are GMMs and finding a good proposal distribution can be challenging. Instead, we first sample the mixture index from $\tilde{p}(z|c)$ and apply DB-CFG to the corresponding mixture components from $\tilde{p}(z|c)$ and $\tilde{p}(z)$ (the components are multivariate Gaussians with diagonal covariance). We find empirically that the unconditional components (*i.e.*, distributions predicted using the $\emptyset$ label) tend to have larger variance than the conditional ones (as visualized in Fig. 6). It is thus sensible to pick sample proposals from $\mathcal{N}(\mu_c, 2\sigma_c)$, where μ_c, σ_c are the parameters predicted by GIVT when given the label c. We empirically find that drawing 1000 samples is enough to find at least one valid sample 99.9% of the time. For the remaining <0.1%, fall back to sampling from $\mathcal{N}(\mu_c, \sigma_c)$.

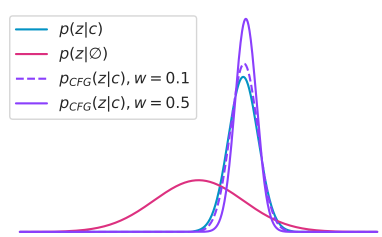

Fig. 6: Visualization of our *Density-Based Classifier-Free Guidance (DB-CFG)*. We show the conditional and unconditional PDFs predicted by GIVT, and the resulting CFG PDF for two values of w. Note how the CFG distributions become more peaked. We use rejection sampling to sample from p_{CFG}.

We emphasize that the overhead of DB-CFG is small: it requires two forward passes (per inference step) instead of one to predict the conditional and unconditional distribution. We then draw 1000 samples from those in parallel on an accelerator, which is very fast. We refer to App. 3.4 for Python code.

4 Experiments

4.1 Image generation

We use ImageNet1k [56] and explore class-conditional generation (where we condition our GIVT on class labels) for 256px and 512px, and *un*conditional generation for 256px.

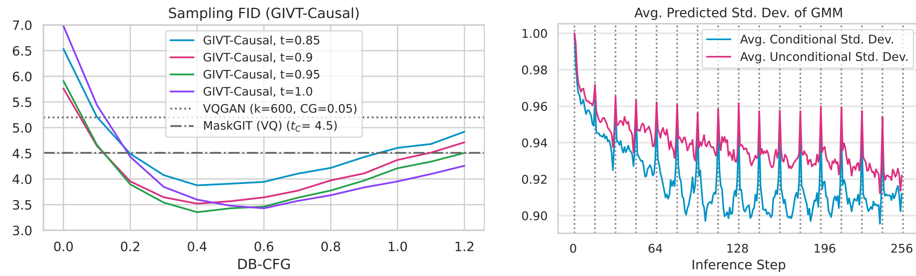

Fig. 7: *Left:* Impact of DB-CFG (Sec. 3.4) and variance scaling (Sec. 3.4) on sampling FID of our class-conditional 256×256 GIVT-Causal models. DB-CFG values in $[0.3, 0.8]$ and variance scaling parameter t in $[0.9, 1.0]$ lead to low FID. *Right:* Average standard deviation of the GMM predicted by GIVT-Causal for class 130, averaged over 128 samples: conditional predictions have lower standard deviation; spikes can be observed when the line changes in the raster scan over the latent feature vectors.

β-VAE We closely follow the setup of MaskGIT [7]. We use their VAE architecture, built of ResBlocks (as detailed in App. C; encoder and decoder have a combined 53.5M parameters), remove the VQ layer and related losses, and replace it with a linear layer predicting μ, σ (Sec. 3.1). We use the same weights for reconstruction, perceptual, and GAN-loss, as well as identical optimizer parameters, as in [7, 46]; we only vary the latent dimension d and weight β of the KL-term. By default, we set the token dimension to $d = 16$ (*i.e.*, the VAE predicts 16 means and variances per token) and $\beta = 5\cdot10^{-5}$. We note that our VAE is trained on 256×256 images, and we also use it for our 512×512 experiments without retraining (like [7]).

