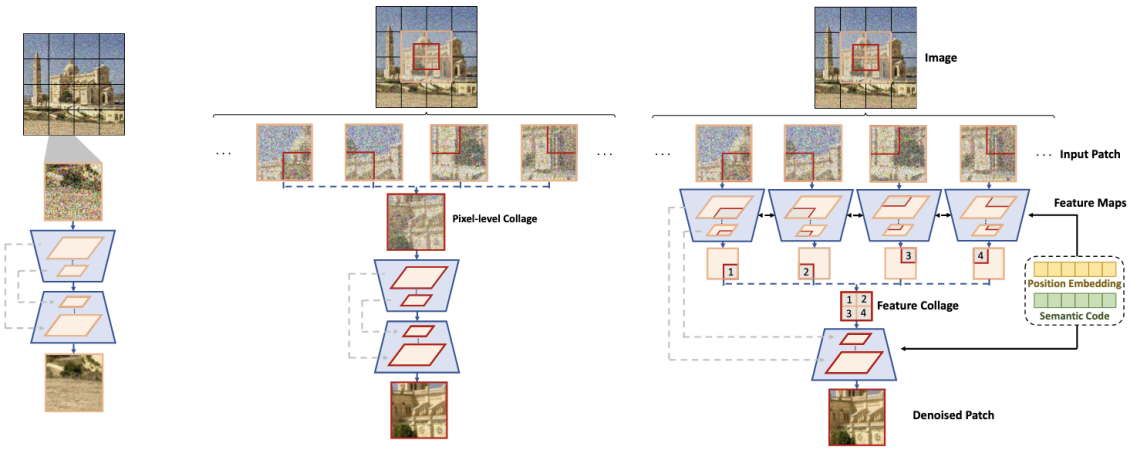
Patch DM

A Preprint

GANs are hard to train; VAE output is blurry; Diffusion models booming recently, however problems occur as we scale diffusion models as they operate directly of image pixel space and also multistep(1000 steps which means 1000 operations on whole image which may be expensive in terms of big resolution). Authors develop a new denoising diffusion model based on patches, Patch-DM, to generate images of highresolutions. Patch-DM can perform direct highresolution image synthesis without introducing boundary artifacts.



(a) No Collage on Patches (b) Patch Collage in Pixel Space (c) Patch Collage in Feature Space (Ours)

igure 2: **Patch Generation For Image Synthesis.** (a) shows a very basic method of patch-wise image synthesis by simply splitting e images and generating patches independently. This method brings severe border artifacts. (b) alleviates the border artifacts by using ifted windows while generating images and doing patch collage in pixel space. (c) is our proposed method which collages the patches in e feature space. The features for neighboring features will be split and collaged for a new patch synthesis. We will show this method is a y design for us to generate high-quality images without border artifacts.

Denoising diffusion models synthesize images from Gaussian noise by iteratively learning a denoising function. Rather than mapping noise to data in a single step, the generation process is divided into T timesteps. The forward (noising) process incrementally adds noise to an image x_0, while the model learns to reverse this process starting from $x_T \sim \mathcal{N}(0, I)$.

The forward transition is defined as:

$$x_t \sim \mathcal{N}(x_{t-1}; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{1}$$

with hyperparameters β_t controlling the noise level. This formulation allows expressing x_t directly as:

$$x_t \sim \mathcal{N}(x_0; \sqrt{\alpha_t}x_0, (1-\alpha_t)I), \tag{2}$$

where $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$.

To reverse the process, the model predicts noise $\hat{\epsilon}_t = f_\theta(x_t, t)$, and samples x_{t-1} as:

$$x_{t-1} \sim \mathcal{N}\left(\frac{1}{\sqrt{1-\beta_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\hat{\epsilon}_t\right), \sigma_t^2 I\right), \tag{3}$$

where σ_t controls the sampling variance. The training objective minimizes the L_2 loss $\|\epsilon_t - \hat{\epsilon}_t\|^2$ between the true noise and predicted noise.

This approach relies on pixel-space denoising, which becomes computationally intensive at high resolutions.

The training image from the dataset is $x_0 \in \mathbb{R}^{C \times H \times W}$. x_0 is split into $x_0^{(i,j)}$ where i,j s the row and column number of the patch. Instead of generating full x_0, PatchDM generates $x_0^{(i,j)}$ and then concatenates. Authors suggest performing collage not on pixel space, but on feature space so no artefacts on borders occur.

$[z_1^{(i,j)}, z_2^{(i,j)} \dots z_n^{(i,j)}] = f_\theta^E(x_{i,j}; t)$ where f_θ^E is UNET encoder and $z_{\dots}$ are the internal feature maps. Later, these maps are split into shift patches and collage: $\hat{z}_k^{'(i,j)} = [P_1(z_k^{(i,j)}), P_2(z_k^{(i,j+1)}), P_3(z_k^{(i+1,j)}), P_4(z_k^{(i+1,j+1)})]$. Then UNet encoder is used to get the predicted shift patch noise. In order to make the model generate more semantically consistent images, authors also add position embedding and semantic embedding to the model so that f_θ takes 2 more inputs: $\mathcal{P}(i,j)$ and $\mathcal{E}(x_0)$ respectively. The pretrained model for obtaining the image embeddings is CLIP. Authors resize the images to 224×224 and send them to ViT-B/16 to obtain the features as global conditions, then optimize these global conditions directly.