Unaligned 2D to 3D Translation with Conditional Vector-Quantized Code Diffusion using Transformers

Abril Corona-Figueroa[1] Sam Bond-Taylor[1] Neelanjan Bhowmik[1]
Yona Falinie A. Gaus[1] Toby P. Breckon[1,2] Hubert P. H. Shum[1] Chris G. Willcocks[1]
Department of {[1]Computer Science | [2]Engineering}, Durham University, Durham, UK

The supplementary material for our work is structured as follows: First, Sec. A provides results for our ablation experiments including latent code sizes, codebook length and number of input views. Next, in Sec. B we present additional qualitative results showcasing: MedNeRF comparison, using out-of-distribution inputs, training on binary data, data augmentation details, and sampling using the autoregressive method. Finally, Sec. D includes code snippets for both the conditional diffusion process and for sampling from our model.

A. Ablations

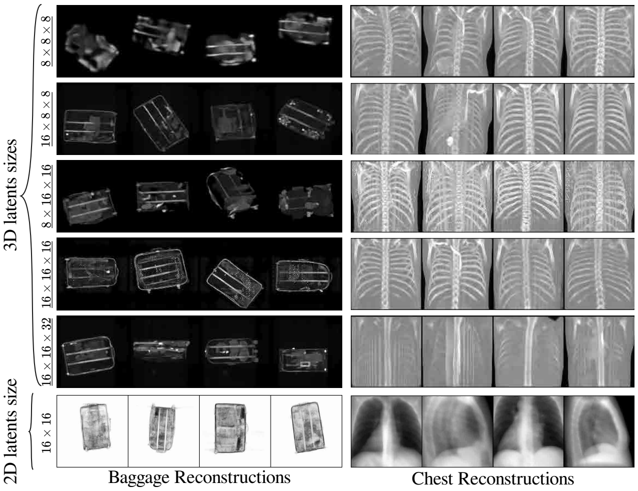


Figure 1. **Comparison of reconstruction quality using different latent code sizes on the two datasets.** 3D latents (ablation) correspond to the 3D VQ-VAE and 2D latents (fixed) to the 2D VQ-VAE. The chest dataset shows faithful reconstruction even with a small size, whereas the baggage security screening dataset requires a more complex latent representation. Larger latent code sizes could allow the model to learn more complex structures (baggage case) or mislead to learn unhelpful features (chest case).

	↓ NLL	
Latent code sizes	Chest	Baggage
8 × 8 × 8	**0.031**	0.0087
16 × 8 × 8	0.032	0.0056
8 × 16 × 16	0.029	0.0055
16 × 16 × 16	0.032	**0.0044**
16 × 16 × 32	0.067	0.0056

Table 1. Quantitative evaluation based on validation Negative Log-Likelihood (NLL) using different latent code sizes for the discrete representations learned by the VQ-VAEs in Stage 1 of our approach.

Codebook, $\mathrm{argmin}(\boldsymbol{x}, \mathcal{B})$			
Length	Chest (10^{-3})	Length	Baggage (10^{-5})
64	4.01	1024	5.8
128	0.85	2048	4.2
512	0.65	**4096**	1.7

Table 2. Quantitative evaluation based on the codebook loss using different codebook lengths for the discrete representations learned by the VQ-VAEs in Stage 1 of our approach. Our approach allows learning complex data distributions (e.g. baggage security dataset) by increasing the length. For our main experiments we used a length of 512 and 4,096 for the chest and baggage dataset, respectively.

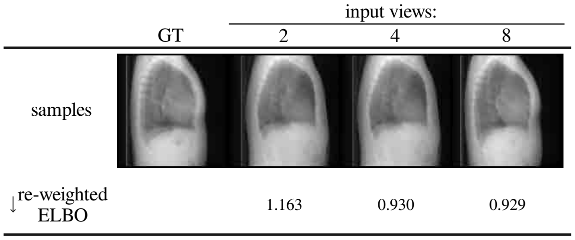


Figure 2. **Ablation using different number of input 2D views for conditional 3D modeling on the LIDC-IDRI (chest) dataset.** While increasing the number of inputs views from 2 to 4 brings additional performance, further increments don't necessarily result in linear returns. For our main experiments we used only 2 inputs views for both datasets.

