# Supplementary Materials for EBDM: Exemplar-guided Image Translation with Brownian-bridge Diffusion Models.

Eungbean Lee[1], Somi Jeong[2], and Kwanghoon Sohn[1,3]

[1] Yonsei University, Seoul, Korea
{eungbean,khsohn}@yonsei.ac.kr
[2] NAVER LABS
somi.jeong@naverlabs.com
[3] Korea Institute of Science and Technology (KIST), South Korea

This supplementary material provides details that are not included in the main paper due to space limitations. We provide the explanation of deduction details at Sec. 1 and advantages over DDPMs Sec. 2. Then the implementation details of EBDM will be presented at Sec. 3. Finally, we will present more qualitative experiment results.

## 1 Brownian Bridge Diffusion Models

In this section, we provide more details of Brownian Bridge Diffusion Models (BBDM) [2]. The BBDM aims to connect two image domains via discrete Brownian bridges. Assuming that the start point and end point of the diffusion process, $(\boldsymbol{x}_0, \boldsymbol{x}_T) = (\boldsymbol{x}, \boldsymbol{y}) \sim q_{data}(\boldsymbol{x}, \boldsymbol{y})$, BBDM learns to approximately sample from $q_{data}(\boldsymbol{x}|\boldsymbol{y})$ by reversing the diffusion bridge with boundary distribution $q_{data}(\boldsymbol{x}, \boldsymbol{y})$, given a training set of paired samples drawn from $q_{data}(\boldsymbol{x}, \boldsymbol{y})$.

### 1.1 Forward Process

Given initial state $\boldsymbol{x}_0$ and destination state $\boldsymbol{y}$, the forward diffusion process of the Brownian Bridge can be defined as:

$$\begin{aligned} p\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{x}_T\right) &= \mathcal{N}\left(\boldsymbol{x}_t; (1-m_t)\boldsymbol{x}_0 + m_t\boldsymbol{y}, \boldsymbol{\delta}_t\boldsymbol{I}\right) \\ \text{where} \quad m_t = \frac{t}{T}, &\quad \delta_t = 2s(m_t - m_t^2) \end{aligned} \tag{10}$$

where $T$ is the total steps of the diffusion process, $s$ is the variance factor, and $\delta_t$ is the variance that is designed to preserve the maximum at $t = 2/T$ as identity, *i.e.* $\delta_{max} = \frac{1}{2}$. The variance factor $s$ scales the maximum variance to control diffusion diversity, and we set $s = 1$ as the default. The intermediate state $\boldsymbol{x}_t$ in its discrete form can be determined by calculating:

$$\boldsymbol{x}_t = (1 - m_t)\,\boldsymbol{x}_0 + m_t\boldsymbol{y} + \sqrt{\delta_t}\boldsymbol{\epsilon}_t \quad \text{where} \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{11}$$

We can express $\boldsymbol{x}_0$ with $\boldsymbol{x}_t$ and Eq. (11):

$$\boldsymbol{x}_0 = \frac{1}{1-m_t}\left(\boldsymbol{x}_t - m_t\boldsymbol{y} - \sqrt{\delta_t}\boldsymbol{\epsilon}_t\right) \tag{12}$$

Thus, the transition probability $q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}\right)$ can be derived by substituting the expression of Eq. (11) and Eq. (12):

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}\right) = \mathcal{N}(\boldsymbol{x}_t;\ \hat{\mu}_t(\boldsymbol{x}_{t-1}, \boldsymbol{y})\ , \hat{\delta}_t\boldsymbol{I}) \tag{13}$$

$$\begin{aligned} \text{where,} \quad \hat{\mu}_t(\boldsymbol{x}_{t-1}, \boldsymbol{y}) = &\ \frac{1-m_t}{1-m_{t-1}}\boldsymbol{x}_{t-1} + \left(m_t - \frac{1-m_t}{1-m_{t-1}}m_{t-1}\right)\boldsymbol{y} \\ \hat{\delta}_t = \delta_{t|t-1} &= \delta_t - \delta_{t-1}\frac{(1-m_t)^2}{(1-m_{t-1})^2} \end{aligned} \tag{14}$$

