Vector Quantized Image-to-Image Translation *Supplementary Materials*

Yu-Jie Chen$^{\star 1,2}$, Shin-I Cheng$^{\star 1,2}$, Wei-Chen Chiu^{1,2},
Hung-Yu Tseng^{3}, and Hsin-Ying Lee^{4}

[1]National Chiao Tung University, Taiwan [2]MediaTek-NCTU Research Center [3]Meta
[4]Snap Inc.

1 Dataset Details

We consider three unpaired datasets: AFHQ dataset [1], Yosemite dataset [12] and Portrait dataset [8], and one paired dataset Cityscapes [2]. AFHQ dataset has three domains: 5153 training and 500 testing images of “cat”, 4739 training and 500 testing images of “dog”, and 4738 training and 500 testing images of “wildlife” (e.g. tiger, lion, wolf, etc). Yosemite dataset contains landscape photos collected from Yosemite National Park with two classes, which are related to photos of two seasons: 1231 training and 309 testing images of “summer” and 962 training and 238 testing images of “winter”. Portrait dataset has two classes (i.e., portraits in photography images and the ones in painting images), It contains 1711 training and 100 testing images of painting portraits, and 6352 training and 100 testing images of photography portraits. Cityscapes dataset is pairwise, having 2975 training and 500 testing images of the cityscape and their corresponding semantic segmentation maps. During the training phase, we resize all images to the resolution of 256×256.

2 Implementation Details

We implement the models with Pytorch. The training of our whole proposed framework is divided into two stages. Firstly, we train our VQ-I2I architecture by using the objective function summarized in Equation 8 of our main manuscript to address the diverse image-to-image translation task and build a representative vector-quantized codebook for the content information, where the details are provided later in Section 2.1. Then, in the second stage, we go on training an autoregresssive transformer model based on the content codebook and the content encoder E^c learnt in the first stage to further address two applications: unconditional image generation and image extension. The details of the second stage are provided later in Section 2.2 and 2.3. Moreover, we provide the training details of the unimodal VQ-I2I (denoted as uni-VQ-I2I) baseline in Section 2.4.

2.1 VQ-I2I

Settings of hyper-parameters. We use the Adam optimizer [3] for model training with a batch size of 1, a learning rate of 0.00001, and exponential decay rates

$(\beta_1, \beta_2) = (0.5, 0.999)$. In all experiments, we set the hyper-parameters λ to balance between different objective functions as follows: $\lambda_{\text{adv}} = 0.1$, $\lambda_1^{\text{recon}} = 5$, $\lambda_{\text{vq}} = 1$, $\lambda_1^{\text{content}} = 0.2$ and $\lambda_1^{\text{style}} = 1$. For the vector-quantized content codebook, different parameters are set up for each of the datasets used in our experiments, as we assume that the dataset with more multifarious content needs the larger codebook size. Yosemite dataset contains rich and complex landscapes, so we adjust its content codebook with having the number of embedding set to 512 and the embedding dimensionality set to 512. For both AFHQ and Photo2Portrait datasets, the number of embeddings is set to 256 and the embedding dimensionality is set to 256. For Cityscapes dataset, we set the number of embeddings to 64 and the embedding dimensionality to 256 for the content codebook. The settings of the content codebook for various datasets are summarized in Table 1.

Table 1. The settings of content codebook for different datasets used in our experiments, including the codebook size (i.e. the number of codes in a codebook) and the dimensionality of each code/embedding.

Datasets	codebook size	code dimensionality
Yosemite	512	512
AFHQ	256	256
Portrait	256	256
Cityscapes	64	256

Network architecture. The shared content encoder E^c, generators $\{G_X, G_Y\}$ and discriminators $\{D_X, D_Y\}$ in our model mostly follow the corresponding architectures proposed in VQGAN [3] but with two modifications: (1) We additionally concatenate four residual blocks with AdaIN layers (as what proposed in MUNIT [5]) to the front of generators; (2) We replace the original normalization layers in discriminators with Instance normalization, as we train the whole model with a batch size of 1. For the style encoders $\{S_X, S_Y\}$, they are identical to the one used in MUNIT [5].

