Unsupervised Image-to-Image Translation with Generative Prior

Shuai Yang Liming Jiang Ziwei Liu Chen Change Loy
S-Lab, Nanyang Technological University
{shuai.yang, liming002, ziwei.liu, ccloy}@ntu.edu.sg

Supplementary Material

Contents

1. Implementation Details of GP-UNIT

1.1. Dataset and Model

SynImageNet-291. For synthesized data, we use the official BigGAN-deep-128 model on TF Hub [3] to generate correlated images associated by random latent codes for each of the 291 domains including dogs, wild animals, birds and vehicles. Their class indexes in the original ImageNet 1000 classes are 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 104, 105, 106, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 380, 382, 383, 384, 385, 386, 387, 388, 407, 436, 468, 511, 555, 586, 609, 627, 654, 656, 675, 717, 734, 751, 757, 779, 803, 817, 829, 847, 856, 864, 866, 867, 874. We apply truncation trick to the latent codes, and obtain 3K images with truncation threshold of 0.5 and 3K images with truncation threshold of 1.0. After filtering low-quality ones, we finally obtain 655 images per domain that are linked across all domains, 600 of which are for training. For Bird $\leftrightarrow$ Dog or Car, four classes of birds (class index: 10, 11, 12, 13), four classes of dogs (class index: 214, 218, 222, 232) and four classes of cars (class index: 436, 511, 627, 656) are used. An overview of images in the 291 domains in *synImageNet-291* is shown in Figs. 1-2

ImageNet-291. For each domain $\mathcal{X}$, we first calculate the mean style feature $S_{\mathcal{X}}$ of the images in $\mathcal{X}$ from *synImageNet-291*. The style feature is defined as the channel-wise mean of the conv5_2 feature of pre-trained VGG [17]. Then, we apply HTC [4] to ImageNet [15] to detect and crop the object regions in the domain $\mathcal{X}$. Small objects are filtered. The remaining images are ranked based on the similarity between their style features and $S_{\mathcal{X}}$. We finally select the top 650 images to eliminate outliers, with 600 images for training and 50 images for testing.

Other datasets. AFHQ [5] uses 4K training images and 500 testing images per domain. CelebA-HQ [9] uses 29K training images and 1K testing images. MS-COCO [13] uses 2K giraffe images for training and 197 images for testing. Yosemite [8] use 1,231 summer images and 962 winter images for training, and use 309 summer images and 238 winter images for testing. Metface [10] uses 1,336 images for training.

License of the Dataset. AFHQ [5] and CelebA-HQ [9] are under CC BY-NC 4.0 license. MS-COCO [13] is under CC BY 4.0 license. Metface [10] is under CC BY-NC 2.0 license. ImageNet [15] provides the terms of access at https://www.image-net.org/download. Head Pose Image Database [6] provides the terms of use "This database can be used for any purpose" (http://crowley-coutaz.fr/Head%20Pose%20Image%20Database.html). Images of TGaGa [18] are kindly provided by the authors. Yosemite are provided by https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix/blob/master/docs/datasets.md.

License of the Model. HTC [4] and BigGAN [3] are under Apache License 2.0. COCO-FUNIT [16] is under NVIDIA Source Code License for Imaginaire. MUNIT [8] is under CC BY-NC-SA 4.0 license. StarGAN2 [5] is under CC BY-NC 4.0 license. U-GAT-IT [11] is under MIT License. TraVeLGAN [2] is provided at https://github.com/KrishnaswamyLab/travelgan without claiming licenses. TraVeL is designed for 128×128 images and does not support multi-modal translation. In the experiment, we upsample its results to 256×256 to calculate FID.

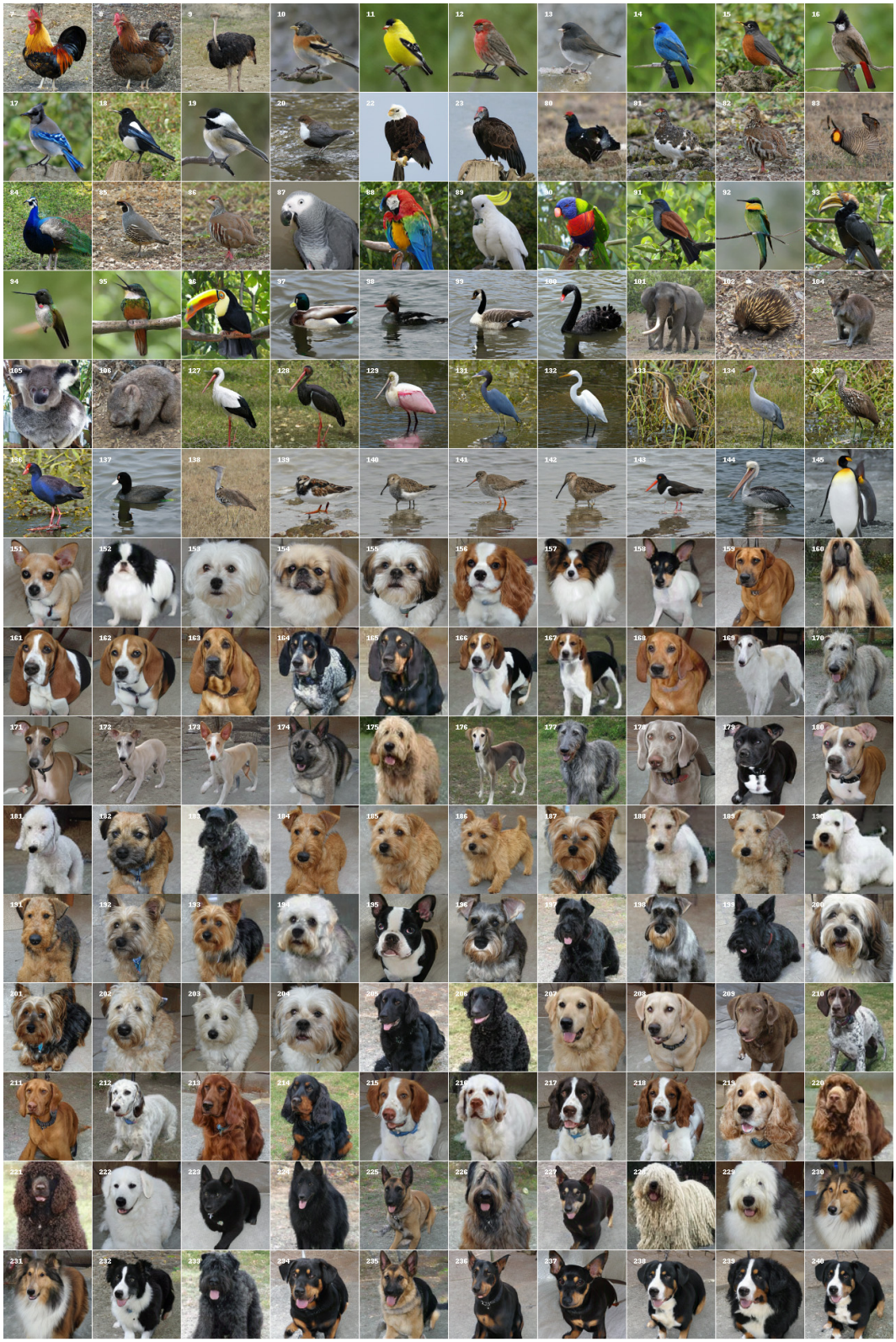


Figure 1. An domain overview of *synImageNet-291* (Part I).

Figure 2. An domain overview of *synImageNet-291* (Part II).

1.2. Network Architecture

Let "Cc(k)/s" denote a Convolution-Normalization-Activation layer, with $k \times k$ convolution kernels, output channel number c and stride s. If not specified, the default value of k and s are 3 and 1, respectively. Let "FCc" be a fully connection layer with output dimension c. "↑" denotes a $2\times$ nearest upsampling layer. Let "Res$c$" denote a Residual Block [7] with output channel number c and "AdaRes$c$" denote a conditional "Res$c$" with each convolution layer followed by an AdaptiveInstanceNorm layer.

Stage I. The content encoder E_c consists of: C64/1-C64/2-C64/1-C128/2-C128/1-C256/2-C256/1-C512/2-C512/1-C512/2-C512/1-C1/1, where "C" is a Convolution-InstanceNorm-LeakyReLU layer.

The style encoder E_s consists of: C64/2-C128/2-C256/2-C512/2-C512/2-C512/2, where "C" is a Convolution-LeakyReLU layer.

The decoder F consists of: C512-C512-C512-C512↑-C512↑-C256↑-C128↑-C64↑-C3, where "C" is a Convolution-AdaptiveInstanceNorm-ReLU layer except the input layer and output layer. The input layer "C512" is a Convolution-InstanceNorm-LeakyReLU layer and the output layer "C3" is a Convolution-Tanh layer. The decoder F_s architecture consists of: {C512-C512-C512-C512↑-C512}-C3. Layers in $\{\cdot\}$ are shared with F and are conditioned by domain labels, while the remaining layers in F are conditioned by the style features.

