DPBridge: Latent Diffusion Bridge for Dense Prediction
(*Supplementary Materials*)

A. Derivation Details

In this section, we'll provide derivation details of the maximum likelihood loss presented in Section 4.4. Its core is the posterior distribution $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0, \mathbf{z}_T)$ as in Equation (5), whose calculation involves three terms: $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_T)$, $q(\mathbf{z}_{t-1}|\mathbf{z}_0, \mathbf{z}_T)$, and $q(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_T)$. The last two terms of the posterior can be retrieved directly through the marginal distribution of the diffusion bridge's forward process, but further derivation is required for the first term. Specifically, as specified in Equation (3):

$$\begin{aligned} q(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_T) &= \mathcal{N}(m_t\mathbf{z}_0 + n_t\mathbf{z}_T, \sigma_t^2\mathbf{I}) \\ \mathbf{z}_t &= m_t\mathbf{z}_0 + n_t\mathbf{z}_T + \sigma_t\boldsymbol{\epsilon}_t \end{aligned} \tag{12}$$

$$\begin{aligned} q(\mathbf{z}_{t-1}|\mathbf{z}_0, \mathbf{z}_T) &= \mathcal{N}(m_{t-1}\mathbf{z}_0 + n_{t-1}\mathbf{z}_T, \sigma_{t-1}^2\mathbf{I}) \\ \mathbf{z}_{t-1} &= m_{t-1}\mathbf{z}_0 + n_{t-1}\mathbf{z}_T + \sigma_{t-1}\boldsymbol{\epsilon}_{t-1} \end{aligned} \tag{13}$$

To derive the transition probability $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_T)$, we substitute $\mathbf{z}_0$ in Equation (12) with derivation from Equation (13), obtain

$$\begin{aligned} \mathbf{z}_t &= a_t\mathbf{z}_{t-1} + b_t\mathbf{z}_T + \delta_t\boldsymbol{\epsilon} \\ a_t &= \frac{m_t}{m_{t-1}} \\ b_t &= n_t - \frac{m_t}{m_{t-1}}n_{t-1} = n_t - a_t n_{t-1} \\ \delta_t^2 &= \sigma_t^2 - \frac{m_t}{m_{t-1}}n_{t-1}^2\sigma_{t-1}^2 = \sigma_t^2 - a_t^2\sigma_{t-1}^2 \end{aligned} \tag{14}$$

Thus, we can get $q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_T) = \mathcal{N}(a_t\mathbf{z}_{t-1} + b_t\mathbf{z}_T, \delta_t^2\mathbf{I})$, which is also a Gaussian distribution.
Putting all these terms back to Equation (5), we'll have:

$$\begin{aligned} &q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0, \mathbf{z}_T) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, \mathbf{z}_0, \mathbf{z}_T), \tilde{\sigma}_t^2\mathbf{I}) \\ =&\frac{q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_T)q(\mathbf{z}_{t-1}|\mathbf{z}_0, \mathbf{z}_T)}{q(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_T)} \\ \propto&\exp\left[-\frac{1}{2}\left(\frac{(\mathbf{z}_t - a_t\mathbf{z}_{t-1} - b_t\mathbf{z}_T)^2}{\delta_t^2} + \frac{(\mathbf{z}_{t-1} - m_{t-1}\mathbf{z}_0 - n_{t-1}\mathbf{z}_T)^2}{\sigma_{t-1}^2} - \frac{(\mathbf{z}_t - m_t\mathbf{z}_0 - n_t\mathbf{z}_T)^2}{\sigma_t^2}\right)\right] \\ =&\exp\left[-\frac{1}{2}\left(\left(\frac{a_t^2}{\delta_t^2} + \frac{1}{\sigma_{t-1}^2}\right)\mathbf{z}_{t-1}^2 - 2\left(\frac{a_t\mathbf{z}_t - a_t b_t\mathbf{z}_T}{\delta_t^2} + \frac{m_{t-1}\mathbf{z}_0 + n_{t-1}\mathbf{z}_T}{\sigma_{t-1}^2}\right)\mathbf{z}_{t-1} + C(\mathbf{z}_t, \mathbf{z}_0, \mathbf{z}_T)\right)\right]. \end{aligned} \tag{15}$$

