Unleashing Transformers: Supplementary Material

The supplementary material for this work is divided into the following sections: Appendix A describes the architectures and hyperparameters for the experiments presented in the main paper; Appendix B illustrates the connection between our proposed ELBO reweighting and the true ELBO; Appendix C contains extra FID comparisons; Appendix D compares our approach with concurrent works; Appendix E gives nearest neighbour examples to demonstrate generalisation; and finally, Appendix F contains additional samples at resolutions higher than the training data.

A Implementation Details

We perform all experiments on a single NVIDIA RTX 2080 Ti with 11GB of VRAM using automatic mixed precision when possible. As mentioned in the main paper, we use the same VQGAN architecture as used by Esser et al. [3] which for 256×256 images downsamples to features of size $16 \times 16 \times 256$, and quantizes using a codebook with 1024 entries. Attention layers are applied within both the encoder and decoder on the lowest resolutions to aggregate context across the entire image. Models are optimised using the Adam optimiser [5] using a batch size of 4 and learning rate of 1.8×10^{-5}. For the differentiable augmentations we randomly change the brightness, saturation, and contrast, as well as randomly translate images. The datasets we use are both publically accessible, with FFHQ availble under the Creative Commons BY 4.0 licence. LSUN models are trained for 2.2M steps and the FFHQ model for 1.4M steps.

For the absorbing diffusion model we use a scaled down 80M parameter version of GPT-2 [9] consisting of 24 layers, where each attention layer has 8 heads, each 64D. The same architecture is used for experiments with the autoregressive model. Autoregressive models' training are stopped based on the best validation loss. We also stop training the absorbing diffusion models based on validation ELBO, however, on the LSUN datasets we found that it always improved or remained consistent throughout training so each model was trained for 2M steps.

Codebook Collapse One issue with vector quantized methods is codebook collapse, where some codes fall out of use which limits the potential expressivity of the model. We found this to occur across all datasets with often a fraction of the codes in use. We experimented with different quantization schemes such as gumbel softmax, different initalisation schemes such as k-means, and 'code recycling', where codes out of use are reset to an in use code. In all of these cases, we found the reconstruction quality to be comparable or worse so stuck with the argmax quantisation scheme used by Esser et al. [3].

Precision, Recall, Density, and Coverage To compute these measures we use the official code releases and pretrained weights in all cases except Taming Transformers on the LSUN datasets where weights were not available; in this case

we reproduced results as close as possible with the hardware available, training the VQGANs and autoregressive models with the same hyperparameters used for the rest of our experiments. Following Nash et al. [7] we use the standard 2048D InceptionV3 features, which are also used to compute FID, $k = 3$ nearest neighbours, and 50k samples, and use the code provided by Naeem et al. [6].

B Reweighted ELBO

In Sec. 3.2 we propose re-weighting the ELBO of the absorbing diffusion model so that the individual loss at each time step is multiplied by $\frac{T-t+1}{T}$ rather than $1/t$. In this section we justify the correctness of this re-weighting by showing it is equivalent to minimising the difference to a forward process that does not have access to $\boldsymbol{x}_t$. As such, the loss takes into account the difficulty of denoising steps and re-weights them down accordingly. This derivation is based on the true ELBO derived by [1]. The loss at time step t can be written as

$$\begin{aligned}\mathcal{L}_t &= D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\ &= \sum_i \sum_j q([\boldsymbol{x}_{t-1}]_{i,j}|\boldsymbol{x}_0) \log \frac{q([\boldsymbol{x}_{t-1}]_{i,j}|\boldsymbol{x}_0)}{p([\boldsymbol{x}_{t-1}]_{i,j}|\boldsymbol{x}_t)},\end{aligned} \tag{1}$$

where the first summation sums over latent coordinates i, and the second summation sums over the probabilities of each code j. For the absorbing diffusion case where tokens in $\boldsymbol{x}_t$ are masked independently and uniformly with probability $\frac{t}{T}$, this posterior is defined as

$$q([\boldsymbol{x}_{t-1}]_i = a|\boldsymbol{x}_0) = \begin{cases} 1 - \frac{t-1}{T}, & \text{if } a = [\boldsymbol{x}_0]_i \text{ and } [\boldsymbol{x}_t]_i = m. \\ \frac{t-1}{T}, & \text{if } a = m \text{ and } [\boldsymbol{x}_t]_i = m. \\ 1, & \text{if } a = [\boldsymbol{x}_0]_i \text{ and } [\boldsymbol{x}_t]_i = [\boldsymbol{x}_0]_i. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The reverse process remains defined in the same way as the standard reverse process:

$$p([\boldsymbol{x}_{t-1}]_i = a|\boldsymbol{x}_t) = \begin{cases} \frac{1}{t} p_\theta([\boldsymbol{x}_0]_i|\boldsymbol{x}_t), & \text{if } a = [\boldsymbol{x}_0]_i \text{ and } [\boldsymbol{x}_t]_i = m. \\ 1 - \frac{1}{t}, & \text{if } a = m \text{ and } [\boldsymbol{x}_t]_i = m. \\ 1, & \text{if } a = [\boldsymbol{x}_0]_i \text{ and } [\boldsymbol{x}_t]_i = [\boldsymbol{x}_0]_i. \end{cases} \tag{3}$$

Substituting these definitions into Eq. (1), the loss can be simplified to Eq. (4); by extracting the constants into a single term out of the sum, C, the loss can be further simplified to obtain Eq. (5), which is equivalent to our proposed reweighted ELBO Eq. (1),