The classifier C consists of: C32(3)/2-C64(3)/1-C292(4)/1-FC292, where "C" is a Convolution-LeakyReLU layer.

Stage II. The architecture of E_c is the same as in Stage I.

The style encoder E_s consists of: C64(7)/1-C128(4)/2-C256(4)/2-C256(4)/2-C256(4)/2-C256(4)/2-GAvgPool-FC256-FC64-FC256, where "C' is a Convolution-ReLU laye. "GAvgPool" is a global averaging pooling layer. "FC" is a fully connection layer followed by a ReLU layer.

The generator G consists of: AdaRes512-AdaRes512↑-AdaRes512-DSC↑-AdaRes256-DSC↑-AdaRes128↑-AdaRes64↑-AdaRes64-C3(7), where "C" is a Convolution-Tanh layer and "DSC" is the proposed dynamic skip connection.

The discriminator D consists of: C64(3)-Res128↓-Res256↓-Res512↓-Res512↓-Res512↓-C512(4)-C512(1), where "C" is a Convolution-InstanceNorm-LeakyReLU layer. "↓" is a 2D average pooling layer with kernel size 2×2 within the Residual Block for downsampling.

For dynamic skip connection, in the main paper, the activation σ in Eq. (8) is a Sigmoid layer. In Eq. (9), σ is a ReLU-Tanh layer to make the mask more sparse. We do not use any activation in Eq. (11) in our implementation.

1.3. Network Training

Stage I. We adopt the Adam optimizer with a fixed learning rate of 0.0002. Each iteration uses 16 image pairs from *SynImageNet-291* and 16 images from *ImageNet-291*+CelebA-HQ. We use one NVIDIA Tesla V100 GPU to train our network for 4 epoches (about 44K iterations), which takes about 11 hours.

Stage II. We adopt the Adam optimizer with a fixed learning rate of 0.0001. The batch size is set to 16. We use one NVIDIA Tesla V100 GPU to train our network for 75K iterations, which takes about 46 hours. To compute the style loss, f_D uses the features of the 5th Resblock for Cat $\leftrightarrow$ Human Face, and the 4th Resblock for all other tasks.

2. Supplementary Experimental Results of GP-UNIT

2.1. Comparison with State-of-the-Art Methods

2.1.1 Visual comparison and multi-modal translation

In addition to the examples shown in the main paper, we show more visual comparison results with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5] in Figs. 3-12. Our method surpasses these methods in:

- more accurate content correspondences with the input images;
- less artifacts caused by the domain-specific information leakage form the input images;
- better matched shape and appearance features with the target domain;
- more realistic image details.

Figure 3. Visual comparison on Male→Female with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

We further show multi-modal translation outputs generated by GP-UNIT from four random style features, demonstrating that our method strikes a good balance between image quality and intra-domain diversity.

Figure 4. Visual comparison on Female → Male with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

Figure 5. Visual comparison on Dog → Cat with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

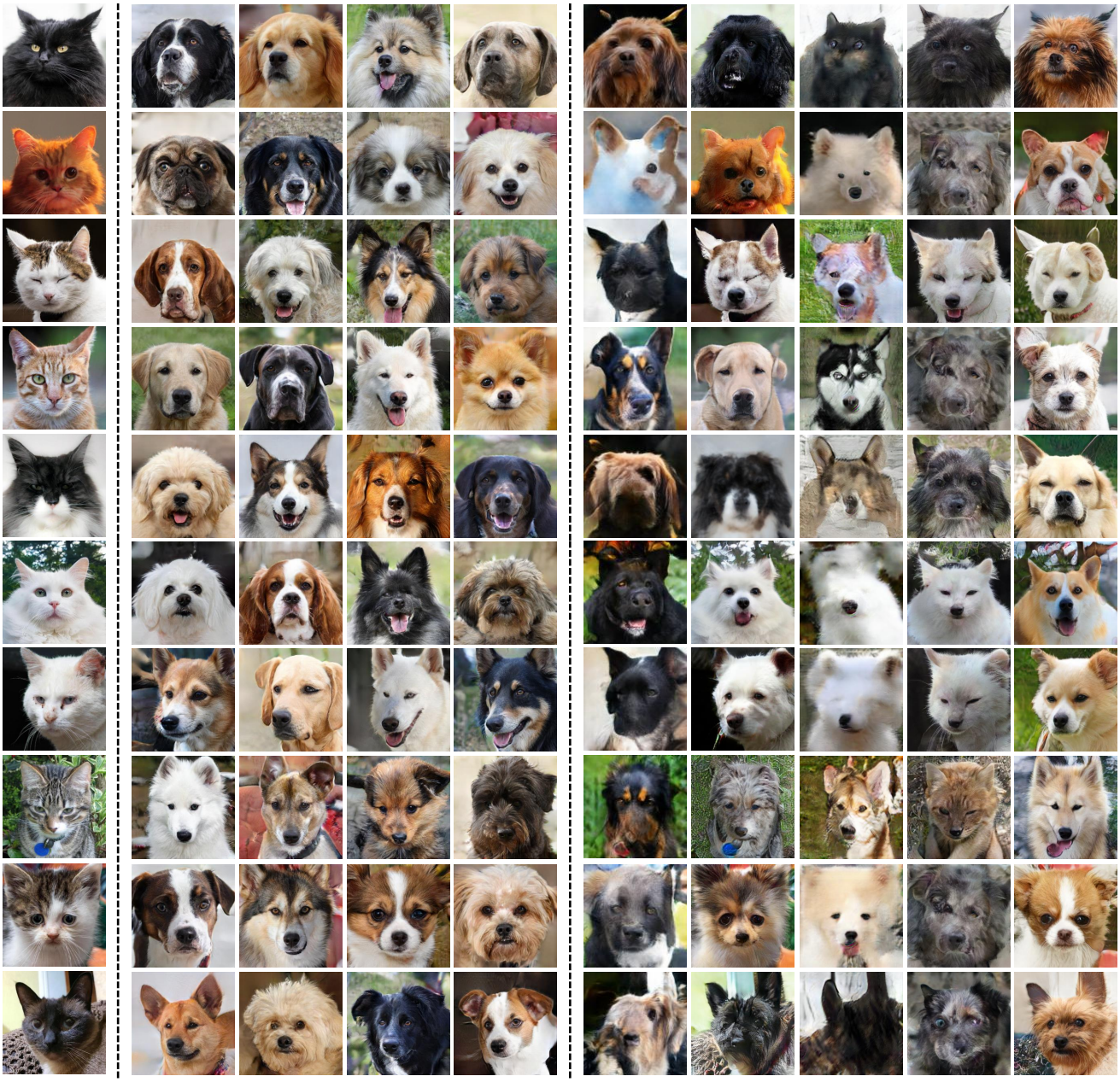


Figure 6. Visual comparison on Cat → Dog with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

Figure 7. Visual comparison on Human Face → Cat with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