GIVT For GIVT-Causal, we follow the original transformer decoder architecture [72] in decoder-only mode, but remove biases from attention layers, MLP blocks, and LayerNorms, and replace ReLU by GELU as is common practice. For GIVT-MaskGIT, we simply remove the attention mask during training and feed masked inputs instead of shifted ones. We use the BERT-Large configuration [13] by default (304M parameters), and also explore a larger backbone with 1.67B parameters, denoted with the suffix "-L" (see App. B for details). For model variants with adapter (suffix "+A"), we use a stack of 8 bijective iRevNet blocks [28] (with hidden channel dimension $4d$, resulting in 112k additional parameters for $d = 16$), applied to the $w \times h \times d$ representation before reshaping it into a sequence. We configure our GIVT models to predict a 16-mixture GMM with factorized components (i.e. the mixture components are multivariate Gaussians with diagonal covariance), and explore predicting a single, multivariate Gaussian modeling the full covariance matrix of the tokens as an alternative. For the conditional generation experiments, we use a learned embedding which we prepend to the embedded token sequence. To train GIVT, we use Adam with a cosine schedule (500 epochs; with linear warmup of 50 epochs), set the learning rate and weight decay to 10^{-3} and 10^{-4}, respectively, the optimizer β_2 parameter to 0.95, the dropout probability to 0.2 for GIVT-causal and 0.4 for

GIVT-MaskGIT, and the batch size to 8192. We use the same data augmentation as during VAE training (see [7, 46]), and sample from the VAE encoder distribution for every batch (an additional source of randomness besides data augmentation).

We implement GIVT in JAX [4] and use distrax [12] to implement the cand compute the log-probabilities.

GIVT-MaskGIT inference Following [7], we fix the number of inference steps to 16 and employ the cosine schedule (*i.e.* letting $r = i/16$ at step i, the fraction of masked tokens is given by $\cos(\pi/2r)$). We also sort tokens by likelihood at each step and sample using a "choice temperature" t_C.

Exploring the VAE latent space To better understand the interplay between the feature dimension d, the KL regularization β, the reconstruction quality of the VAE, and the sampling quality of the GIVT, we train VAEs with latent dimension in $\{4, 8, 16, 32\}$ and β in $\{2.5 \cdot 10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 2 \cdot 10^{-4}\}$ using the VAE-training setup described at the beginning of this section. For each of the resulting VAEs we train a GIVT-Causal with the smaller BERT-Base [13] dimensions and a range of values for the number of mixtures k.

Evaluation For the VAEs we report "reconstruction FID", the FID obtained when reconstructing the 50k ImageNet validation images. For our GIVT variants and baselines, we report the sampling FID [23] when sampling a balanced set of 50k images covering all ImageNet classes. In both cases, we rely on the well-established ADM TensorFlow Suite [14], which uses the entire ImageNet training set as a reference. Furthermore, we also report Precision and Recall [58]. Finally, we evaluate the representation learning capabilities by training a linear classifier on an average-pooled intermediate representation of unconditional GIVT-Causal as in prior work [8, 76] (see App. F for details).

4.2 Panoptic segmentation and depth estimation

We build on the UViM framework [33], which uses a VQ-VAE to compress the label space of computer-vision dense-prediction tasks, and an encoder-decoder transformer taking the RGB image as an input and predicting the associated dense labels as discrete codes in the VQ-VAE latent space. Here, we replace the VQ-VAE with a β-VAE and use a GIVT encoder-decoder to model the continuous latent code. For the VAE, we use the same transformer-based autoencoder architecture (6-layer encoder and 12-layer decoder) and cross-entropy loss as [33]. We set $d = 16$, $k = 1$, and KL weight $\beta = 2.5 \cdot 10^{-4}$ for panoptic segmentation and $\beta = 2 \cdot 10^{-4}$ for depth estimation. To build an encoder-decoder GIVT model the same way as in [33], we employ the causal variant described for ImageNet generation and insert a cross-attention layer after each self-attention layer. Following [33], we use the ImageNet-21k-pretrained ViT-L/16 from [62] as the encoder, set the image resolution to 512px, and adopt the preprocessing and optimization hyper-parameters from [33]. We use the UViM variant without encoder and decoder context [33]. Finally, we consider variance scaling and beam search, selecting the parameters on a held-out subset of the training set as [33].

Table 2: Results on class-conditional 512×512 ImageNet. We use the standard ADM evaluation suite for metrics, where FID is calculated w.r.t. the training set. GIVT-MaskGIT obtains competitive FID scores with only 16 inference steps and outperforms its VQ-counterpart. GIVT-Causal-L+A outperforms the best DiT variant, DiT-XL/2-G. †Values obtained by us from public code. ⋆Inference uses activation caching.

