A ILLUSTRATION: PRIVACY-SENSITIVE TRANSLATION

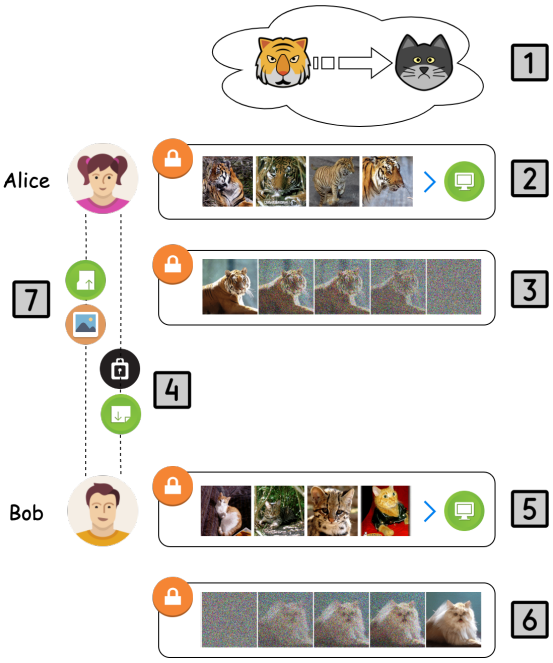

Figure 5

Alice is the owner of the source (tiger) domain, and Bob is the owner of the target (cat) domain. Alice intends to translate tiger images to cat images, but in a privacy-sensitive manner without releasing the source dataset. Bob does not wish to make the cat dataset public, either.

Fig. 5 illustrates the process of privacy-sensitive domain translation. The process contains the following steps, with indexes in the figure.

1. Alice intends to translate tiger images to cat images.
2. Alice trains a diffusion model with the source tiger images.
3. Alice uses the pretrained, tiger diffusion model to convert a source tiger image to its latent code.
4. Alice sends the latent code to Bob.
5. Bob similarly trains a diffusion model on the cat domain.
6. Bob uses the pretrained, cat diffusion model to convert the received latent code to a cat image.
7. Bob then sends the translated image back to Alice.

Clearly, during the translation process, only the latent code and the translated cat image are transmitted via the public channel, while both source and target datasets are private to the two parties. This is a significant advantage of DDIBs over alternate methods, as we enable strong privacy protection of the datasets.

B DETAILS OF SGM TRAINING AND DDIM ODE SOLVER

B.1 TRAINING SCORE NETWORKS

While the description in Section 2 is based on continuous SDEs, actual implementations of diffusion models use discrete time steps. Given samples from a data distribution $q(\mathbf{x}_0)$, diffusion models attempt to learn a model distribution $p_\theta(\mathbf{x}_0)$ that approximates $q(\mathbf{x}_0)$, and is easy to sample from. Specifically, diffusion probabilistic models are latent variable models of the form

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T})\, \mathrm{d}\mathbf{x}_{1:T}\,, \text{ where } p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_T$ are latent variables in the same sample space as $\mathbf{x}_0$. The parameters θ are trained to approximate the data distribution $q(\mathbf{x}_0)$, by maximizing a variational lower bound:

$$\max_\theta \mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \leq \max_\theta \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)}[\log p_\theta(\mathbf{x}_{0:T}) - \log q(\mathbf{x}_{1:T}|\mathbf{x}_0)]$$

where $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is some inference distribution over the latent variables. It is known that when the conditional distributions are modeled as Gaussians with trainable mean functions and fixed variances, the above objective can be simplified to:

$$L(\epsilon_\theta) := \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon_\theta^{(t)}(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon_t) - \epsilon_t \right\|_2^2 \right]$$

The resulting noise prediction functions $\epsilon_\theta^{(t)}$, are equivalent to the score networks $\mathbf{s}_{t,\theta}$ mentioned in Section 2 due to Tweedie's formula (Stein, 1981; Efron, 2011). For details, we refer the reader to Ho et al. (2020); Song et al. (2020a).

B.2 DDIM ODE SOLVER

With a trained noise prediction model $\epsilon_\theta^{(t)}(\mathbf{x})$, the DDIM iterate between adjacent variables $\mathbf{x}_{t-\Delta t}$ and $\mathbf{x}_t$, considered in Song et al. (2020a), assumes the following form:

$$\frac{\mathbf{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}} = \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}} \right) \epsilon_\theta^{(t)}(\mathbf{x}_t)$$

In our experiments, we implement the above equation between adjacent diffusion steps. The equation is deterministic, and can be considered as a Euler method over the following ODE:

$$\mathrm{d}\bar{\mathbf{x}}(t) = \epsilon_\theta^{(t)} \left(\frac{\bar{\mathbf{x}}(t)}{\sqrt{\sigma^2 + 1}} \right) \mathrm{d}\sigma(t) \tag{9}$$

where we adopt the reparameterization:

$$\sigma(t) = \sqrt{\frac{1-\alpha(t)}{\alpha(t)}}, \quad \bar{\mathbf{x}}(t) = \frac{\mathbf{x}(t)}{\sqrt{\alpha(t)}}$$

Importantly, the ODE in Eq. (9) with the optimal model $\epsilon_\theta^{(t)}(\mathbf{x})$, has an equivalent probability flow ODE corresponding to the "Variance-Exploding" SDE in Song et al. (2020b).

C LIMITATIONS OF OPTIMAL TRANSPORT-BASED TRANSLATION

DDIBs contain deterministic bridges between distributions, and are a form of entropy-regularized optimal transport. The learned diffusion models can be effectively considered as a digest or summary of the datasets. While doing translation, they attempt to create images in the target domain, that are closest in optimal transport distances to the source images. Such OT-based process is both an advantage and a limitation of our method.