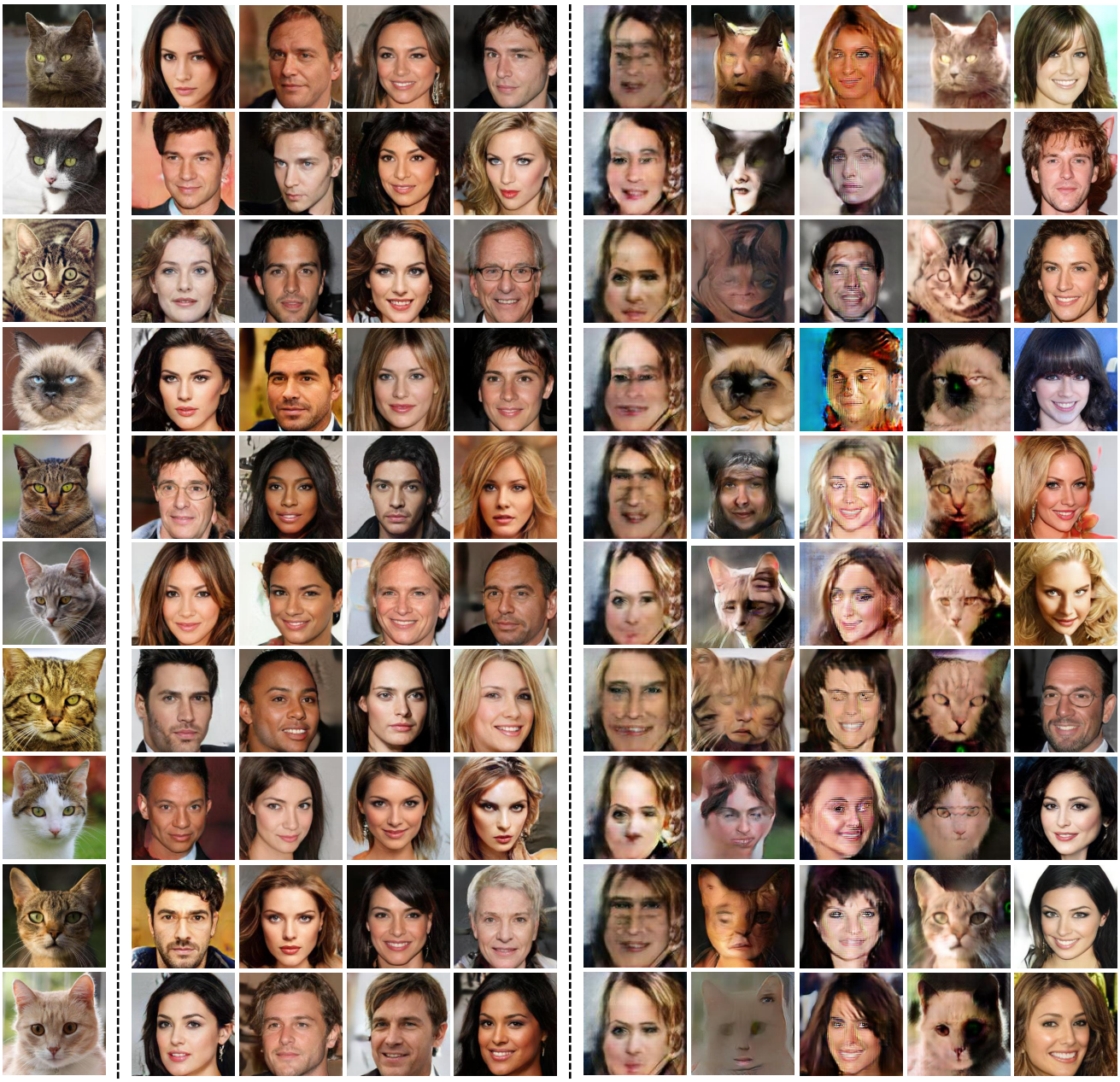


Figure 8. Visual comparison on Cat → Human Face with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

Figure 9. Visual comparison on Dog → Bird with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

Figure 10. Visual comparison on Bird → Dog with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

Figure 11. Visual comparison on Car → Bird with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

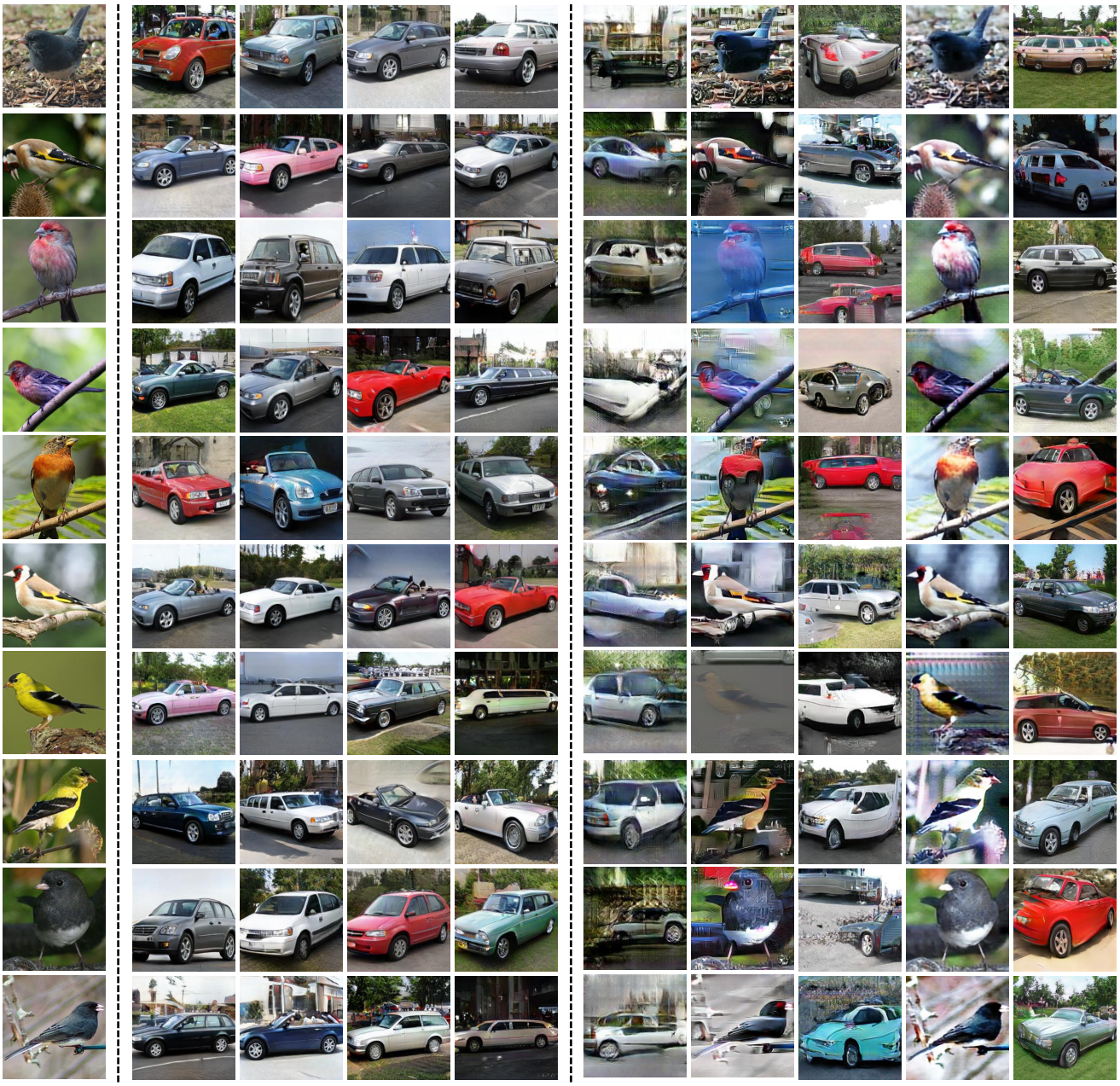


Figure 12. Visual comparison on Bird → Car with TraVeLGAN [2], U-GAT-IT [11], MUNIT [8], COCO-FUNIT [16] and StarGAN2 [5].

2.1.2 User study

We conduct a user study to evaluate the input-output content consistency and overall translation performance. A total of 25 subjects participate in this study to select the best ones from the results of six methods. Because in some tasks like Male $\leftrightarrow$ Female, the performance of each method is similar, we allow multiple selections. For each selection, if a user select results from N methods as the best results, those methods get $1/N$ scores, and other methods get 0 scores. A total of 2,500 selections on 50 groups of results (Every first five groups of Figs. 3-12) are tallied. Table 1 and Table 2 demonstrate the average user scores, where the proposed method receives notable preference for both content consistency and overall performance.

Table 1. User preference scores in terms of content consistency. Best scores are marked in bold.

Task	Male $\leftrightarrow$ Female	Dog $\leftrightarrow$ Cat	Human Face $\leftrightarrow$ Cat	Bird $\leftrightarrow$ Dog	Bird $\leftrightarrow$ Car	Average
TraVeLGAN	0.017	0.015	0.004	0.015	0.008	0.012
U-GAT-IT	0.195	0.099	0.032	0.022	0.032	0.076
MUNIT	0.111	0.016	0.078	0.000	0.022	0.045
COCO-FUNIT	0.092	0.104	0.036	0.057	0.036	0.065
StarGAN2	0.232	0.310	0.170	0.175	0.106	0.199
GP-UNIT	**0.353**	**0.456**	**0.680**	**0.731**	**0.796**	**0.603**

Table 2. User preference scores in terms of overall preference. Best scores are marked in bold.

Task	Male $\leftrightarrow$ Female	Dog $\leftrightarrow$ Cat	Human Face $\leftrightarrow$ Cat	Bird $\leftrightarrow$ Dog	Bird $\leftrightarrow$ Car	Average
TraVeLGAN	0.006	0.012	0.000	0.000	0.009	0.006
U-GAT-IT	0.162	0.079	0.001	0.004	0.005	0.050
MUNIT	0.099	0.007	0.053	0.000	0.009	0.033
COCO-FUNIT	0.098	0.085	0.000	0.033	0.004	0.044
StarGAN2	0.240	0.240	0.153	0.157	0.063	0.171
GP-UNIT	**0.394**	**0.576**	**0.793**	**0.805**	**0.910**	**0.696**

2.1.3 Content correspondence

Discussion on content correspondence. In this paper, we mainly use user scores to evaluate the content consistency. For objective evaluation, landmark correspondence might be one potential metric. We conduct human/cat face landmark detection to predict eye and nose correspondences. GP-UNIT is comparable to other baselines in normalized point-to-point error (GP-UNIT/MUNIT/StarGAN2/COCO-FUNIT/TraVeLGAN: 0.27/0.23/0.19/0.36/0.31). However, these scores still does not well match the subjective user scores in Table 1. The reason is the **content-style trade-off problem** as discussed in Sec. 2.2. A robust method should adjust the locations of facial features to match the target domains, which does not favor this metric. Since this adjustment is task-dependent, it is nearly impractical to define a universal metric. Therefore, landmark correspondence is less explored as evaluation metrics in UNIT baselines. UNIT still lacks a good objective evaluation metric for content consistency, which is an important research direction.