Adversarial loss. The adversarial loss is applied on the translated images u and v (cf. Eq.4) with respect to the real images $x \in X$ and $y \in Y$ respectively, where the discriminators (inherited from VQGAN) is used to matching their distributions (u versus x; v versus y). Specifically, we implement the adversarial loss as follow:

$$\begin{aligned}
L_{\text{adv}} &= L_{D_X} + L_{D_Y}, \\
L_{D_X} &= -[\log D_X(x) + \log(1 - D_X(u))], \\
L_{D_Y} &= -[\log D_Y(y) + \log(1 - D_Y(v))].
\end{aligned}$$

2.2 Transformer

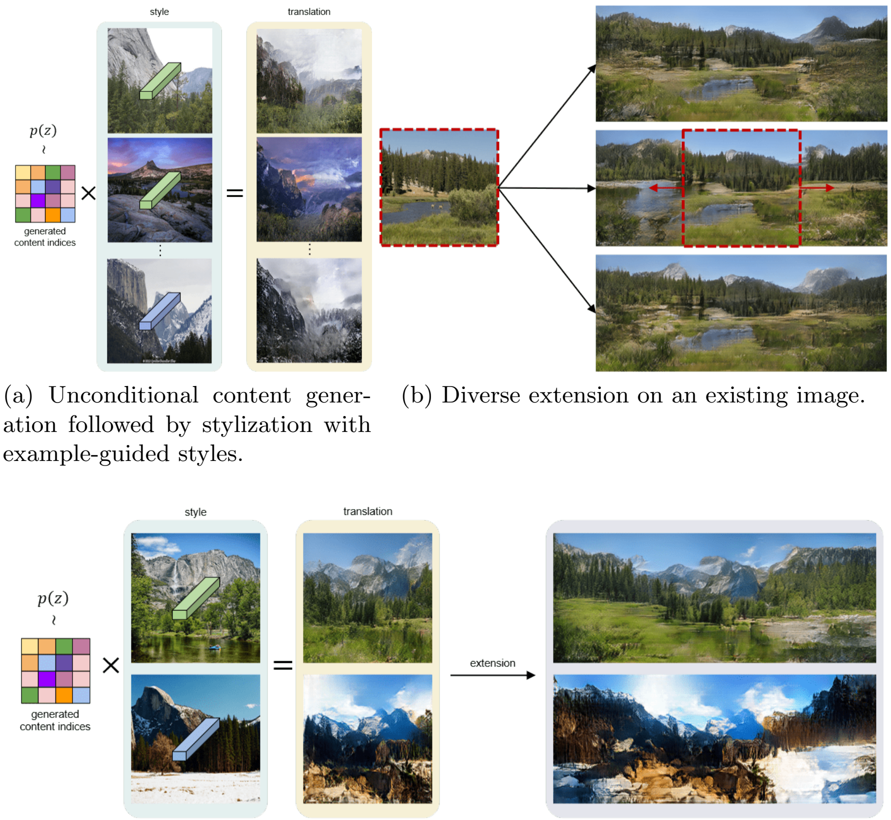
Setting of hyper-parameters and network architecture. For learning the autoregressive transformer model on the Yosemite dataset, we use the Adam optimizer with a batch size of 1, a learning rate of 0.00001, and exponential decay rates $(\beta_1, \beta_2) = (0.5, 0.999)$. The transformer we use is the same as the one used in VQGAN [3], which is identical to the GPT2 architecture [10]. When in the testing phase, we set the parameters of sampling as follows: temperature $t = 10$ and a top-k cutoff at $k = 2$.

Ordering difference between training and testing phases. In the training phase, we simply unfold the 2-dimensional quantized content representation of each training image on a row-major ordering (as the way VQGAN does) into a form of a discrete sequence and feed it into the transformer model for learning of content distribution. In other words, the transformer accesses the complete index sequence of an image at each time. In the testing phase, where we perform unconditional image generation and content extension, we design a square sliding window and feed only the indices in the current sliding window into the transformer (similarly on a row-major ordering as VQGAN). The transformer predicts a new content index only based on its previous indices within the sliding window, and the whole generation is done by moving the sliding window (in a left-to-right, top-to-bottom manner) and repeating this process. Here we use a sliding window with a size of 16×16.

2.3 Applications

In addition to the transitional stylization generation described in the main paper, we further demonstrate several applications that our proposed method is able to unleash, as described in the following and shown in Figure 1:

- **Unconditional content generation followed by stylization with example-guided styles.** We first generate a sequence of content indices unconditionally from the learned transformer model and then modulate it with different styles from various domains.
- **Diverse extension on an existing image.** We utilize content indices from an existing image as the condition for the transformer model and generate the indices for the extended image region. We are able to produce diverse results of extension through having multiple samples drawn from the conditional distribution predicted by our transformer model. Figure 1 (b) demonstrates the diversity on the results of such content extension.
- **Unconditional content generation with both translation/stylization and extension.** We combine both content generation and extension with translation/stylization to showcase the flexibility on applications enabled by our proposed method as well as the variability of styles. We perform this combination in three steps: generate content indices unconditionally, modulate content indices with different style vectors, and apply extension.



(a) Unconditional content generation followed by stylization with example-guided styles.

(b) Diverse extension on an existing image.

(c) Unconditional content generation with both translation/stylization and extension.