where $C(\mathbf{z}_t, \mathbf{z}_0, \mathbf{z}_T)$ indicates the constant term that is independent of $\mathbf{z}_{t-1}$. Since the Gaussian distribution probability density $\mathbf{z}_{t-1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\sigma}_t^2\mathbf{I})$ can be expressed in the following form:

$$\exp\left(-\frac{(\mathbf{z}_{t-1} - \tilde{\boldsymbol{\mu}}_t)^2}{2\tilde{\sigma}_t^2}\right) = \exp\left[-\frac{1}{2}\left(\frac{1}{\tilde{\sigma}_t^2}\mathbf{z}_{t-1}^2 - \frac{2\tilde{\boldsymbol{\mu}}_t}{\tilde{\sigma}_t^2}\mathbf{z}_{t-1} + \frac{\tilde{\boldsymbol{\mu}}_t^2}{\tilde{\sigma}_t^2}\right)\right]. \tag{16}$$

We can thus derive:

$$\begin{aligned}
\frac{1}{\tilde{\sigma}_t^2} &= \frac{a_t^2}{\delta_t^2} + \frac{1}{\sigma_{t-1}^2} \\
\Rightarrow \tilde{\sigma}_t^2 &= \frac{\sigma_{t-1}^2(\sigma_t^2 - a_t^2\sigma_{t-1}^2)}{\sigma_t^2} = \frac{\sigma_{t-1}^2}{\sigma_t^2}\delta_t^2 \qquad (17) \\
\frac{\tilde{\boldsymbol{\mu}}_t}{\tilde{\sigma}_t^2} &= \frac{a_t\mathbf{z}_t - a_t b_t \mathbf{z}_T}{\delta_t^2} + \frac{m_{t-1}\mathbf{z}_0 + n_{t-1}\mathbf{z}_T}{\sigma_{t-1}^2} \\
\Rightarrow \tilde{\boldsymbol{\mu}}_t &= \left[\frac{\sigma_{t-1}^2\big(a_t\mathbf{z}_t - a_t b_t\mathbf{z}_T\big) + \delta_t^2\big(m_{t-1}\mathbf{z}_0 + n_{t-1}\mathbf{z}_T\big)}{\delta_t^2\sigma_{t-1}^2}\right]\frac{\sigma_{t-1}^2}{\sigma_t^2}\delta_t^2 \\
&= \frac{\sigma_{t-1}^2}{\sigma_t^2}a_t\mathbf{z}_t + \frac{\delta_t^2}{\sigma_t^2}m_{t-1}\mathbf{z}_0 + \left(\frac{\delta_t^2}{\sigma_t^2}n_{t-1} - \frac{\sigma_{t-1}^2}{\sigma_t^2}a_t b_t\right)\mathbf{z}_T \qquad (18)
\end{aligned}$$

whose results are equivalent to Equation (6) and Equation (7).

Accordingly, maximizing the Evidence Lower Bound (ELBO) is equivalent to minimizing the KL divergence between ground truth and predicted posterior:

$$\begin{aligned}
&KL\left(q(\mathbf{z}_{t-1}|\mathbf{z}_0, \mathbf{z}_t, \mathbf{z}_T)\,||\,p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, \mathbf{z}_T)\right) \\
=&\mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{z}_0,\mathbf{z}_t,\mathbf{z}_T)}\left[\log\frac{\frac{1}{\sqrt{2\pi}\tilde{\sigma}_{t-1}}e^{-\frac{(\mathbf{z}_{t-1}-\tilde{\boldsymbol{\mu}}_{t-1})^2}{2\tilde{\sigma}_{t-1}^2}}}{\frac{1}{\sqrt{2\pi}\tilde{\sigma}_{\theta,t-1}}e^{-\frac{(\mathbf{z}_{t-1}-\tilde{\boldsymbol{\mu}}_{\theta,t-1})^2}{2\tilde{\sigma}_{\theta,t-1}^2}}}\right] \qquad (19) \\
=&\mathbb{E}_{q(\mathbf{z}_{t-1}|\mathbf{z}_0,\mathbf{z}_t,\mathbf{z}_T)}\left[\log\tilde{\sigma}_{\theta,t-1} - \log\tilde{\sigma}_{t-1} - \frac{(x_{t-1} - \tilde{\boldsymbol{\mu}}_{t-1})^2}{2\tilde{\sigma}_{t-1}^2} + \frac{(x_{t-1} - \tilde{\boldsymbol{\mu}}_{\theta,t-1})^2}{2\tilde{\sigma}_{\theta,t-1}^2}\right] \\
=&\log\frac{\tilde{\sigma}_{\theta,t-1}}{\tilde{\sigma}_{t-1}} - \frac{1}{2} + \frac{\tilde{\sigma}_{t-1}^2}{2\tilde{\sigma}_{\theta,t-1}^2} + \frac{(\tilde{\boldsymbol{\mu}}_{t-1} - \tilde{\boldsymbol{\mu}}_{\theta,t-1})^2}{2\tilde{\sigma}_{\theta,t-1}^2}
\end{aligned}$$

If we assume that only the mean term is learnable during training, and $\tilde{\sigma}_{\theta,t-1} = \tilde{\sigma}_{t-1}$, we can ignore all the unlearnable constants and arrive at the final training objective:

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{t,\mathbf{z}_0,\mathbf{z}_t,\mathbf{z}_T}\left[\frac{1}{2\tilde{\sigma}_t^2}\left\|\tilde{\boldsymbol{\mu}}_{t-1} - \tilde{\boldsymbol{\mu}}_{\theta,t-1}(\mathbf{z}_t, \mathbf{z}_T, t)\right\|^2\right] \qquad (20) \\
&= \mathbb{E}_{t,\mathbf{z}_0,\mathbf{z}_t,\mathbf{z}_T}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{z}_T, t)\right\|^2\right]
\end{aligned}$$

which is the same as in Equation (8). This concludes our derivation.

B. Training and Inference Pipeline

B.1. Pipeline Pseudo-code

B.2. Accelerate Inference

Similar to DDIM [5] and DBIM [7], the inference process of DPBridge can be accelerated using non-Markovian sampling scheme that preserves the marginal distributions of intermediate states. Specifically, we perform sampling over a reduced set of timesteps $\{t_i\}_{i=1}^N$, where $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$, allowing the number of inference steps N to be decoupled from the total number of training steps T. The accelerated inference process is given by:

$$\begin{aligned}
\mathbf{z}_{t_n} &= m_{t_n}\hat{\mathbf{z}}_0 + n_{t_n}\mathbf{z}_T + \\
&\sqrt{\sigma_{t_n}^2 - g_n^2}\frac{\mathbf{z}_{t_{n+1}} - m_{t_{n+1}}\hat{\mathbf{z}}_0 - n_{t_{n+1}}\mathbf{z}_T}{\sigma_{t_{n+1}}} + g_n\boldsymbol{\epsilon} \qquad (21)
\end{aligned}$$

where g_n denotes variance parameter that governs the stochasticity of the reverse sampling process. Notably, when $g_n = \tilde{\sigma}_t^2$, as defined in Equation (5), the process becomes a Markovian bridge.

Algorithm 1 Training

```
Require: Encoder $\mathcal{E}$; Decoder $\mathcal{D}$; Loss weights $\omega_1, \omega_2$
repeat
    Retrieve paired data $(\mathbf{x}, \mathbf{y})$
    $\mathbf{z}_0 = \mathcal{E}(\mathbf{y})$, $\mathbf{z}_T = \mathcal{E}(\mathbf{x})$
    $t \sim$ Uniform$(\{1, \dots, T\})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{z}_t = m_t \mathbf{z}_0 + n_t \mathbf{z}_T + \sigma_t \boldsymbol{\epsilon}$
    $\mathbf{z}'_t = \frac{1}{\sqrt{n_t^2 + \sigma_t^2}} (\mathbf{z}_t - n_t \mathbf{z}_T)$
    $\mathcal{L}_{elbo} = \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}'_t, \mathbf{z}_T, t)\|^2$ ,
    $\hat{\mathbf{z}}_0 = \frac{1}{m_t} (\mathbf{z}_t - n_t \mathbf{z}_T - \sigma_t \boldsymbol{\epsilon}_\theta(\mathbf{z}'_t, \mathbf{z}_T, t))$
    $\mathcal{L}_{ic} = \|\mathcal{D}(\hat{\mathbf{z}}_0) - \mathbf{y}\|^2$
    Take gradient step on $\nabla_\theta \left(\omega_1 \mathcal{L}_{elbo} + \omega_2 \mathcal{L}_{ic}\right)$
until Converged
```