2.2. Comparison with TGaGa

In Fig. 13, we present translation results with large geometric deformations. As one of the most related works to ours that deal with drastic geometric deformations, TGaGa [18] and our method both effectively build geometric correspondences between two distant domains. The main superiority of GP-UNIT over TGaGa lies in more realistic texture generation. We compare with TGaGa on multi-modal generation in Fig. 14. The results of TGaGa are blurry and our method generates more vivid details.

For large geometric deformations, there is an inherent content-style trade-off problem, which is valuable to further discuss:

- **Content-style trade-off**: Due to the inherent differences in the proportions of facial features of different species, it is impossible to generate a realistic human or dog face from a cat face with the locations of eyes/nose/mouth unchanged. A robust method should adjust the locations of such facial features to match the style of target domains while maintaining the original geometry as much as possible. Therefore, there is a trade-off between realism and content consistency.
- **Cycle consistency overemphasizes content**. Standard cycle consistency is often too restrictive and results in only texture transfer without geometry adjustment. Therefore, methods like MUNIT overemphasize content consistency, sacrificing realism as in Figs. 5-6.
- **GP-UNIT strikes a good balance**. TGaGa solves this problem with explicit geometry adjustment, while we learn a high-level correspondence based on which only necessary mid-level correspondences are then added. It can be seen in Fig. 13(a) and Fig. 14 that both TGaGa and GP-UNT successfully adjust the geometry, but in our results, the locations of the facial features better match the input (although not exactly the same), which proves that GP-UNIT strikes a better balance between realism and content consistency than TGaGa.

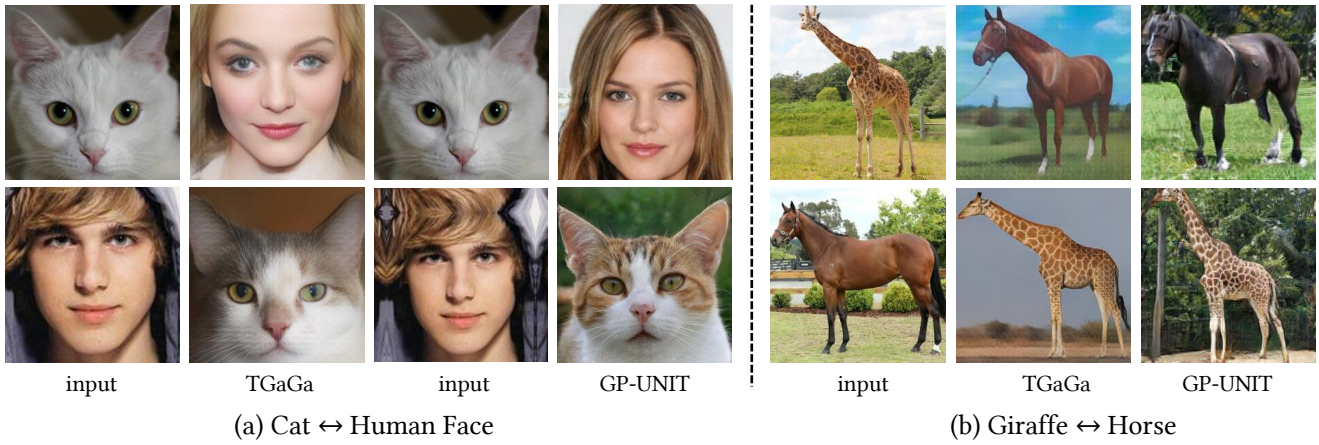


Figure 13. Comparison in geometry-preserving with TGaGa[1].

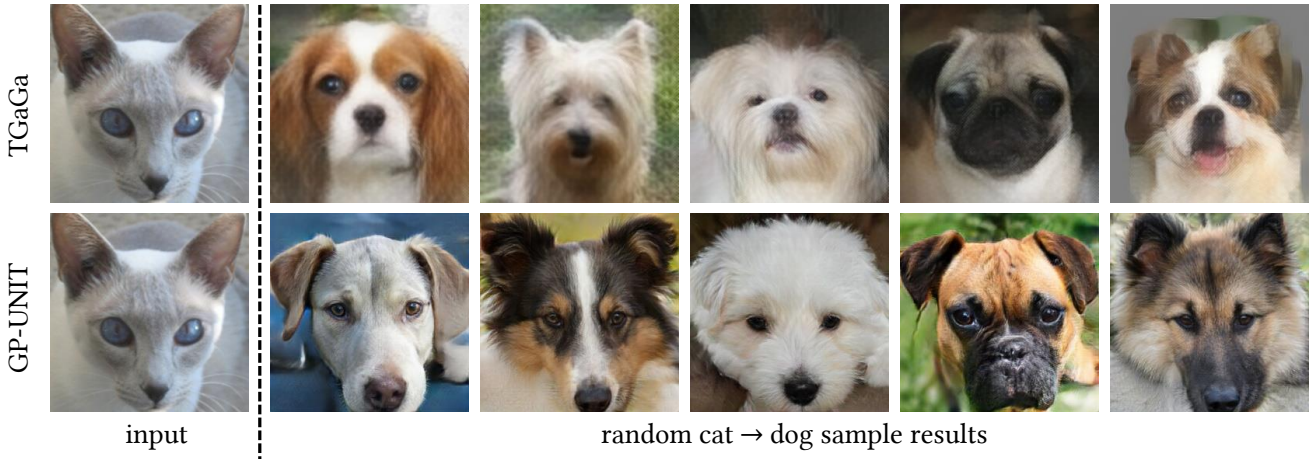


Figure 14. Comparison in multi-modal generation with TGaGa.

2.3. Comparison with StyleGAN Prior Guided Model

In Figs. 15-16, we compare with UI2I-Style [12] to further analyze the limitation of StyleGAN prior compared to our method. StyleGAN-based methods achieves unsupervised translations between two domains using finetuning [14, 12][2]. By assuming a small distance between the models before and after finetuning, images generated by the original StyleGAN and the finetuned StyleGAN belong to two domains, respectively, but have strong content correspondences. Layer-swap is proposed by [14] to control how many content features from the source domain are preserved. UI2I-Style [12] shows good results on Face $\rightarrow$ Art Portrait and Cat $\rightarrow$ Dog with layer- swap at resolution 16×16. However, when we apply UI2I-Style to domains with more visual discrepancies like human face and cat, layer- swap results in fused and very unreal results. Even without using layer- swap, the assumption of a small distance between the models does not hold. Therefore, the content correspondences using the same latent code are drastically weakened. For example, the position of the eyes of the cats does not match those of human faces. By comparison, the results of our method are better in content consistency. When it comes to more challenging Dog $\leftrightarrow$ Bird, the results of UI2I-Style has little consistency with the input image. Moreover, the performance of StyleGAN relies on sufficient training data. Even we use the adaptive discriminator augmentation [10], 2.4K dog training images and 2.4K bird training images seem to be not enough for StyleGAN to produce high-quality results.

In summary, our method is superior to StyleGAN prior guided models in the following aspects:

- GP-UNIT is capable of translations between domains with high discrepancy that StyleGAN prior is not applicable to;
- GP-UNIT can also handle the translation tasks between close domains that StyleGAN prior based methods mainly solve. We expand their application scenarios;
- Our way of distilling BigGAN prior is different from the mainstream way of using StyleGAN prior, which enables us to learn universal content features applicable to various tasks without retraining the content encoder, while for each task, a StyleGAN of a certain domain need to be pretrained;
- Our framework can produce high-quality results with less training data than StyleGAN.