### 1.2 Reverse Process

The reverse process of BBDM is to predict $\boldsymbol{x}_{t-1}$ given $\boldsymbol{x}_t$:

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}\right) = \mathcal{N}\left(\boldsymbol{x}_{t-1};\ \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t, t\right),\ \tilde{\delta}_t\boldsymbol{I}\right) \tag{15}$$

where $\boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t, t\right)$ represents the predicted mean, and $\tilde{\delta}_t$ denotes the variance of the noise at each step.

### 1.3 Training Objectives

The training procedure involves optimizing the Evidence Lower Bound (ELBO) for the Brownian Bridge diffusion process, which is expressed as:

$$\begin{aligned} ELBO = -\,\mathbb{E}_q\big(&\ D_{KL}\left(q\left(\boldsymbol{x}_T \mid \boldsymbol{x}_0, \boldsymbol{y}\right) \| p\left(\boldsymbol{x}_T \mid \boldsymbol{y}\right)\right) \qquad \because \boldsymbol{x}_T = \boldsymbol{y} \\ &+ \sum_{t=2}^{T} D_{KL}\left(q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}\right) \| p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{y}\right)\right) \\ &- \log p_\theta\left(\boldsymbol{x}_0 \mid \boldsymbol{x}_1, \boldsymbol{y}\right)\big) \end{aligned} \tag{16}$$

By combining Eq. (13) and Eq. (14), the formula $q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}\right)$ in the second term can be derived from Bayes' theorem and the Markov chain property:

$$\begin{aligned} q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}\right) &= \frac{q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}\right) q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{y}\right)}{q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{y}\right)} \\ &= \mathcal{N}\left(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}\right), \tilde{\delta}_t\boldsymbol{I}\right) \end{aligned} \tag{17}$$

The mean value term $\tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}\right)$ can be reformulated as $\tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_0, \boldsymbol{y}\right)$ by utilizing reparameterization method [1]:

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_t\left(\boldsymbol{x}_t, \boldsymbol{y}\right) &= c_{xt}\boldsymbol{x}_t + c_{yt}\boldsymbol{y} + c_{\epsilon t}\left(m_t\left(\boldsymbol{y}-\boldsymbol{x}_0\right) + \sqrt{\delta_t}\boldsymbol{\epsilon}\right) \\
\text{where,} \quad c_{xt} &= \frac{\delta_{t-1}}{\delta_t}\frac{1-m_t}{1-m_{t-1}} + \frac{\hat{\delta}_t}{\delta_t}\left(1-m_{t-1}\right) \\
c_{yt} &= m_{t-1} - m_t\frac{1-m_t}{1-m_{t-1}}\frac{\delta_{t-1}}{\delta_t} \\
c_{\epsilon t} &= \left(1-m_{t-1}\right)\frac{\hat{\delta}_t}{\delta_t}
\end{aligned} \tag{18}$$

And the variance term is:

$$\tilde{\delta}_t = \frac{\hat{\delta}_t \cdot \delta_{t-1}}{\delta_t} \tag{19}$$

As the neural network $\boldsymbol{\epsilon}_\theta$ predict the noise, thus, the reverse process Eq. (15) can be reformulated as:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_t, \boldsymbol{y}, t\right) = c_{xt}\boldsymbol{x}_t + c_{yt}\boldsymbol{y} + c_{\epsilon t}\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, t\right) \tag{20}$$

Therefore, the training objective ELBO in Eq. (16) can be simplified as:

$$\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{y},\boldsymbol{\epsilon}}\left[c_{\epsilon t}\left\|m_t\left(\boldsymbol{y}-\boldsymbol{x}_0\right) + \sqrt{\delta_t}\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, t\right)\right\|^2\right] \tag{21}$$

The weighting function $c_{xt}, c_{yt}, c_{\epsilon t}$ in Eq. (18) is used in Eq. (7) and (8).