Model	Inference	Steps	FID↓	Precision↑	Recall↑
ADM [14]		250	23.20	0.73	0.60
ADM-G [14]	CG = 1.0	250	7.72	**0.87**	0.42
DiT-XL/2 [51]		250	12.03	0.75	**0.64**
DiT-XL/2-G [51]	CFG = 1.5	250	**3.04**	0.84	0.54
MaskGIT [7]	$t_C = 4.5$	16	7.80†	0.86†	**0.46†**
GIVT-MaskGIT *(Ours)*	$t_C = 140$	16	**4.86**	**0.88**	**0.48**
GIVT-Causal-L *(Ours)*	$t = 0.9$	512⋆	8.35	0.79	**0.61**
GIVT-Causal-L+A *(Ours)*	$t = 0.9$, DB-CFG = 0.9	512⋆	**2.92**	**0.84**	0.55

5 Results

5.1 Image generation

VAE latent space In Fig. 4 we show how varying the weight β of the KL term affects 1) the VAE reconstruction FID and 2) the sampling FID of a Base-size GIVT-Causal trained on the corresponding latent sequence. For 1), increasing β leads to worse reconstruction FID since the VAE can store less information in the latent. It shifts more of the modeling effort to the VAE decoder, so that the decoder becomes gradually more generative, which affects sampling quality (see [67, Sec. 7] [55] for more discussion).

For 2), we see the opposite trend: increasing β leads to decreased (better) sampling FID for GIVT models trained on the latent sequence. Arguably, this is because the VAE latent sequence more closely follows the Gaussian prior, and hence becomes easier for the GIVT to model. Finally, increasing the number of mixtures k initially reduces the sampling FID substantially, reaching a plateau at $k = 16$. We hence set $k = 16$ and $\beta = 5 \cdot 10^{-5}$ by default, and use a VAE with $\beta = 10^{-5}$ for the larger (-L) GIVT models. We emphasize that it is common in the literature to explore and tune hyper-parameters such as β or analogously the vocab size and commitment loss in VQ [20, Table 5] [55, Table 8] [51] [46, Fig. 3].

Sampling FID In Table 1 we show the sampling FID for four model classes on class-conditional 256×256 ImageNet: GANs, diffusion-based approaches, as well as masked and sequence modeling approaches. GIVT-MaskGIT outperforms MaskGIT [7] which has comparable model size and inference cost, and DB-CFG leads to an additional improvement. In absence of guidance techniques, our GIVT-Causal models outperform all diffusion baselines as well as VQGAN by a large margin. Using guidance techniques, GIVT-Causal obtains FID of 3.35 compared to 5.20 for VQGAN with a more than 4.5× smaller model (0.3B for GIVT vs. 1.4B parameters), and also outperforms the 32% larger LDM-4-G. Our larger GIVT-Causal-L+A obtains 16% and 17% reduction in FID without and with guidance, respectively, compared to ViT-VQGAN which has the same

Table 3: UViM based on GIVT-Causal and VQ-VAE evaluated on panoptic segmentation (COCO Panoptic 2017) and depth estimation (NYU Depth v2). We report the panoptic quality (PQ) and RMSE for the VAE/VQ-VAE reconstructions of the validation set label maps (recon.) and the inference metrics on the actual dense prediction tasks (inference). GIVT obtains metrics comparable to the VQ-based UViM.

	COCO Pan. (PQ↑)		NYU Depth v2 (RMSE↓)	
	recon.	inference	recon.	inference
UViM [33]	66.0	39.0	0.183	**0.459**
GIVT (ours)	71.0	**40.2**	0.195	0.474

generative transformer size but a 4× larger sequence length (resulting in more than 4× slower sampling) and a 10× larger VAE.

We present sampling FID for 512×512 ImageNet in Table 2. GIVT-MaskGIT obtains a 38% lower FID than MaskGIT with comparable model size and inference cost. GIVT-Causal-L+A outperforms DiT-XL/2, the best available DiT model, both without and with guidance (albeit with a larger model).

Finally, we present *un*conditional results in App. E. This task is considerably harder, but GIVT-Causal beats the diffusion-based ADM [14] by a large margin.

Ablations and visualizations Fig. 5 compares the effect of model configuration (number of mixtures k, adapter) and sampling algorithm (variance scaling, beam search, DB-CFG) on FID. For every model configuration, all the sampling algorithms lead to solid improvements in FID, with DB-CFG being the most effective one across all configurations. Increasing k from 1 to 16 overall leads to somewhat larger improvements than keeping $k = 1$ and adding an adapter. Moreover, combining adapter with $k = 16$ results in compounding improvements across sampling algorithms.