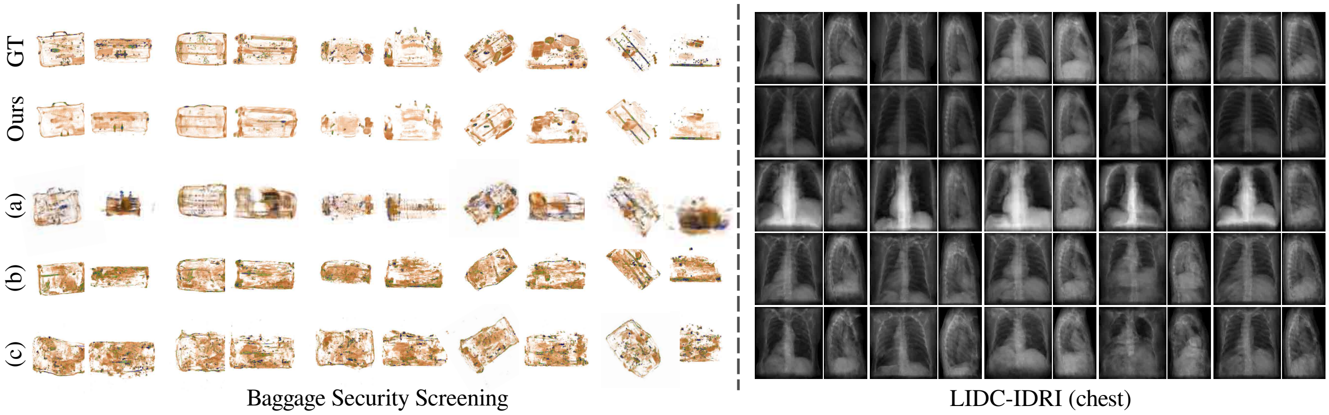
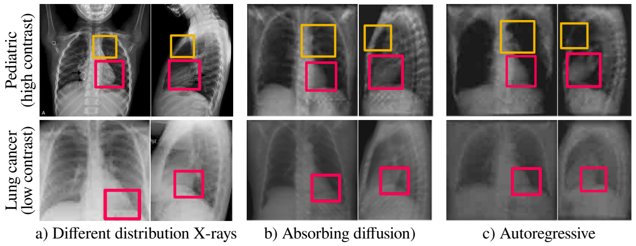


Figure 3. Comparison of Biplanar Maximum Intensity Projections (MIP) on the baggage security screening dataset and on LIDC-IDRI (chest) dataset. (a) denotes the MedNeRF model, (b) CCX-rayNet and (c) X2CT-GAN.



~ time per 1k iter.	0.0864 hs	1.086 hs
coverage	0.97	0.90

Figure 4. **Absorbing diffusion vs autoreggressive samples on out-of-distribution input X-rays.** We test our model on different distributions including pediatric and lung cancer X-rays with different radiographic contrast (high and low contrast).

B. Additional Results

2D Comparison Fig. 3 contains qualitative results for both datasets of the 2D evaluation that also includes the MedNeRF method, which renders 3D-aware CT projections. This evaluation consists of comparing Maximum Intensity Projections from the spatial dimensions. Our method allows faithful samples compared to competing methods. Specifically, our samples better highlight denser objects which are of particular interest in the detection of prohibited items.

Diffusion comparison We test our model robustness on out-of-distribution input X-rays and generate samples comparing absorbing diffusion and the autoregressive method using the same transformer architecture. Despite domain differences, our model is able to generate accurate samples without requiring any kind of domain supervision, suggesting that our learned discrete representations achieve effective compression to remain invariant to low-level features like contrast while encoding essential structures such as bone and soft tissue (Fig. 4).

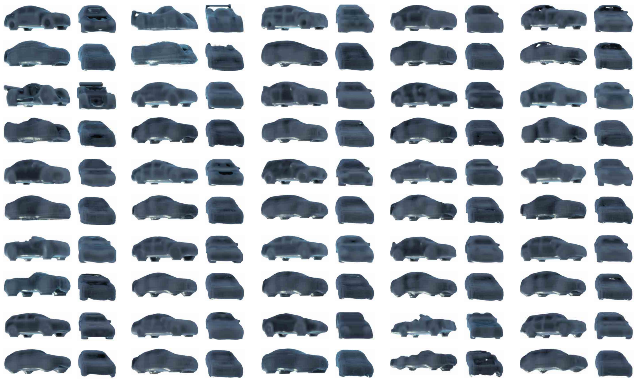


Figure 5. **Visualization of additional results on ShapeNet [3].** Training on binary volumetric data in contrast to continuous intensities isn't trivial as it might lead to instabilities in architectures like simple CNN-based GANs. Voxels with values other than 0 or 1 are flagged as fake by the discriminator, hindering preventing continuous optimization [1]. However, our model can effectively learn this type of data as our approach doesn't (necessarily) rely on adversarial training.