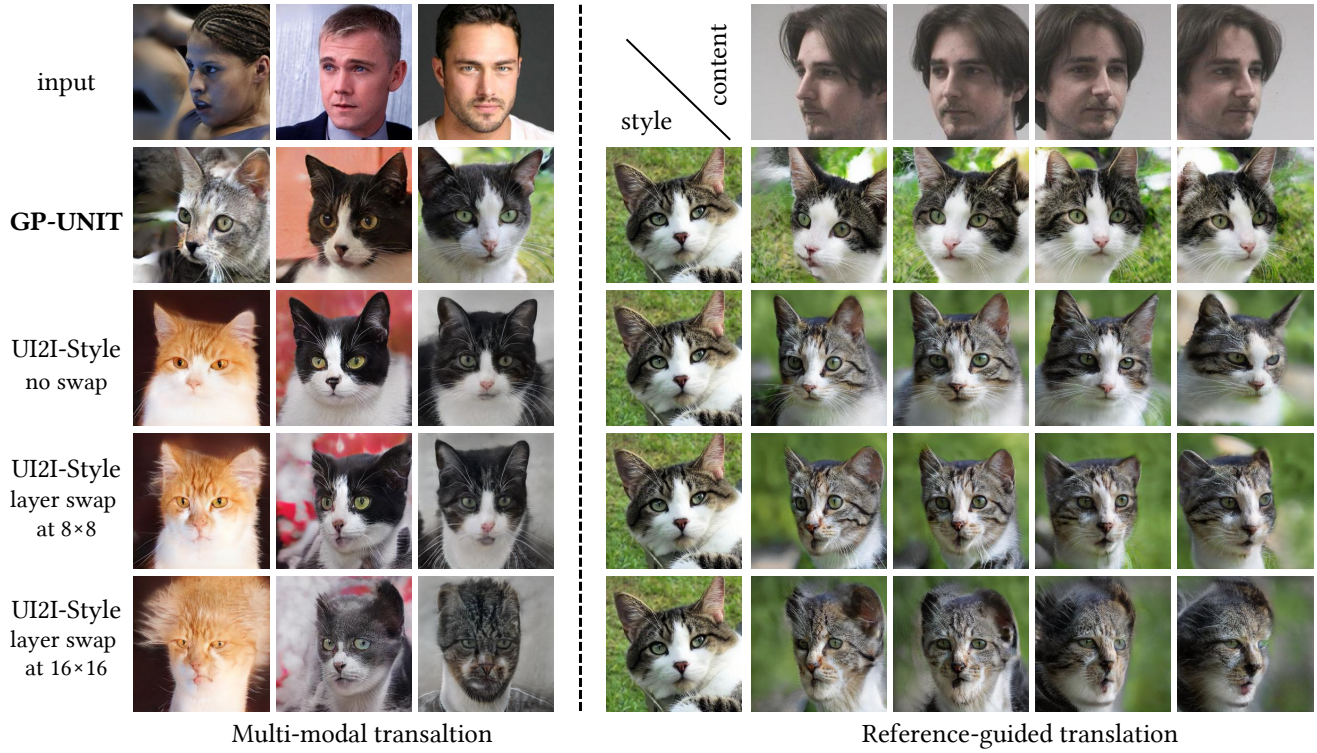


Figure 15. Visual comparison on Human Face $\rightarrow$ Cat with UI2I-Style [12].

[1]The training data of GP-UNIT and TGaGa does not match (CelebA-HQ and CelebA). For Human Face $\rightarrow$ Cat, GP-UNIT uses reflect padding to enlarge the non-face region of the content image to match the scale of our training data.

[2]Although the W+ space of StyleGAN enables image reconstruction in arbitrary domains, the W+ latent code still does not support semantic editing like interpolation, style fusing and translation between domains beyond the domains StyleGAN is trained or finetuned on [1]

Figure 16. Visual comparison on Dog ↔ Bird with UI2I-Style [12]

2.4. Ablation Study

2.4.1 Dynamic skip connection

As shown in Fig. 17, without dynamic skip connections (DSC), the content features $E_c(x)$ at the most abstract level can only provide very rough content information like the position of the head and the ears (the dark region in $E_c(x)$). And the resulting dog faces fail to match the pose of the input cat faces. For example, the cat in the fourth row is facing forward, while the generated dog without DSC is facing left. The dynamic skip connections learn to locate the key eye features and mouth features through the 301st and the 135th channels of the mask m^1, respectively, which effectively provides fine-level content correspondences. Therefore, our full model can keep the relative position of the nose and eyes as in the input cat faces.

Figure 17. Ablation study on the dynamic skip connection.

2.4.2 Generative prior distillation

As shown in Fig. 18, if we train our content encoder from scratch along with all other subnetworks in the second stage, like most image translation frameworks, our model fails to preserve the content features such as the head pose. By comparison, our pre-trained content encoder successfully exploits the generative prior to build effective content mappings.

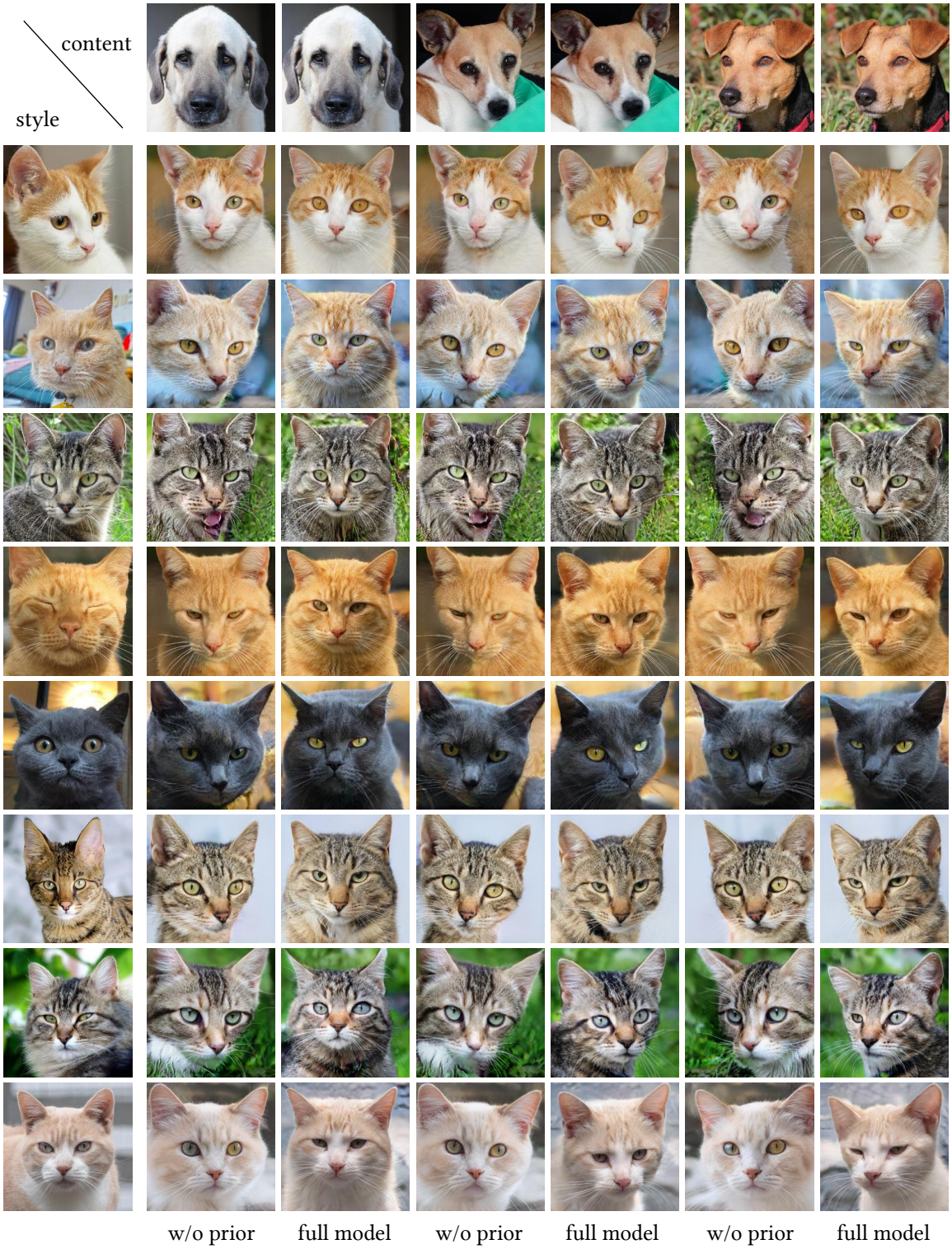


Figure 18. Ablation study on the generative prior.

2.4.3 Quantitative comparison

To better understand the effect of the submodules, we perform quantitative comparison in terms of quality, diversity and content consistency. For content consistency, ten users are invited to select the best one from the results of three configurations in terms of content consistency. FID, Diversity averaged over the whole testing set and the user score averaged over six groups of results are presented in Table 3.

- **FID**: Results of our full model have better quality (low FID)
- **Diversity**: Three configurations have comparable diversities. With fewer content constraints, results of model without the generative prior or without dynamic skip connection are more diverse.
- **Content Consistency**: The generative prior and the dynamic skip connection effectively help our model better capture high-level and mid-level content correspondences (high Content Consistency)

Table 3. Ablation study on DIF, Diversity and input-output content consistency. Best scores are marked in bold.

Metric	FID	Diversity	Content Consistency
without generative prior	16.11	**0.55**	0.02
without dynamic skip connection	15.83	0.52	0.15
full model	**15.29**	0.51	**0.83**

2.5. Analysis on Multi-Level Content Feature

In Fig. 19, we analyze the effect of our multi-level content features on Cat → Wildcat. The most abstract $E_c(x)$ only gives layout cues and can solely generate a rough tiger face without details. Meanwhile, m^1 focuses on distinct details (*e.g.*, nose and eyes in #305, coarse-grained cat whiskers in #321, foreheads in #299), which is enough to generate realistic results with $E_c(x)$. Finally, m^2 pays attention to subtle details (*e.g.*, finer-grained cat whiskers in #169). Therefore, our full multi-level content features enable us to simulate the extremely fine-level long whiskers in the input as indicated by the difference maps. Note that the learned attentions are both channel-wise (28 out of 512 for m^1 and 2 out of 256 for m^2 have large activation) and spatial-wise sparse. Such reasonable semantic attentions are learned merely via a generation task, without any explicit correspondence supervision.