Fig. 7 (left) shows the impact of the variance scaling and CFG parameters on the sampling FID. In Fig. 7 (right), we visualize the predicted standard deviation as a function of the GIVT-Causal inference step. The standard deviation gradually decreases, meaning that the predictions later in the sampling process become more certain. Furthermore, the unconditional predictions generally have a higher standard deviation, as expected.

For GIVT-MaskGIT, predicting a single Gaussian with full covariance matrix per latent vector, rather than assuming a diagonal covariance, only led to very modest gains of about 3%. A GMM with factorized component densities therefore seems to be the more effective alternative. Furthermore, a full covariance matrix makes DB-CFG less tractable than the diagonal covariance (because the high dimensional multivariate distribution has more regions of low density).

Samples Fig. 1 shows ten 512×512 samples from GIVT-Causal-L+A, and App. I shows samples for other GIVT-Causal variants and GIVT-MaskGIT. We can see that the model produces high-fidelity, coherent samples. To study sample diversity, we show multiple samples from different models for a fixed class.

In Fig. 15 in App. I, one can see two samples from our VAE (obtained by decoding latents sampled from the prior), which show a soup of image textures. We then show different steps of the GIVT-MaskGIT inference, and observe similar behavior as in the VQ-based model ([7, Fig. 2]).

Representation learning Table 4 shows the ImageNet linear probing accuracy of unconditional GIVT-Causal and generative models from the literature (we chose the model variants closest in terms of model size and compute). GIVT-Causal matches VIM+ViT (ViT-VQGAN) [76] which has more than 2× the model parameters and 4× the sequence length (and hence FLOPs). GIVT-Causal is only outperformed by MAGE [39], whose latent encoder-decoder architecture is better suited for representation learning than decoder-only models. An investigation of the probing accuracy as function of the layer index can be found in App. F.

Table 4: ImageNet linear probing accuracy of unconditional GIVT-Causal and generative models from the literature. GIVT-Causal matches VIM+ViT (ViT-VQ-GAN) [76] which has more than 2× the model parameters and 4× the sequence length (and hence FLOPs). *Type*: (Latent) generative model type. *#Param.*: Number of parameters of the full (latent) generative model.

Model	Type	#Tok.	#Param.	Acc.↑
BigBiGAN [17]			344M	61.3
iGPT-L [8]	dec.-only	1024	1362M	60.3
VIM+CNN [76]	dec.-only	1024	650M	61.8
VIM+ViT [76]	dec.-only	1024	650M	**65.1**
MAGE ViT-L [39]	enc.-dec.	256	404M	78.9
GIVT-Causal *(Ours)*	dec.-only	256	304M	**65.1**

5.2 Panoptic segmentation and depth estimation

Table 3 compares the performance of a GIVT-based UViM variant with a VQ-VAE-based baseline (both without encoder/decoder context) on COCO Panoptic 2017 [32] and NYU Depth v2 [61]. We report the panoptic quality metric (PQ) [32] and RMSE, respectively, and find that our GIVT-based model outperforms the baseline in panoptic segmentation and performs slightly worse in depth estimation. In App. I we show visual results.

6 Conclusion

In this paper, we proposed simple modifications to standard transformer decoder-only models enabling them to generating real-valued vectors. To our knowledge, this is the first decoder-only model amenable to generating sequences of real-valued vectors. In the context of image generation with VQ-GAN or MaskGIT, this side-steps training difficulties such as low codebook usage in VQ-VAEs and corresponding mitigations like entropy losses or codebook-splitting algorithms, by enabling the use of standard VAEs which are much easier to train. Furthermore, our method avoids large embedding matrices because the feature representations can directly be consumed and predicted by our GIVT model. Our simple, quantization-free approach outperforms its VQ-based counterparts in class-conditional image generation and image representation learning, often by significant margins. GIVT also obtains strong performance in dense prediction tasks when applied to UViM. We hope that future work explores applications of GIVT to other modalities such as audio and time-series modeling.

Acknowledgments

We would like to thank André Susano Pinto, Neil Houlsby, Eirikur Agustsson, Lucas Theis, and Basil Mustafa for inspiring discussions and helpful feedback on this project. We also thank Han Zhang for support with the VAE training code.