Fig. 1. Various applications with VQ-I2I. (a) With generated content indices produced by the learned transformer model, we can combine some style features to synthesize diverse images. (b) Given an existing image, we are able to extend the diverse content on both sides. (c) We can combine unconditional content generation, diverse stylization, and extension together as a new application, which is easily achieved by our proposed method.

2.4 Unimodal VQ-I2I baseline (i.e., uni-VQ-I2I)

As described in Section 4 of our main manuscript, we construct a uni-modal VQ-I2I variant as an additional baseline (denoted as uni-VQI2I), in which its latent space is not disentangled (i.e. there is no explicit separation between the content and style latent factors as our VQ-I2I). The architecture of uni-VQ-I2I is illustrated in Figure 2. Specifically, in such uni-VQ-I2I baseline, we assume that the domain-specific style information is implicitly modeled by the generators, thus the domain-specific style features are discarded.

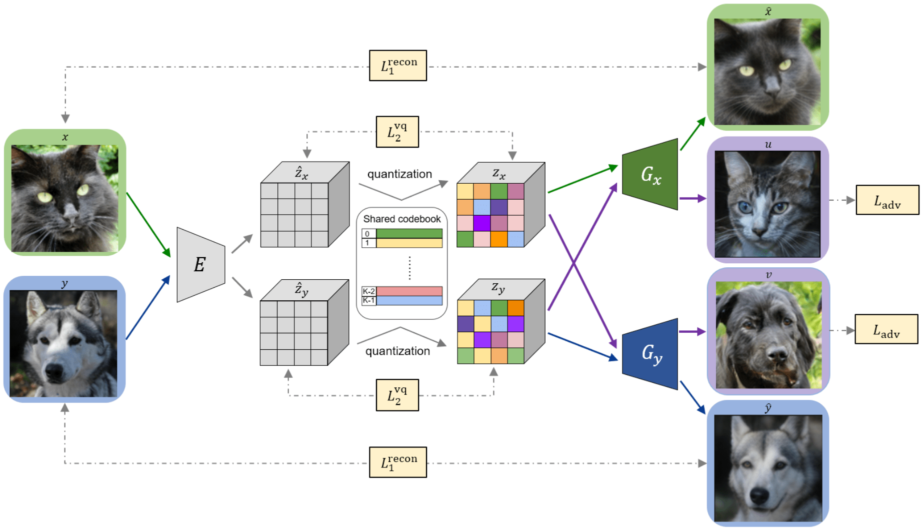


Fig. 2. Unimodal VQ-I2I. The architecture illustration for the uni-VQ-I2I baseline.

As uni-VQ-I2I does not disentangle the latent space into the content and style parts, there are three main differences of unimodal VQ-I2I from multimodal VQ-I2I (i.e. our full model):

- **Encoder.** uni-VQ-I2I only uses an public encoder E to learn the joint latent space across domains.
- **Generators.** We remove the AdaIN normalization layers [4,5] from G_X, G_Y, as there is only a single latent vector as input.
- **Loss function**. uni-VQ-I2I does not contain the style regression loss L_1^{style} and the content regression loss L_1^{content}, and we modify the full loss function to

$$\begin{aligned} L_D &= L_{\text{adv}}, \\ L_{E,Z,G} &= -\lambda_{\text{adv}} L_{\text{adv}} + \lambda_1^{\text{recon}} L_1^{\text{recon}} + \lambda_{\text{vq}} L_{\text{vq}}. \end{aligned} \tag{1}$$

Network architecture Similar to what has been described in Section 2.1, the encoder, generators and discriminators in the architecture of uni-VQ-I2I are mostly inherited from VQGAN [3].

Uni-VQ-I2I exploration. To verify whether the generators in uni-VQ-I2I are able to handle the translation without having the disentangled representations, we record the total number of embeddings/codes being used in two domains. Take the training set of AFHQ dataset as an example, Figure 3 reveals that the used embeddings/codes of cat and dog images are highly overlapped. This fact indicates that the model tends to learn a general latent representation for the codebook, rather than using specific codes for specific domains. That is, the