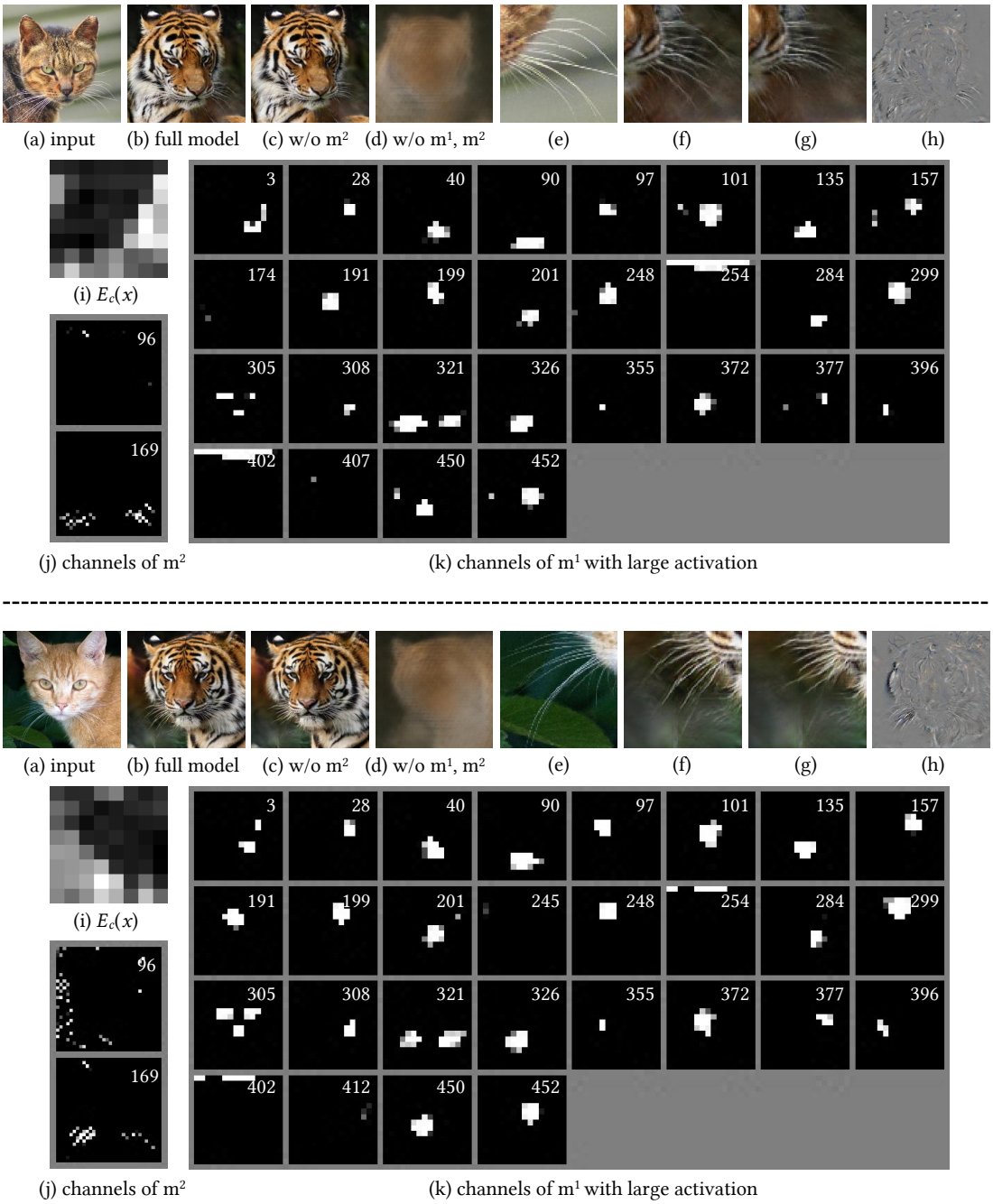
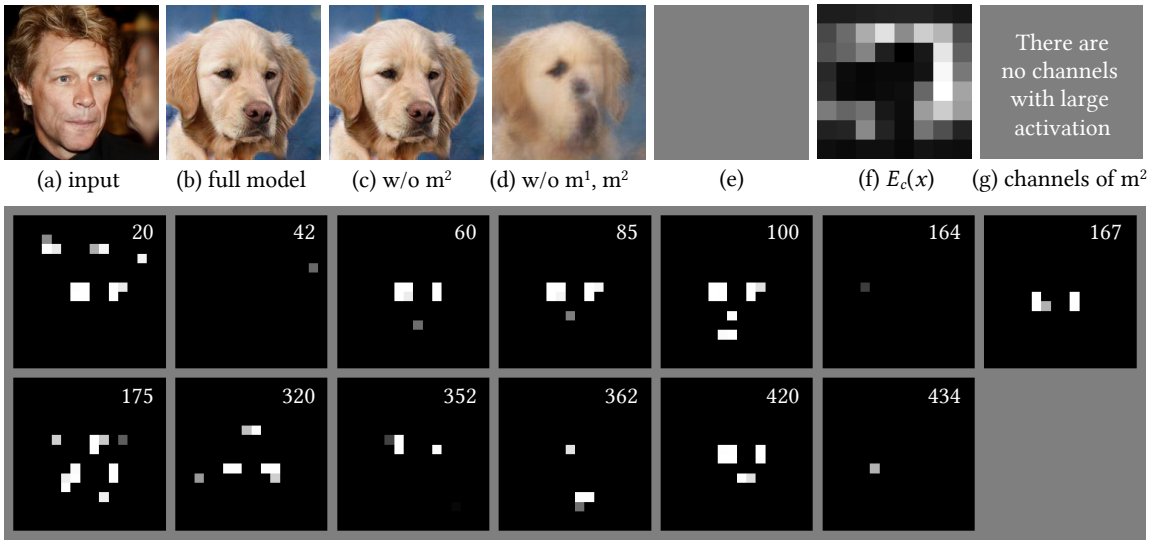


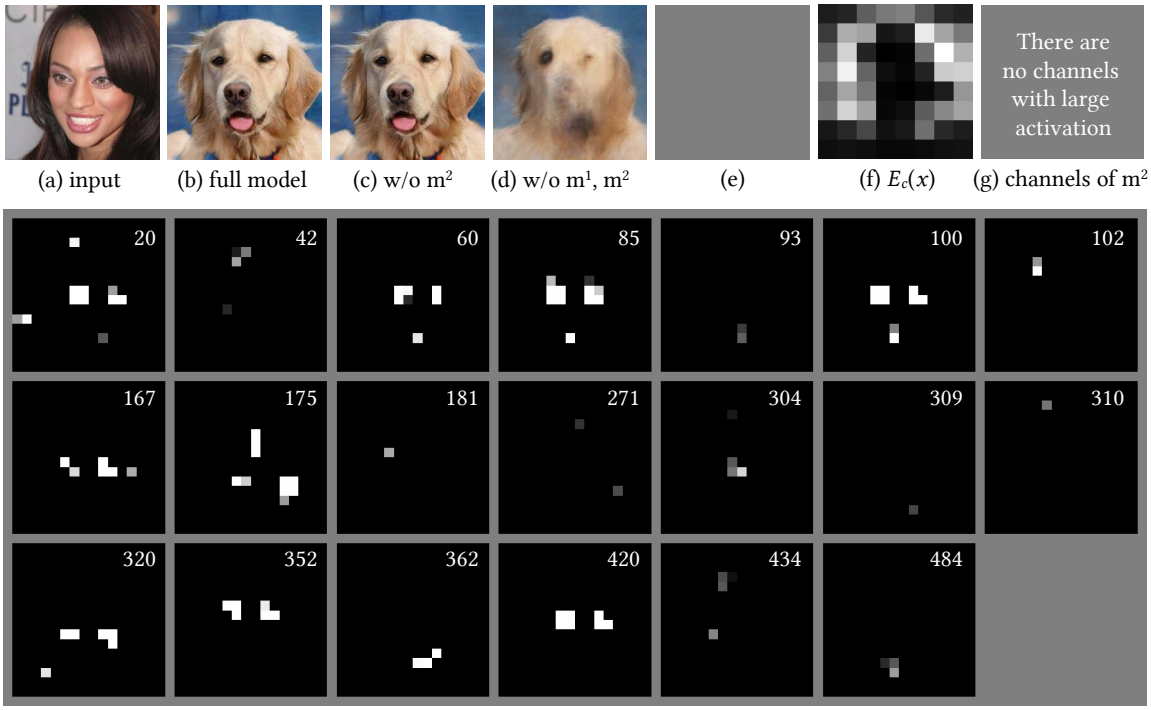
Figure 19. Effect of the multi-level content features. (a): Input. (b)-(d): Results by full model, by setting m^1 to **0**, by setting both m^1 and m^2 to **0**, respectively. (e)-(g): Local enlarged region of (a)-(c), respectively. (h): Difference map between (b) and (c). (i): $E_c(x)$. (j)-(k): Channels of m^1 and m^2 with activation values greater than 0.2. The channel index is on the top right of each channel.