References

1. Aghajanyan, A., Huang, P.Y., Ross, C., Karpukhin, V., Xu, H., Goyal, N., Okhonko, D., Joshi, M., Ghosh, G., Lewis, M., Zettlemoyer, L.: CM3: A causal masked multimodal model of the internet. arXiv:2201.07520 (2022)
2. Aghajanyan, A., Yu, L., Conneau, A., Hsu, W.N., Hambardzumyan, K., Zhang, S., Roller, S., Goyal, N., Levy, O., Zettlemoyer, L.: Scaling laws for generative mixed-modal language models. In: ICML (2023)
3. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT pre-training of image transformers. In: ICLR (2021)
4. Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: composable transformations of Python+NumPy programs (2018), `http://github.com/google/jax`
5. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: ICLR (2018)
6. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M., Murphy, K.P., Freeman, W.T., Rubinstein, M., Li, Y., Krishnan, D.: Muse: Text-to-image generation via masked generative transformers. In: ICML (2023)
7. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: Masked generative image transformer. In: CVPR. pp. 11315–11325 (2022)
8. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: ICML. pp. 1691–1703 (2020)
9. Chen, X., Kingma, D.P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., Abbeel, P.: Variational lossy autoencoder. In: ICLR (2016)
10. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: CVPR. pp. 7939–7948 (2020)
11. Das, A., Kong, W., Sen, R., Zhou, Y.: A decoder-only foundation model for time-series forecasting. arXiv:2310.10688 (2023)
12. DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kapturowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J., Ring, R., Ruiz, F., Sanchez, A., Sartran, L., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stanojević, M., Stokowiec, W., Wang, L., Zhou, G., Viola, F.: The DeepMind JAX Ecosystem (2020), `http://github.com/deepmind`
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT (2019)
14. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. NeurIPS pp. 8780–8794 (2021)

15. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. In: ICLR (2015)
16. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. In: ICLR (2017)
17. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: NeurIPS (2019)
18. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
19. Eisenach, C., Patel, Y., Madeka, D.: MQTransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. arXiv:2009.14799 (2020)
20. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12868–12878 (2020)
21. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: ACL. pp. 889–898 (2018)
22. Garza, A., Mergenthaler-Canseco, M.: TimeGPT-1. arXiv:2310.03589 (2023)
23. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
24. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: Beta-VAE: Learning basic visual concepts with a constrained variational framework. In: ICLR (2016)
25. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv:2207.12598 (2022)
26. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: ICLR (2019)
27. Huh, M., Cheung, B., Agrawal, P., Isola, P.: Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In: ICML (2023)
28. Jacobsen, J.H., Smeulders, A.W., Oyallon, E.: i-revnet: Deep invertible networks. In: ICLR (2018)
29. Kim, S., Jo, D., Lee, D., Kim, J.: MAGVLT: Masked generative vision-and-language transformer. In: CVPR. pp. 23338–23348 (2023)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv:1312.6114 (2013)
31. Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. NeurIPS (2016)
32. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR. pp. 9404–9413 (2019)
33. Kolesnikov, A., Susano Pinto, A., Beyer, L., Zhai, X., Harmsen, J., Houlsby, N.: UViM: A unified modeling approach for vision with learned guiding codes. NeurIPS pp. 26295–26308 (2022)
34. Kumar, S., Anastasopoulos, A., Wintner, S., Tsvetkov, Y.: Machine translation into low-resource language varieties. In: ACL. pp. 110–121 (2021)
35. Kumar, S., Tsvetkov, Y.: Von Mises-Fisher loss for training sequence to sequence models with continuous outputs. In: ICLR (2018)
36. Kunz, M., Birr, S., Raslan, M., Ma, L., Li, Z., Gouttes, A., Koren, M., Naghibi, T., Stephan, J., Bulycheva, M., Grzeschik, M., Keki'c, A., Narodovitch, M., Rasul, K., Sieber, J., Januschowski, T.: Deep learning based forecasting: a case study from the online fashion industry. arXiv:2305.14406 (2023)