Algorithm 2 Inference

```
Require: Encoder $\mathcal{E}$; Decoder $\mathcal{D}$; Trained $\boldsymbol{\epsilon}_\theta(\cdot, \cdot, \cdot)$
$\mathbf{z}_T = \mathcal{E}(\mathbf{x})$
for $t = T, \dots, 1$ do
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\boldsymbol{\epsilon} = \mathbf{0}$
    $\mathbf{z}'_t = \frac{1}{\sqrt{n_t^2 + \sigma_t^2}} (\mathbf{z}_t - n_t \mathbf{z}_T)$
    $\hat{\mathbf{z}}_0 = \frac{1}{m_t} (\mathbf{z}_t - n_t \mathbf{z}_T - \sigma_t \boldsymbol{\epsilon}_\theta(\mathbf{z}'_t, \mathbf{z}_T, t))$
    $\mathbf{z}_{t-1} = k_1 \mathbf{z}_t + k_2 \hat{\mathbf{z}}_0 + k_3 \mathbf{z}_T + \tilde{\sigma}_t \boldsymbol{\epsilon}$
end for
$\mathbf{y} = \mathcal{D}(\hat{\mathbf{z}}_0)$
return $\mathbf{y}$
```

C. Network Architecture

The basic structure of our network follows the original design of Stable Diffusion. To better utilize visual information ($\mathbf{z}_T$ in our case), we concatenating both intermediate samples and input image along the feature dimension and send in the noise prediction network as a whole. To address the dimension incompatibility with pretrained backbone, the input channels of the network are thus doubled to accommodate the concatenated input. In addition, as suggested by [3], we duplicate the weight tensor of these accommodated layers and divide the values by two to preserve the input magnitude to subsequent layers. As DPBridge is an image-conditioned generative framework, we employ null-text embedding for all cross-attention layers to eliminate the impact of text prompts.

D. Implementation Details

During training, the number of timesteps of is set to be 1000, and we use 50 sampling steps during inference with considerations of both quality and efficiency. Training our method takes 30K iterations using the Adam optimizer with a base learning rate of 3×10^{-5} and an exponential decay strategy after 100-steps warm-up. The batch size is 2, but with 8 steps gradient accumulation, the effective batch size is equivalent to 16. We also apply random horizontal flipping augmentation during training. The whole process takes approximately 3 days to complete on a single Nvidia RTX 3090Ti GPU.

E. Experiments

E.1. Additional Quantitative Results

In this section, we further evaluate the robustness of DPBridge system by perturbing the input image with different types and different levels of intensity of noise. In this experiment setting, we use Stable Diffusion 1.5 as pretrained backbone. The evaluation is conducted under both depth estimation and surface normal prediction tasks. We use NYUv2 and KITTI to report the depth estimation results, and for surface normal prediction, we report the results on NYUv2 and iBims-1. Tab. 4 reveals that the model's performance degrades as noise intensity increases, but the model performance remains acceptable under moderate-level-intensity noise. Among noise types, uniform noise shows the mildest effect on degradation, while salt & pepper and high-intensity Gaussian noise cause the most significant performance decline. In addition, the noise intensity has more impact on surface normal prediction than depth estimation, showcasing the inherent characteristics between different tasks. These results emphasize the model's robustness under moderate noise but still expose its vulnerability to extreme or specific types of noise.