In Fig. 20, we analyze the effect of our multi-level content features on Dog → Human Face. The most abstract $E_c(x)$ only gives layout cues and can solely generate a rough dog face with blurry eyes, noses and mouths in the wrong positions. Meanwhile, m^1 focuses on distinct details (*e.g.*, eyes in #60, #85, #100, #352 and #420), which helps locate eyes, noses and mouths to generate realistic results with $E_c(x)$. Finally, we observe that there is no valid activation in m^2 for fine-level content correspondences, likely due to the fact that dogs and humans have a large appearance disparity. Therefore, our method automatically ignores the content features at this level.



(a) input (b) full model (c) w/o m^2 (d) w/o m^1, m^2 (e) (f) $E_c(x)$ (g) channels of m^2

(h) channels of m^1 with large activation



(a) input (b) full model (c) w/o m^2 (d) w/o m^1, m^2 (e) (f) $E_c(x)$ (g) channels of m^2

(h) channels of m^1 with large activation

Figure 20. Effect of the multi-level content features. (a): Input. (b)-(d): Results by full model, by setting m^1 to **0**, by setting both m^1 and m^2 to **0**, respectively. (e): Difference map between (b) and (c). (f): $E_c(x)$. (g)-(h): Channels of m^1 and m^2 with activation values greater than 0.2. The channel index is on the top right of each channel.

In Fig. 21, we analyze the effect of our multi-level content features on Bird → Car. The most abstract $E_c(x)$ can solely generate a rough layout of the car. The meaning of m^1 is not as intuitive as other tasks, since birds and cars have almost no semantic relationship. However, we can still find some reasonable cues. For example, the channels #9, #240 and #258 of m^1 focuses on the background areas around the foreground object. The channels #134 and #499 of m^1 locate the bird tails and backs, respectively. Finally, the channel #232 of m^2 extracts high-frequency signals (similar to isophote map) from the input to add texture details in the background, as indicated by the difference map. Such behavior reduces the learning difficulty of the generator G, allowing G to better focus on the realistic structure synthesis. In Figs. 19-20, we do not find such behavior, likely due to the fact that the backgrounds in AFHQ [5] and CelebA-HQ [9] are mostly blurry with few high-frequency details.

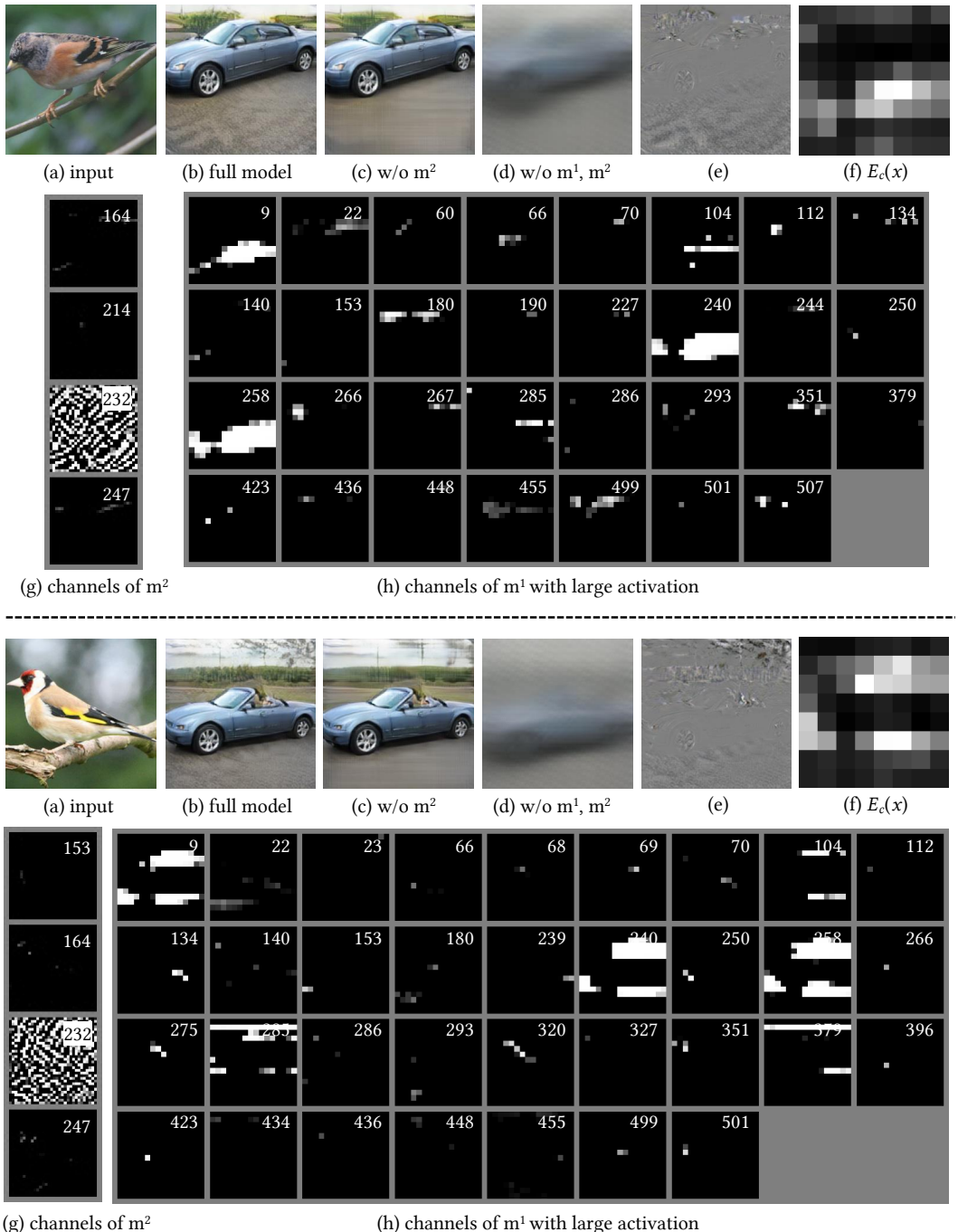


Figure 21. Effect of the multi-level content features. (a): Input. (b)-(d): Results by full model, by setting m^1 to **0**, by setting both m^1 and m^2 to **0**, respectively. (e): Difference map between (b) and (c). (f): $E_c(x)$. (g)-(h): Channels of m^1 and m^2 with activation values greater than 0.2. The channel index is on the top right of each channel.

2.6. Generalization to the Giraffe Domain

In Fig. 22, we show additional translation results between birds and giraffes.

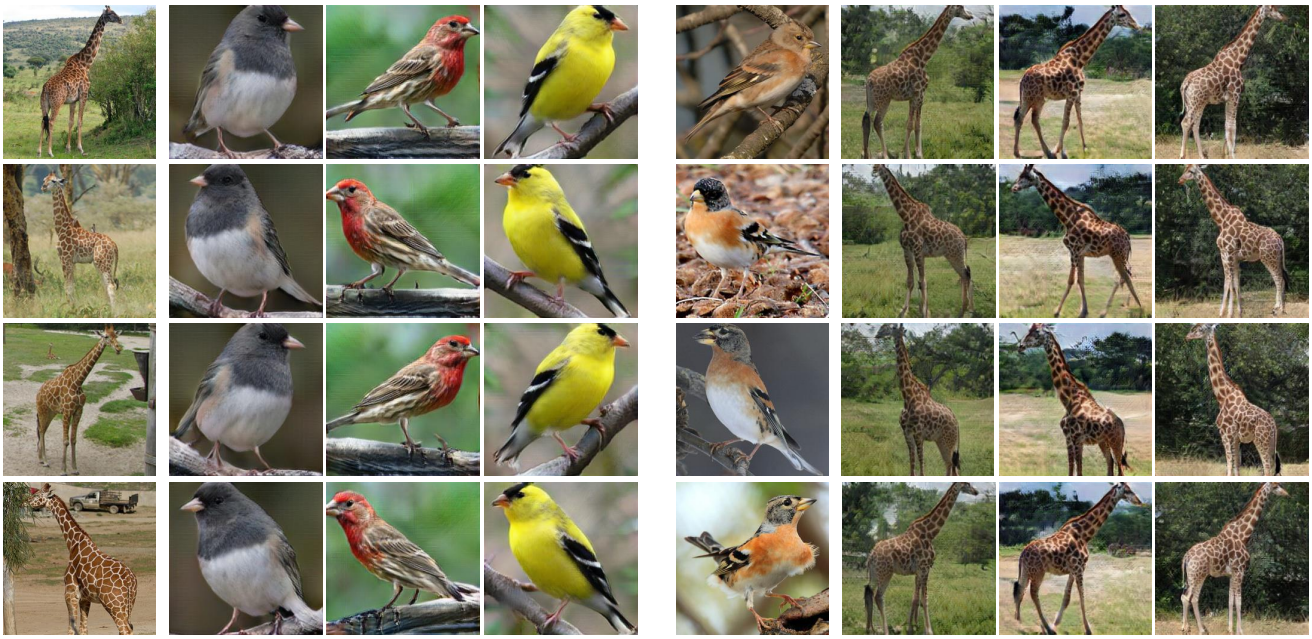


Figure 22. Performance on the Giraffe domain from MS-COCO dataset.