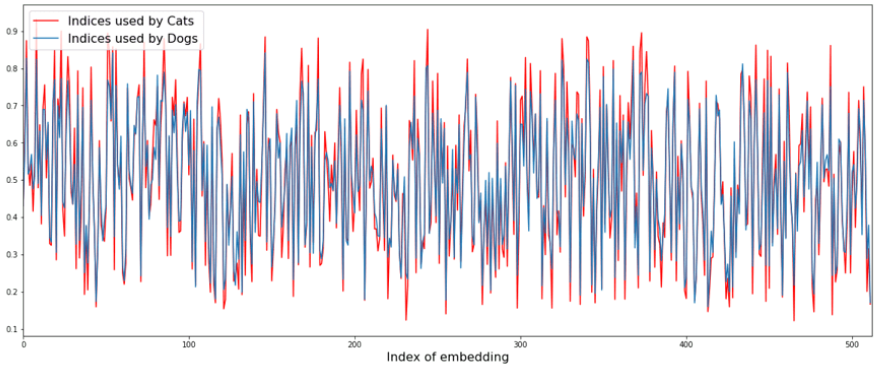


Fig. 3. Embeddings/codes used by training uni-VQ-I2I model for the translation between Cats and Dogs in AFHQ dataset. It is observable that the embeddings/codes used by these two domains are highly overlapped. Noting the numerical range on y-axis has been normalized with respect to total number of training images of each domain.

translation is implicitly handled by domain-specific generators. Besides, all the codes in the codebook (number of embeddings = 512) are used.

We provide additional qualitative results in Figure 7 and 8 (from AFHQ and Yosemite datasets respectively) to make comparison between VQ-I2I and uni-VQ-I2I, where we can observe that: although uni-VQ-I2I seems to be capable of addressing the I2I task, the content of the translation results is not consistent with the corresponding source images.

2.5 Details of compared baselines.

For the I2I baselines, we follow their official code and default settings on training and hyper-parameters (noting that for U-GAT-IT [7] we adopt its *light* version out of consideration for computational cost). As DRIT [8], MUNIT [5] and BicycleGAN [13] are multi-modal models, we sample one translation result for every input image for FID computation.

Regarding the baseline of image extension, Boundless [11], we adopt the implementation from `https://github.com/recong/Boundless-in-Pytorch`, in which we train the models without modifying any hyperparameters to perform horizontal extension for 50% and 75% toward the right-hand side as mentioned in the main paper.

3 Additional Experiments

3.1 Multimodal Translation on Cat→Wild.

In our main manuscript, we have presented the intra- and inter-domain multimodal translation. Here we also provide more experimental results on the AFHQ dataset for translating from cats to wildlife animals. The wildlife animals in AFHQ dataset contains various animals, such as lions, tigers and leopards. We present the inter-domain multimodal results in Figure 4. The generator is able to learn a general representation of various styles in the same domain.

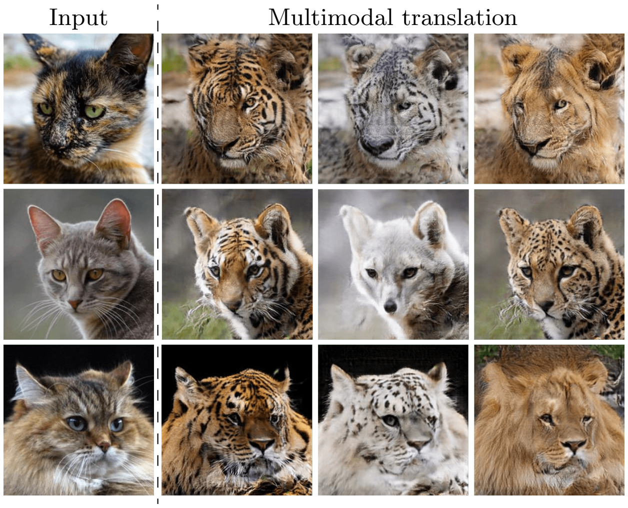


Fig. 4. Multimodal translation on cat→wild. We present the inter-domain translations on cat→wild. Our VQ-I2I model is able to generate different categories of wide animals in the target domain.

3.2 Having Different Number of Splits for Transitional Stylization

In Figure 5 we present the results of transitional stylization with using different numbers of splits (noting that we partition the content map horizontally into several equal splits and modulate different parts of the content map independently with different proportions by mixing the two styles, as described in Sec. 4.3 of the main manuscript). We observe that when the number of splits increase, the transitional stylization gets more smooth to gradually change from one style to another.

3.3 Baseline via sequential combination of SOTA methods.

We build a baseline based on a sequential combination of StyleGAN2 (generation), U-GAT-IT (translation), and Boundless (extension) to make a comparison between VQ-I2I and a sequential combination of SOTA mothods. With conducting an experiment of firstly generating 100 landscape images of size 256×256, translating them to summer styles, and finally extending the width for 128 pixels toward the right-hand side, our VQ-I2I is able to provide superior performance