Noise Type	**Noise Level**	**Depth Estimation**				**Surface Normal Prediction**			
		NYUv2		**KITTI**		**NYUv2**		**iBims-1**	
		AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	Mean↓	11.25° ↑	Mean↓	11.25° ↑
Original Version	-	6.9	95.7	11.2	87.3	18.1	54.2	24.3	51.2
Gaussian	$\mathcal{N}(0, 0.05)$	7.2	94.4	11.8	86.0	18.6	52.2	25.6	45.1
	$\mathcal{N}(0, 0.1)$	8.4	92.7	12.1	85.3	20.3	49.4	27.4	40.4
	$\mathcal{N}(0, 0.2)$	10.3	90.0	15.7	78.6	24.6	43.7	37.0	25.4
	$\mathcal{N}(0, 0.5)$	15.9	77.2	22.5	67.1	41.3	22.5	50.0	13.2
Uniform	$\mathcal{U}(-0.05, 0.05)$	7.0	95.0	11.4	86.5	18.3	53.6	24.5	50.7
	$\mathcal{U}(-0.1, 0.1)$	7.2	94.7	11.5	85.4	18.7	53.0	24.8	49.4
	$\mathcal{U}(-0.2, 0.2)$	7.6	94.0	12.0	85.0	19.5	50.5	26.0	46.1
	$\mathcal{U}(-0.5, 0.5)$	9.3	91.1	16.3	76.1	22.4	42.2	30.2	31.6
Poisson	*Poisson*$(\lambda = 0.05)$	10.2	90.2	13.1	82.5	21.3	45.6	28.1	38.9
	Poisson$(\lambda = 0.1)$	11.8	88.9	15.4	79.3	23.2	40.3	34.9	24.8
Salt & Pepper	*Probability*=5%	12.0	85.4	13.3	81.7	26.3	28.8	31.8	29.6
	Probability=10%	14.3	82.2	16.0	76.2	28.6	26.3	35.3	21.4

Table 4. Robustness evaluation on depth estimation and surface normal prediction tasks with different types and different intensities of noise. During all experiments, the input images are normalized between -1 to 1.

E.2. Additional Qualitative Results

To further evaluate the effectiveness and generalization capability of DPBridge, we extend the framework to additional dense prediction tasks, including semantic segmentation, optical flow estimation, edge detection, and style transfer. Qualitative results are shown in Figure 6, Figure 8, Figure 7, and Figure 9. These extensions are implemented by framing each task as a continuous-valued image-to-image translation problem and replacing the paired ground truth data accordingly. For semantic segmentation, we convert discrete labels into color maps and train the bridge model to predict colors in RGB space rather than class indices. For optical flow, we define the bridge process between the flow image and the first frame of the input RGB pair. To accommodate this, we modify the UNet architecture to condition on both RGB frames during inference and represent flow as a color-encoded image, enabling prediction in RGB space rather than raw uv coordinates. Across all tasks, DPBridge produces visually compelling results, demonstrating its versatility as a unified framework that can be easily adapted to a range of dense prediction scenarios. We note that these results are qualitative; quantitative evaluation and task-specific architectural refinements are left for future work.

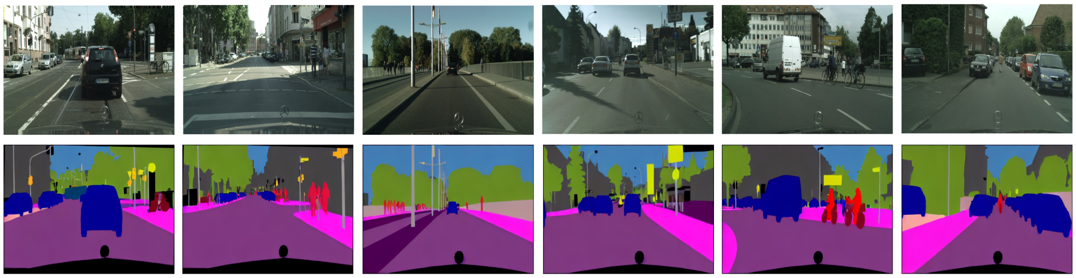

Figure 6. Qualitative examples of applying DPBridge to semantic segmentation task on CityScape [2] dataset. We transform discrete labels into color maps and us our bridge model to predict colors in RGB space instead of categorical labels.