In ImageNet translation, when the source and target datasets are similar, DDIBs are generally able to identify correct animal postures. For example, we have shouting lions and tigers, because these animals have similar behaviors that are observed in the datasets and then internalized by DDIBs. However, in datasets that are less similar (*e.g.* birds and dogs), DDIBs sometimes fail to produce translation results that retain the postures precisely. We encountered significantly less such cases in AFHQ translation, since the dataset is more standardized and homogeneous.

Fig. 6 illustrates the optimal transport mappings among images as well as some failure cases. Clearly, the translation processes flowing from left to right minimize the Euclidean transportation distances between images. Some of these translated samples may be classified "failure cases" in actual user studies. Such are considered both a feature and a limitation of DDIBs.

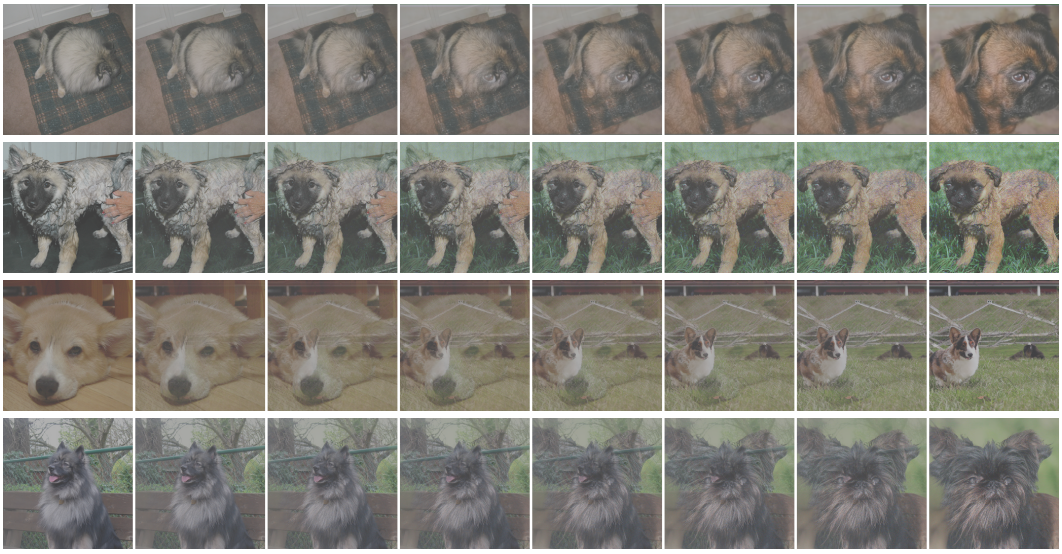

Figure 6: Optimal transport translation processes in DDIBs. *(Leftmost)* Source images. *(Rightmost)* Translated images.

D PROOF OF PROPOSITION 3.2

Proof. The proof proceeds by substituting the values of $(\mathbf{z}_t, \hat{\mathbf{z}}_t) = (0, g(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}))$ into Eq. (6),

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) + g(t)\,\mathbf{z} - \frac{1}{2} g(t)(\mathbf{z} + \hat{\mathbf{z}})\right] \mathrm{d}t \tag{10}$$

$$= \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t \tag{11}$$

This is exactly Eq. (2). □

E ADDITIONAL EXPERIMENTAL DETAILS

E.1 OPTIMAL TRANSPORT IN PAIRED DATASETS

Color Conversion In Fig. 7, a simple examination of the original and segmentation images reveals significant differences in color configurations. In the Maps dataset, while the real, satellite images are composed of dark colors, the segmentation images are light-toned. The same observation applies to other datasets. The shark contrasts in colors intuitively present a large transportation cost, that probably hinders the progress of DDIBs, as we have demonstrated its relationship to OT in Section 3.

To facilitate the workings of DDIBs, we follow a heuristic to transform the colors of the segmentation images. Specifically, on a small subset of the train dataset, we run an OT algorithm to compute a color correspondence that minimizes the color differences in terms of Sinkhorn distances between the real and segmentation images. The segmentation (target) datasets undergo this color conversion before they are fed into a diffusion model for training. During evaluation, when we compute MSEs, the images are converted to the original color space.

Privacy Protection Color conversion requires considering both datasets jointly to compute a color mapping, and seems to betray the original purpose of DDIBs on protection of dataset privacy. We comment that the amount of leaked information is minimal: for example, to compute a color correspondence for the Maps dataset, we sampled only around 1000 pixels from the two datasets, to summarize the color composition information. DDIBs still conserve privacy at large.

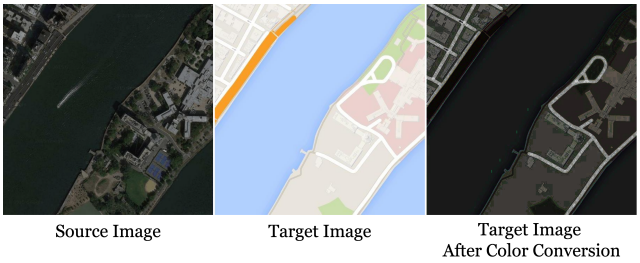

Figure 7: **Color Conversion**. In the paired translation tasks, we are given the real and segmentation images. Before training the diffusion models, we first transform the segmentation images to a color palette that is closer to the real images. While evaluating MSEs, we convert the images back to the original colors.

E.2 EXAMPLE-GUIDED COLOR TRANSFER

We present additional qualitative comparison between DDIBs and common OT methods, in Fig. 8.

Figure 8: Full color transfer results on example images.