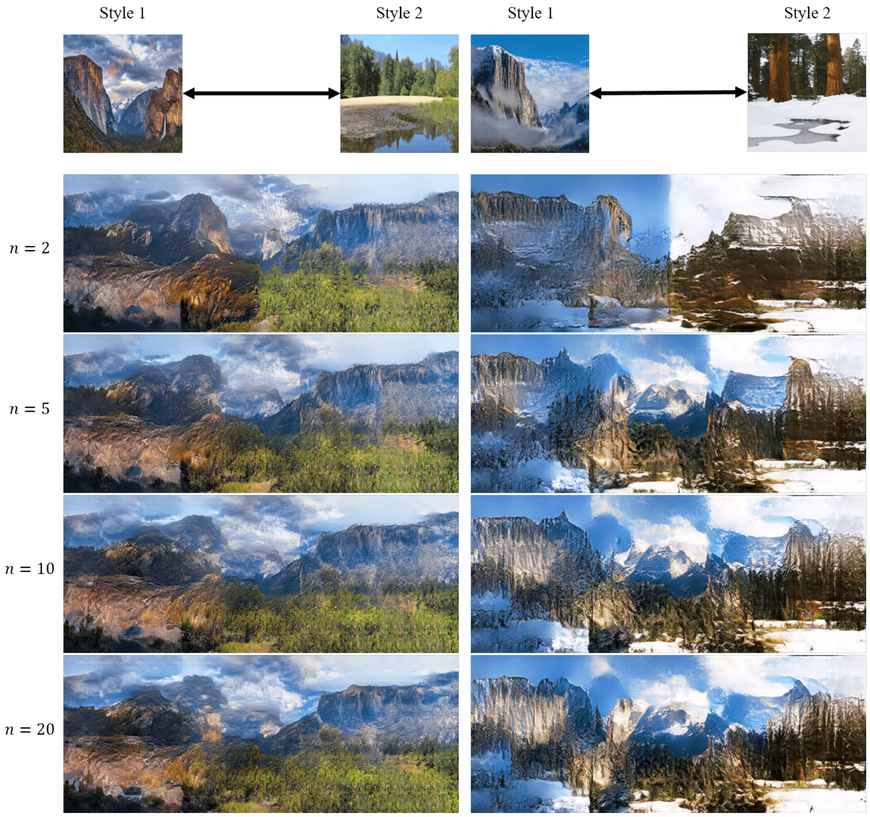


Fig. 5. Experiments to have different number of splits on the content map to perform transitional stylization. The first row shows the two referenced styles, and the second to the last rows are the transitional results when using 2, 5, 10 and 20 splits, respectively.

with FID 107.62 than such baseline with FID 128.73 (w.r.t. summer images from Yosemite dataset).

3.4 More comparisons with recent papers on unconditional generation and extension tasks.

We include more recent baselines on unconditional generation (i.e. StyleGAN2 [6]) and extension (i.e. InfinityGAN [9]). For unconditional generation, StyleGAN2, our VQ-I2I, and VQGAN achieve FID 106.35, 127.31, and 127.84 respectively; For image extension, following InfinityGAN's setting (i.e. given images of size 256×128 and extending them to 256×256), InfinityGAN, our VQ-I2I, and Boundless achieve FID 143.97, 109.86, and 101.68 respectively (Noting FID scroes above are all evaluated on 100 generated/extended images w.r.t. Yosemite dataset). Though the main focus of our VQ-I2I is to handle multiple tasks in a unified framework instead of targeting the state-of-the-art performance, it still provides comparable performance with the recent works on generation or extension.

4 Additional Results

4.1 Image-to-Image Translation

In Figure 6, we present additional results for dog→cat, winter→summer, and photo→portrait, obtained by various methods. Besides, we show the translation results of VQ-I2I and uni-VQ-I2I for both directions on shape-variant (AFHQ) and shape-invariant (Yosemite) datasets in Figure 7 and 8, respectively.

4.2 Applications

We demonstrate more application results. We show the combination of unconditional image generation, image translation, and image extension in Figure 9. In Figure 10, we show the extension on both summer and winter images in Yosemite dataset. Since the content indices are sampled from the content distribution, the transformer model is able to generate diverse extension results.