C. Controlled Data Augmentations

To avoid overfitting, we extended the stochastic discriminator augmentation framework from StyleGAN2-ADA [2] to handle 3D data. This solution involves augmenting both real and generated data by the VQ-VAE using a set of style and spatial augmentations with a probability $p < 1$. Unlike other data augmentation strategies, non-invertible augmentations can be incorporated with an adaptive p-value based on an overfitting heuristic. We found that this prevents the discriminator becoming more confident, and thus both real and fake predictions take more time to diverge. As a result, the VQ-VAEs to learn richer representations while delaying the drop in its the validation accuracy. Note that our approach doesn't rely on adversarial training, thus, the incorporation of a discriminator is optional.

D. Example Code

We include python-like code for training a conditional absorbing diffusion process Fig. 6a, and sampling from our 2D to 3D translation model Fig. 6b. The use of a Transformer allows the learned distribution to be easily conditioned on arbitrary input shapes, by simply concatenating the conditioning signal with the noisy data. The linear masking schedule allows sampling with smaller numbers of steps, to speed the process up, by simply passing in a smaller value for T.

```
def diffusion_training_loss(c_0, Z, T, mask_id):
    c_t, b = c_0.clone(), c_0.size(0)
    # Randomly sample diffusion time steps
    t = torch.randint(1, T+1, (b,))
    # Randomly mask tokens with probability t/T
    mask = torch.rand_like(c_0) < (t / T)
    c_t[mask] = mask_id
    # Calculate p(c_0 | c_t, Z)
    logits = Transformer(torch.cat(Z, c_t))
    # Calculate reweighted ELBO loss
    loss = cross_entropy(logits, c_0) * (T-t+1)/T
    return loss
```

(a) Python-like code snippet for training a conditional Absorbing Diffusion model.

```
def sample(imgs_2d, T, mask_id, latent_size):
    b = imgs_2d.size(0)
    # Compress 2D images with 2D VQ-Encoder
    Z = vqae_2d.encoder(imgs_2d)
    # Initialise 3D latents with all masks
    c_t = torch.full((b, latent_size), mask_id)
    # Track which latents have been unmasked
    unmasked = torch.zeros_like(c_t, dtype=bool)

    # Loop over sampling steps
    for t in reversed(list(range(1, T+1))):
        # Randomly choose where to unmask
        changes = torch.rand(c_t.shape) < 1/t
        # Don't unmask anywhere already unmasked
        changes = torch.bitwise_xor(changes, \
            torch.bitwise_and(changes, unmasked))
        # Update unmasked
        unmasked = torch.bitwise_or(unmasked, changes)

        # Sample from p(c_{t-1} | c_t, Z)
        logits = Transformer(torch.cat(Z, c_t))
        dist = Categorical(logits)
        c_0_hat = dist.sample()
        c_t[changes] = c_0_hat[changes]

    # Decompress 3D latents with 3D VQ-Decoder
    imgs_3d = vqae_2d.decoder(c_t)
    return imgs_3d
```

(b) Python-like code snippet for 2D to 3D translation using our approach.

References

[1] Zhiqin Chen, Vladimir G. Kim, Matthew Fisher, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. Decor-gan: 3d shape detailization by conditional refinement. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 15740–15749, June 2021. 2

[2] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Adv. in Neural Inf. Process. Syst.*, volume 33, pages 12104–12114. Curran Associates, Inc., 2020. 2

[3] Manolis Savva, Fisher Yu, Hao Su, Masaki Aono, Baoquan Chen, Daniel Cohen-Or, Weihong Deng, Hang Su, Song Bai, Xiang Bai, et al. Large-scale 3d shape retrieval from shapenet core55. In *Proc. of the Eurographics 2016 Workshop on 3D Object Retrieval*, pages 89–98, 2016. 2