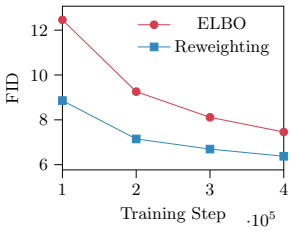
$$\mathcal{L}_t = \sum_i \left[1 \log \frac{1}{1} + \frac{t-1}{T} \log \frac{\frac{t-1}{T}}{1 - \frac{1}{t}} + \left(1 - \frac{t-1}{T} \log \frac{1 - \frac{t-1}{T}}{\frac{1}{t} p_\theta([\boldsymbol{x}_0]_i|\boldsymbol{x}_t)} \right) \right], \tag{4}$$

$$= C - \sum_i \left[\frac{T-t+1}{T} \log p_\theta([\boldsymbol{x}_0]_i|\boldsymbol{x}_t) \right]. \tag{5}$$

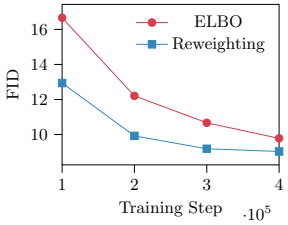
C Additional Comparisons

In Fig. 6 we demonstrated that models trained using our proposed ELBO reweighting converge faster in terms of validation ELBO. To further substantiate this and show that improvements extend to sample quality we compare models trained directly on ELBO and our reweighting in terms of FID in Fig. 1. The same trend is observed, with the models trained on the reweighting converging faster.

Since a key property of DDPMs is that sampling times can be reduced by skipping time steps, in Fig. 2 we compare FID scores for various numbers of sampling steps with a continuous DDPM applied in pixel space [8]. We find that our approach using a discrete DDPM and Vector-Quantized image model degrades in performance at a slower rate than the continuous DDPM likely due to the reduced dimensionality, allowing sampling with fewer steps while maintaining quality. In both cases, the performance for very low numbers of sampling steps could potentially be improved with more sophisticated step selection schemes.



(a) LSUN Churches



(b) FFHQ

Fig. 1: Models trained with our reweighted ELBO converge faster than models trained directly on ELBO.

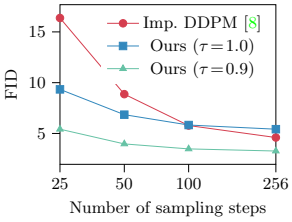


Fig. 2: FID vs number of sampling steps on LSUN Bedroom.

D Concurrent Works

Concurrent with our work, a number of similar approaches independently proposed using diffusion-like models to model VQGAN latents, these approaches are complementary to ours and distinct in a number of ways. VQ-Diffusion [4] use a combination of multinomial and absorbing diffusion to encourage the model to focus less on mask tokens. This, however, requires the use of an additional auxilliary objective function to improve stability, and in practice our approach achieves lower FID on the only shared dataset, FFHQ. MaskGIT [2] models discrete latents by learning to unmask tokens using a similar training scheme to ours; during sampling, tokens are unmasked based on the model's confidence. This approach allows sampling in very few steps, but the lack of theoretical justification makes it unclear how representative samples are. Latent Diffusion [10] relaxes the discrete assumption, using continuous diffusion parameterised by a convolutional U-Net to model latents of greater spatial size, but

with lower dimensional codes. Both compressing spatially/depth-wise and discrete/continuous diffusion come with different trade-offs such as sampling time.

E Nearest Neighbours

When training generative models, being able to detect overfitting is key to ensure the data distribution is well modelled. Overfitting is not detected by popular metrics such as FID, making overfitting difficult to identify in approaches such as GANs. With our approach we are able to approximate the ELBO on a validation set making it simple to prevent overfitting. In this section we demonstrate that our approach is not overfit by providing nearest neighbour images from the training dataset to samples from our model, measured using LPIPS [11].

F Additional Samples

Fig. 6 contains unconditional samples with resolutions larger than observed in the training data from a model trained on LSUN Bedroom.

Fig. 3: Nearest neighbours for a model trained on LSUN Churches based on LPIPS distance. The left column contains samples from our model and the right column contains the nearest neighbours in the training set (increasing in distance from left to right).

Fig. 4: Nearest neighbours for a model trained on FFHQ based on LPIPS distance. The left column contains samples from our model and the right column contains the nearest neighbours in the training set (increasing in distance from left to right).

Fig. 5: Nearest neighbours for a model trained on LSUN Bedroom based on LPIPS distance. The left column contains samples from our model and the right column contains the nearest neighbours in the training set (increasing in distance from left to right).

Fig. 6: Unconditional samples from a model trained on LSUN Bedroom larger than images in the training dataset.

References

1. Austin, J., Johnson, D., Ho, J., Tarlow, D., Berg, R.v.d.: Structured Denoising Diffusion Models in Discrete State-Spaces. arXiv preprint arXiv:2107.03006 (2021) 2
2. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: Masked Generative Image Transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11315–11325 (2022) 3
3. Esser, P., Rombach, R., Ommer, B.: Taming Transformers for High-Resolution Image Synthesis. arXiv:2012.09841 (2021), http://arxiv.org/abs/2012.09841 1
4. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector Quantized Diffusion Model for Text-to-Image Synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10696–10706 (2022) 3
5. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014) 1
6. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable Fidelity and Diversity Metrics for Generative Models. In: International Conference on Machine Learning. pp. 7176–7185 (2020) 2
7. Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating Images with Sparse Representations. arXiv preprint arXiv:2103.03841 (2021) 2
8. Nichol, A.Q., Dhariwal, P.: Improved Denoising Diffusion Probabilistic Models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021) 3

9. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners (2019) 1
10. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022) 3
11. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 4