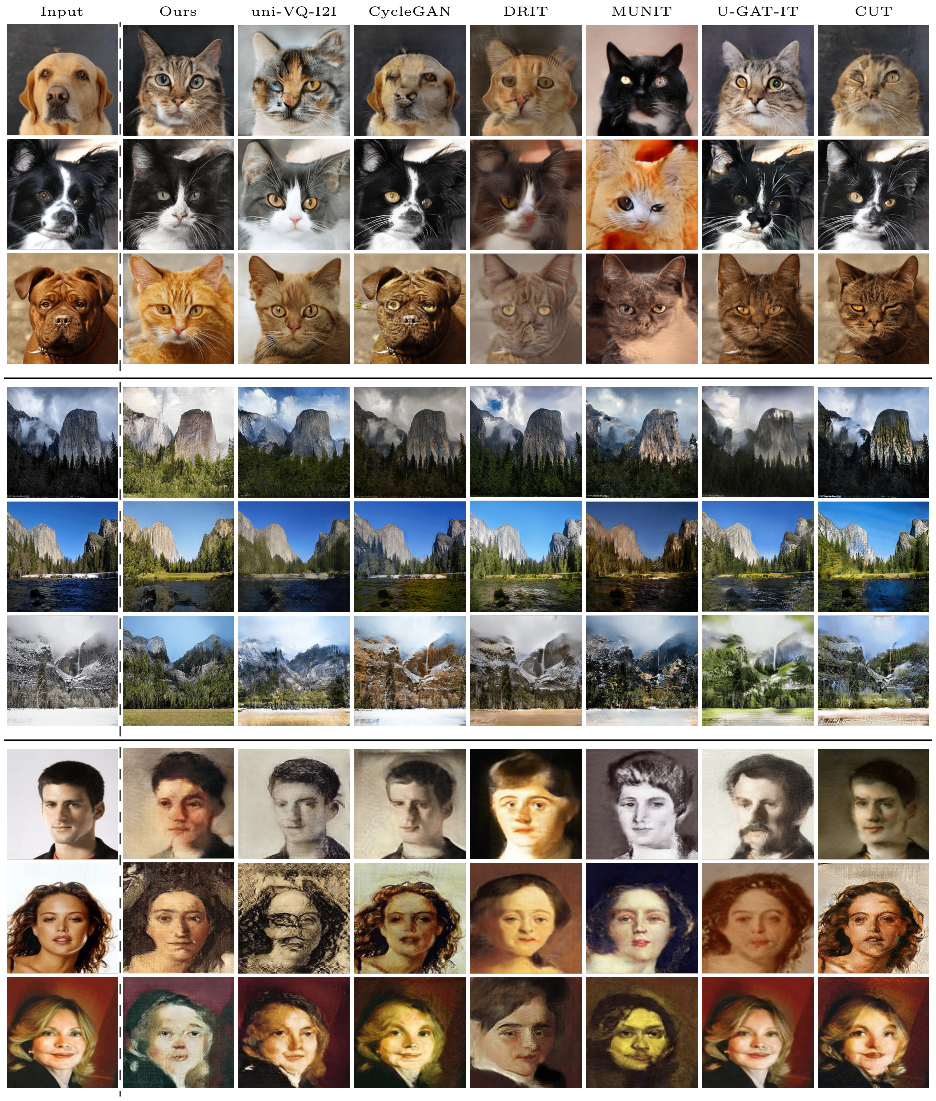


Fig. 6. More qualitative comparisons with conventional image-to-image translation methods. We provide qualitative examples of the translation results produced by various methods, trained on unpaired datasets. The left-most column shows the input images in the source domain. The other seven columns show the corresponding translated images in the target domain. Every three rows from top to bottom are: dog→cat, winter→summer, and photo→portrait.

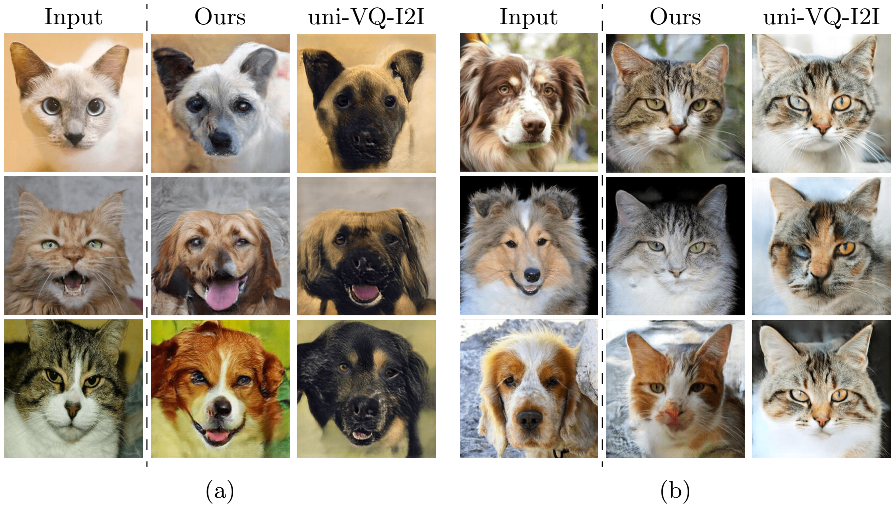


Fig. 7. Qualitative comparison between our VQ-I2I and uni-VQ-I2I baseline on AFHQ dataset. Three columns on the left show the translation cat→dog, while three columns on the right show the translation dog→cat.