## 2 Advantages over DDPMs.

The primary motivation for choosing BBDMs is to 1) reduce the number of conditions and 2) utilize an end-to-end training framework.

**1) Reducing the number of conditions.** The primary motivation for choosing the Brownian Bridge is to simplify the conditioning mechanism. By reducing the number of conditions, thereby minimizing the parameters, training times, and the risk of overfitting, while enhancing robustness. Increasing the number of conditions $\boldsymbol{c} = \{c_1, \cdots, c_n\}$ significantly impacts both training and performance. The conditional distribution (Eq. 22), and reverse process (Eq. 23) can be described as:

$$P(x|\boldsymbol{c}) = \frac{P(x)}{P\left(c_1, \ldots, c_n\right)}\prod_{i=1}^{N} P\left(c_i \mid x\right) \propto \prod_{i=1}^{N}\frac{P\left(x \mid c_i\right)}{P(x)} \tag{22}$$

$$p_\theta\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}\right) := \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t), \Sigma_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t)) \tag{23}$$

As the number of conditions $n$ grows, the loss function becomes more complex affecting the modeling the $\mu_\theta$ and $\Sigma_\theta$. This complexity can be quantified by the KL divergence between the true conditional distribution and the model distribution, indicating a more complex distribution that the model must learn to approximate accurately, leading to convergence difficulties, gradient instability, and the need for stronger regularization techniques. Simplifying the conditioning mechanism mitigates these issues by:

- **Reducing parameters:** Lower dimensionality in the conditional space decreases the number of parameters, leading the optimization landscape less complex.
- **Reduced demand for Data Requirement:** Less data is needed to cover the distributions with the same density due to the curse of dimensionality.
- **Less training time:** The Computational costs are reduced as the complexity of computing the gradients reduces.
- **Lower risk of overfitting**. A simpler model is less likely to capture noise and specific characteristics of the training data, due to the variance of the function $\boldsymbol{\epsilon}_\theta(\boldsymbol{c}, t)$ increases, adversely affecting generalization and stable training.

**2) End-to-end training:** SD-based method takes modular approaches[4] that are not trained end-to-end, posing a risk of unwanted information influencing the inference. In contrast, our method benefits from an end-to-end training framework, enhancing integration and performance, particularly in exemplar-guided image translation tasks.

[4] *e.g.* Stable Diffusion equipped with ControlNet and IP-Adapter.

| model | z-shape | channels | channel multiplier | attention resolutions | total parameters | trainable parameters |
|---|---|---|---|---|---|---|
| BBDM-f4 | $64 \times 64 \times 3$ | 128 | $1, 4, 8$ | $32, 16, 8$ | 437.81M | 382.49M |
| Exemplar Net | $64 \times 64 \times 3$ | 128 | $1, 4, 8$ | $32, 16, 8$ | 404.82M | 382.48M |
| Global Encoder | - | - | - | - | 86.58M | 0 |
| EBDM-f4 | $64 \times 64 \times 3$ | 128 | $1, 4, 8$ | $32, 16, 8$ | 929.21M | 764.97M |

**Table 5:** Network hyperparameters for EBDM and modules.

## 3 More Experiment Details

In this section, further implementation specifics of the EBDM are elucidated, encompassing network hyperparameters (Tab. 5), optimization strategies, as well as computational efficiency.

### 3.1 Datasets

We conduct three tasks to evaluate our model: Edge-to-photo, mask-to-photo, and pose-to-photo. For mask-guided and edge-guided image generation tasks, the CelebA-HQ [4] dataset is used and we construct the edge maps using the Canny edge detector following [7, 9]. For the pose-guided image generation task, we use deepfashion [3] dataset that consists of $52,712$ images with a keypoints annotation. The split of train and validation pairs is consistent with CoCosNet [7] policies.