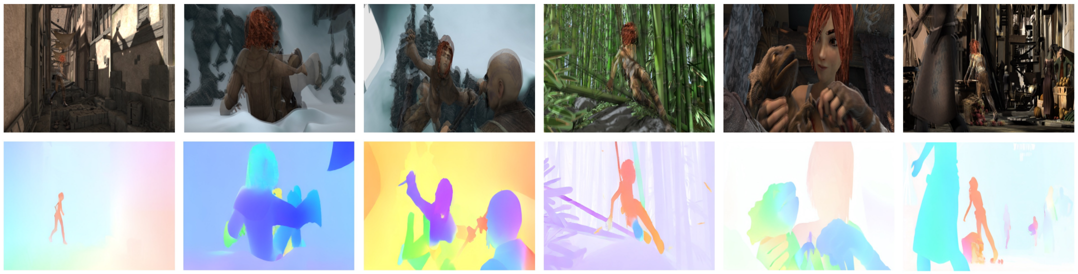

Figure 7. Qualitative examples of applying DPBridge to optical flow prediction task on MPI-Sintel [1] dataset. The input image is the overlayed RGB pair, and we transform the flow map into RGB space so that the diffusion bridge process can be constructed accordingly.

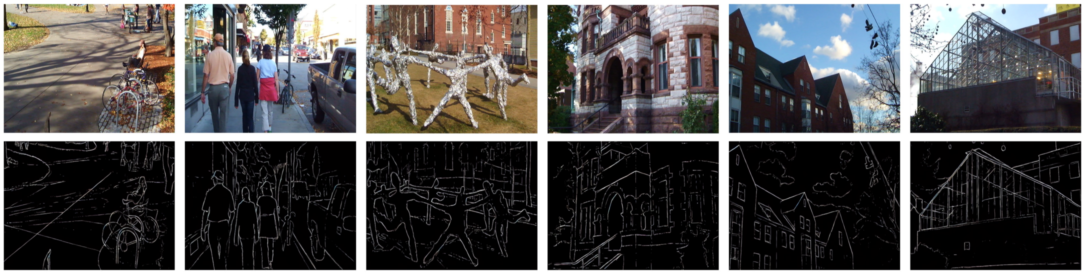

Figure 8. Qualitative examples of applying DPBridge to edge detection task on MultiCue [4] dataset.

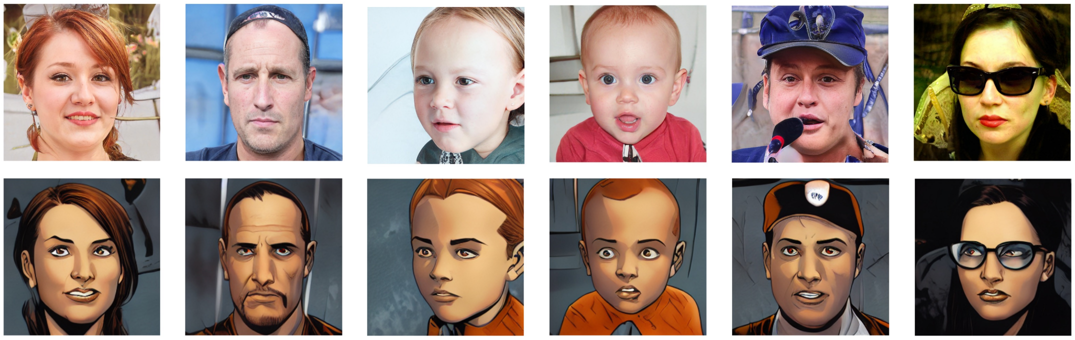

Figure 9. Qualitative examples of applying DPBridge to style transfer task on Face-to-Comics [6] dataset.

References

[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 5

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5

[3] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 3

[4] David A Mély, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. A systematic comparison between visual cues for boundary detection. *Vision research*, 120:93–107, 2016. 5

[5] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[6] Sxela. https://github.com/sxela/face2comics. `https://github.com/Sxela/face2comics`, 2021. 5

[7] Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024. 2