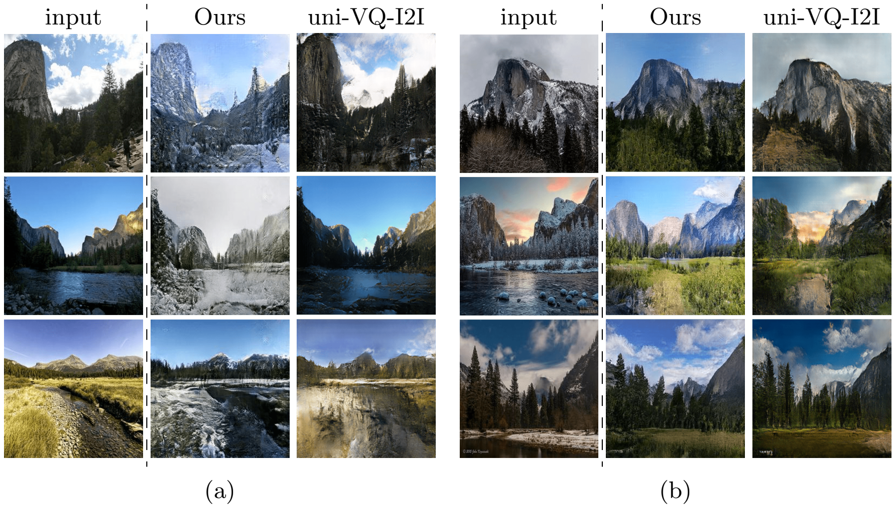


Fig. 8. Qualitative comparison between our VQ-I2I and uni-VQ-I2I baseline on Yosemite dataset. Three columns on the left show the translation summer→winter, while three columns on the right show the translation winter→summer.

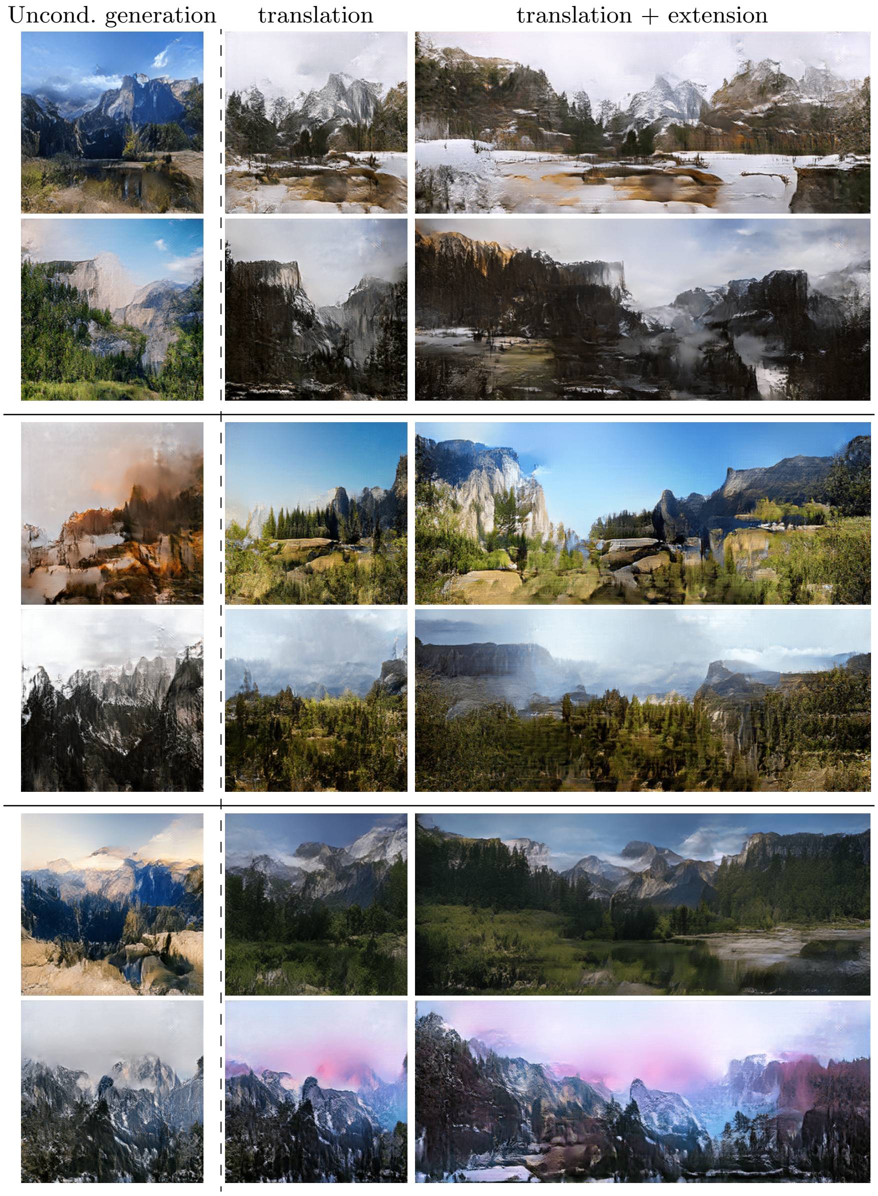


Fig. 9. Unconditional image generation combined with translation and extension. From top to bottom: summer→winter, winter→summer and intra-domain extension.