### 3.2 Training

All experiments are conducted utilizing a spatial resolution of $64 \times 64$ within the latent space. During training, we use a batch size of 8 with gradient accumulation 2, each batch containing pairs of an input exemplar and condition following [7]. The model is trained with `AdamW` optimizer for the learning rate of 1.0e$-$5 and learning rate decay with $\gamma = 0.2$. The Exponential Moving Average (EMA) was adopted in the training procedure together with `ReduceLROnPlateau` learning rate scheduler. Training is done on `Pytorch` framework and Nvidia RTX A6000 48GB GPU.

### 3.3 Autoencoders

We adopt the pretrained VQGAN presented in [5], which reduces images to $64 \times 64$ resolution in latent space. In edge-to-photo and mask-to-photo tasks using CelebA-HQ [4], we use VQ-regularized autoencoder with downsampling factor $f = 4$ and channel dimension 3. For the pose-to-photo task using DeepFashion [3], we use KL-regularized autoencoder with downsampling factor $f = 8$ and channel dimension 4. Both the encoder and decoder are frozen during training for fair comparison.

| Methods | FLOPs (1 steps) | FLOPs (50 steps) | # Parameters |
|---|---|---|---|
| SD-based | 11.14 T | 86.08 T | 1308.7 M |
| Ours | 11.37 T (+2.0%) | 61.72 T (-28.21%) | 764.97 M (-41.55%) |

**Table 6: Comparisions of computational costs.** Number of parameters and FLOP counts with single and 50 steps in inference.

### 3.4 Computational Efficiency

Our method improves computational cost, as demonstrated in Tab. 6. Our method achieves a -28.21% reduction of FLOPs indicating faster inference time. Furthermore, in the inference stage, the SD-based model requires extensive grid searches across the conditional parameters (*e.g.* guidance scale, control weight, IP-adapter scale, *etc.*) to achieve plausible results, which consumes significant resources. By reducing the number of conditions, our method improves efficiency in both computational and practical uses.

### 3.5 Additional Qualitative Results

Lastly, we present further qualitative results in comparison with other techniques in Figs. 7 to 9. Additional diverse samples with various control inputs are shown in Figs. 10 to 12.

## 4 Limitations

Our approach utilizes the Brownian Bridge diffusion process in latent space [5] to connect control and image latents effectively. However, the pre-trained VAE Encoder that focuses on image representation limits its ability to process control signals accurately, especially when differentiating semantically diverse elements (such as background and face in mask), focusing more on color distance rather than semantic discrepancies.

To mitigate this, prior studies [6,8] have introduced additional control guiders. Yet, integrating these with the Brownian Bridge model, characterized by its reliance on two fixed endpoints, complicates the direct integration of such solutions.


**Fig. 7: Mask-to-image** Qualitative comparisons on the CelebAHQ-HQ Dataset.


**Fig. 8: Edge-to-image** Qualitative comparisons on the CelebA-HQ Dataset.


**Fig. 9: Pose-to-image** Qualitative comparisons on the DeepFashion Dataset.


**Fig. 10: Mask-to-image** on the CelebAHQ Dataset.


**Fig. 11: Edge-to-image** on the CelebA-HQ Dataset.


**Fig. 12: Pose-to-image** on the DeepFashion Dataset.

## References

1. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
2. Li, B., Xue, K., Liu, B., Lai, Y.K.: Bbdm: Image-to-image translation with brownian bridge diffusion models. In: CVPR (2023)
3. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR. pp. 1096–1104 (2016)
4. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset. Retrieved August **15**, 11 (2018)
5. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
6. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
7. Zhang, P., Zhang, B., Chen, D., Yuan, L., Wen, F.: Cross-domain correspondence learning for exemplar-based image translation. In: CVPR. pp. 5143–5153 (2020)
8. Zhao, S., Chen, D., Chen, Y.C., Bao, J., Hao, S., Yuan, L., Wong, K.Y.K.: Uni-controlnet: All-in-one control to text-to-image diffusion models. In: NeurIPS (2024)
9. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Cocosnet v2: Full-resolution correspondence learning for image translation. In: CVPR. pp. 11465–11475 (2021)