37. Łańcucki, A., Chorowski, J., Sanchez, G., Marxer, R., Chen, N., Dolfing, H.J., Khurana, S., Alumäe, T., Laurent, A.: Robust training of vector quantized bottleneck models. In: IJCNN. pp. 1–7 (2020)
38. Li, L.H., Chen, P.H., Hsieh, C.J., Chang, K.W.: Efficient contextual representation learning without softmax layer. arXiv:1902.11269 (2019)
39. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. In: CVPR. pp. 2142–2152 (2023)
40. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: ECCV. pp. 280–296 (2022)
41. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting pp. 1748–1764 (2021)
42. Lu, J., Clark, C., Zellers, R., Mottaghi, R., Kembhavi, A.: Unified-IO: A unified model for vision, language, and multi-modal tasks. In: ICLR (2022)
43. Menick, J., Kalchbrenner, N.: Generating high fidelity images with subscale pixel networks and multidimensional upscaling. arXiv:1812.01608 (2018)
44. Mentzer, F., Agustsson, E., Tschannen, M.: M2T: Masking transformers twice for faster decoding. In: ICCV (2023)
45. Mentzer, F., Gool, L.V., Tschannen, M.: Learning better lossless compression using lossy compression. In: CVPR. pp. 6638–6647 (2020)
46. Mentzer, F., Minnen, D., Agustsson, E., Tschannen, M.: Finite scalar quantization: VQ-VAE made simple. arXiv:2309.15505 (2023)
47. Nachmani, E., Levkovitch, A., Salazar, J., Asawaroengchai, C., Mariooryad, S., Skerry-Ryan, R., Ramanovich, M.T.: Lms with a voice: Spoken language modeling beyond speech tokens. arXiv:2305.15255 (2023)
48. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: ICLR (2022)
49. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. NeurIPS (2017)
50. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: ICML. pp. 4055–4064 (2018)
51. Peebles, W., Xie, S.: Scalable diffusion models with transformers. arXiv:2212.09748 (2022)
52. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
53. Rasul, K., Ashok, A., Williams, A.R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Bilovs, M., Ghonia, H., Hassen, N., Schneider, A., Garg, S., Drouin, A., Chapados, N., Nevmyvaka, Y., Rish, I.: Lag-Llama: Towards foundation models for time series forecasting. arXiv:2310.08278 (2023)
54. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. NeurIPS (2019)
55. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
56. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**, 211–252 (2015)
57. Sadeghi, H., Andriyash, E., Vinci, W., Buffoni, L., Amin, M.H.: PixelVAE++: Improved pixelvae with discrete prior. arXiv:1908.09948 (2019)
58. Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. NeurIPS (2018)

59. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In: ICLR (2016)
60. Sauer, A., Schwarz, K., Geiger, A.: StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In: SIGGRAPH (2022)
61. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV. pp. 746–760 (2012)
62. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your ViT? data, augmentation, and regularization in vision transformers. TMLR (2021)
63. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: CVPR. pp. 7262–7272 (2021)
64. Tokarchuk, E., Niculae, V.: On target representation in continuous-output neural machine translation. In: ACL (2022)
65. Tokarchuk, E., Niculae, V.: The unreasonable effectiveness of random target embeddings for continuous-output neural machine translation. arXiv:2310.20620 (2023)
66. Tomczak, J., Welling, M.: Vae with a vampprior. In: AISTATS. pp. 1214–1223 (2018)
67. Tschannen, M., Bachem, O., Lucic, M.: Recent advances in autoencoder-based representation learning. arXiv:1812.05069 (2018)
68. Tschannen, M., Kumar, M., Steiner, A., Zhai, X., Houlsby, N., Beyer, L.: Image captioners are scalable vision learners too. In: NeurIPS (2023)
69. Vahdat, A., Andriyash, E., Macready, W.: Dvae#: Discrete variational autoencoders with relaxed boltzmann priors. NeurIPS (2018)
70. Vahdat, A., Kautz, J.: NVAE: A deep hierarchical variational autoencoder. NeurIPS pp. 19667–19679 (2020)
71. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: ICML. pp. 1747–1756 (2016)
72. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
73. Villegas, R., Babaeizadeh, M., Kindermans, P.J., Moraldo, H., Zhang, H., Saffar, M.T., Castro, S., Kunze, J., Erhan, D.: Phenaki: Variable length video generation from open domain textual descriptions. In: ICLR (2022)
74. Wang, J., Du, Z., Chen, Q., Chu, Y., Gao, Z., Li, Z., Hu, K., Zhou, X., Xu, J., Ma, Z., Wang, W., Zheng, S., Zhou, C., Yan, Z., Zhang, S.: LauraGPT: Listen, attend, understand, and regenerate audio with GPT. arXiv:2310.04673 (2023)
75. Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.G., Zhou, L., Yuan, L.: Bevt: Bert pretraining of video transformers. In: CVPR. pp. 14733–14743 (2022)
76. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved VQGAN. ICLR (2022)
77. Yu, J., Xu, Y., Koh, J.Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B.K., Hutchinson, B.C., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., Wu, Y.: Scaling autoregressive models for content-rich text-to-image generation. TMLR (2022)
78. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: CVPR. pp. 12104–12113 (2022)
79. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: AAAI. pp. 11106–11115 (2021)