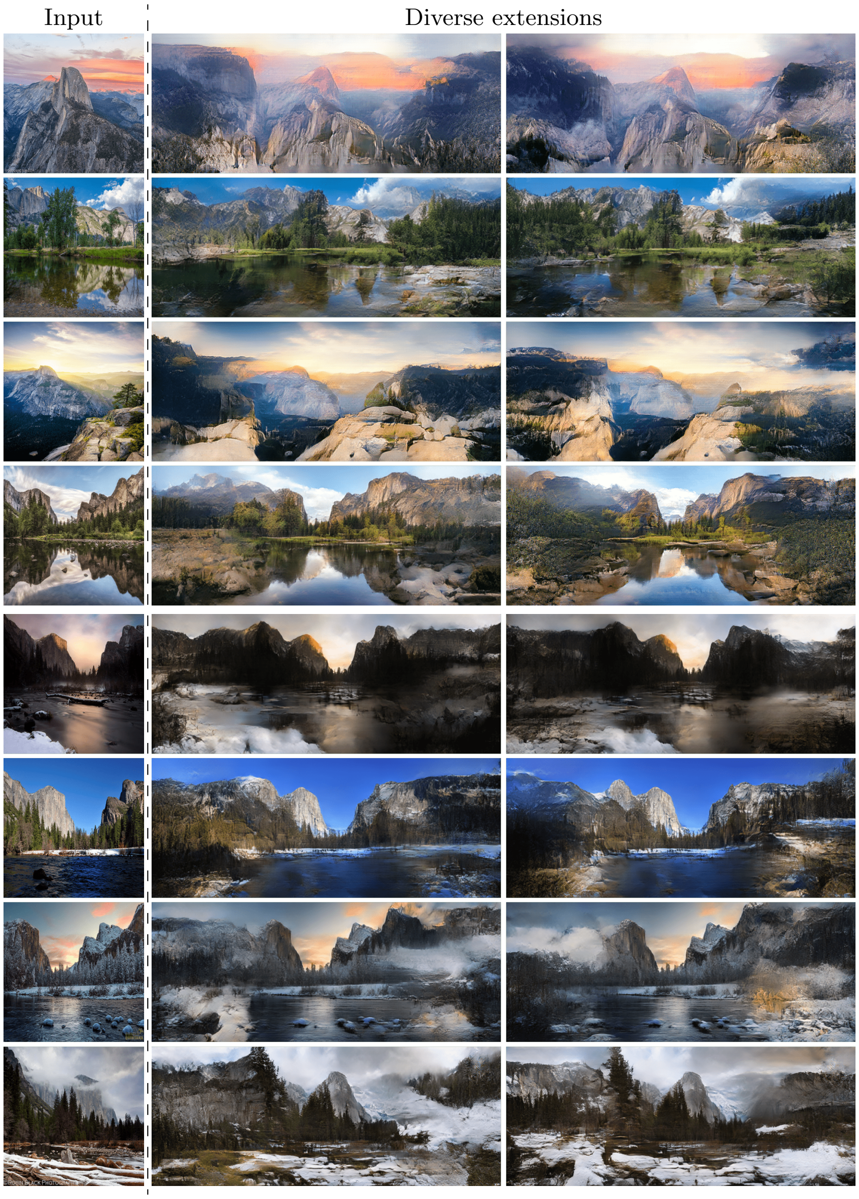


Fig. 10. Diverse image extension. The transformer is able to generate diverse extension results given an existing image.

References

1. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
3. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
5. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: European Conference on Computer Vision (ECCV) (2018)
6. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
7. Kim, J., Kim, M., Kang, H., Lee, K.: U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In: International Conference on Learning Representations (ICLR) (2020)
8. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: European Conference on Computer Vision (ECCV) (2018)
9. Lin, C.H., Lee, H.Y., Cheng, Y.C., Tulyakov, S., Yang, M.H.: Infinitygan: Towards infinite-pixel image synthesis. In: International Conference on Learning Representations (ICLR) (2021)
10. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog (2019)
11. Teterwak, P., Sarna, A., Krishnan, D., Maschinot, A., Belanger, D., Liu, C., Freeman, W.T.: Boundless: Generative adversarial networks for image extension. In: IEEE International Conference on Computer Vision (ICCV) (2019)
12. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (ICCV) (2017)
13. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)