2.7. Style Blending

In Figs. 23-25, we perform a linear interpolation to style feature, and observe smooth changes along with the latent space from one to another, while keeping the high-level content feature intact.

Figure 23. Unseen style blending on cars. The anchor styles are marked by red boxes.

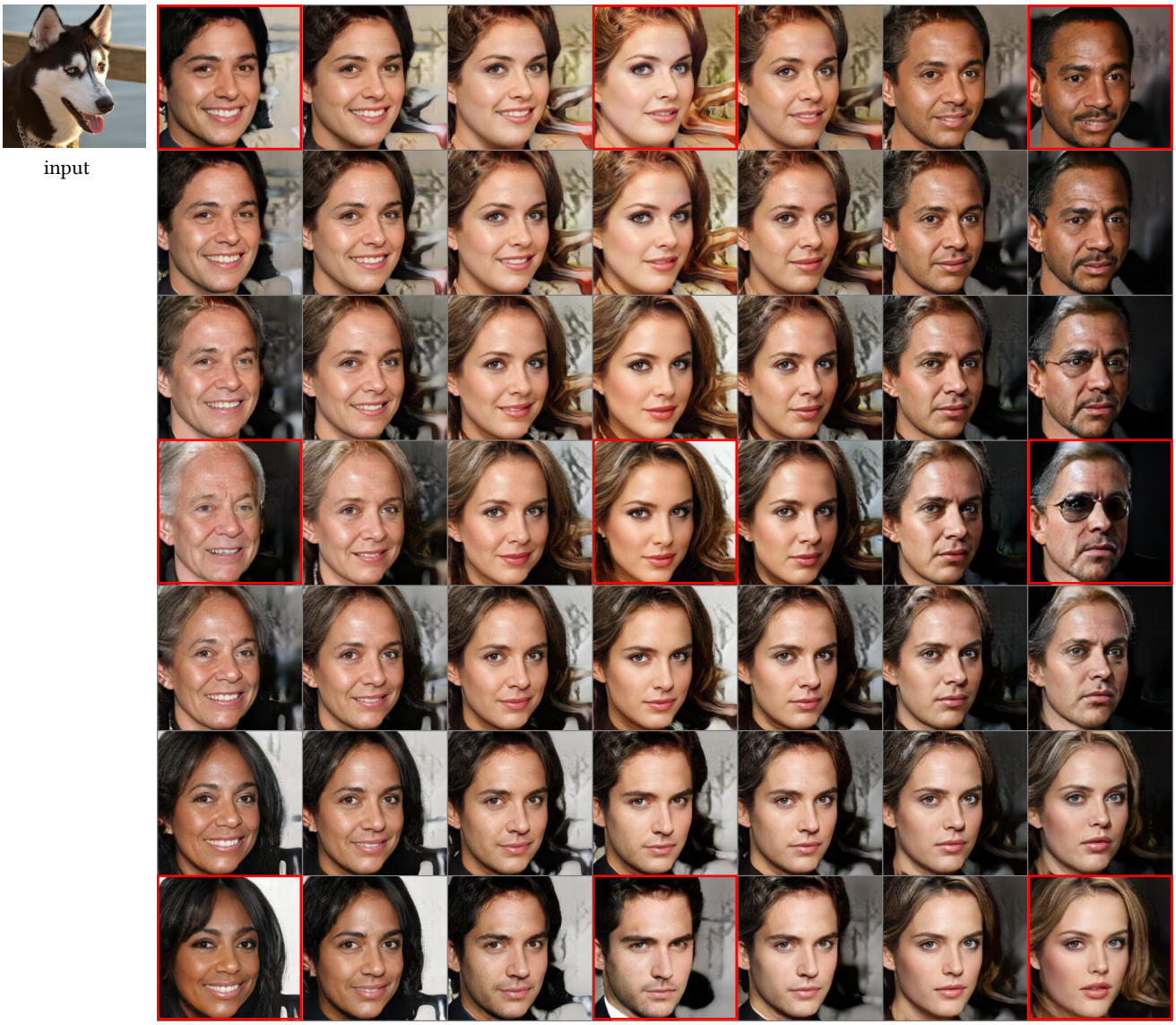


Figure 24. Unseen style blending on human faces. The anchor styles are marked by red boxes.

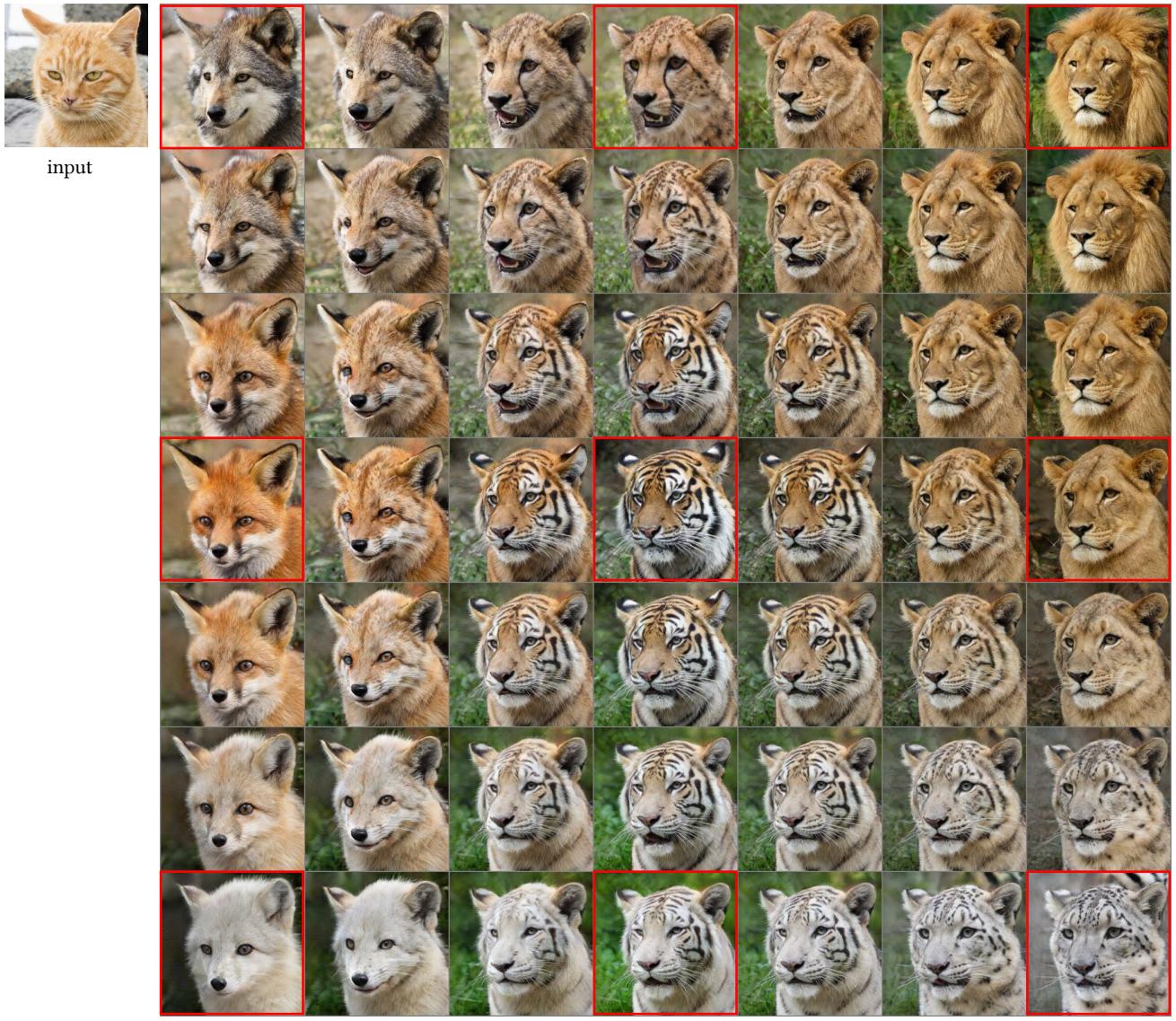


Figure 25. Unseen style blending on wild animal faces. The anchor styles are marked by red boxes.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proc. Int'l Conf. Computer Vision*, pages 4432–4441, 2019. 18

[2] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8983–8992, 2019. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proc. Int'l Conf. Learning Representations*, 2019. 2

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 2

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 25

[6] Nicolas Gourier, Daniela Hall, and James L Crowley. Estimating face orientation from robust detection of salient facial features. In *ICPR International Workshop on Visual Observation of Deictic Gestures*. Citeseer, 2004. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[8] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. European Conf. Computer Vision*, pages 172–189. Springer, 2018. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *Proc. Int'l Conf. Learning Representations*, 2018. 2, 25

[10] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020. 2, 18

[11] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *Proc. Int'l Conf. Learning Representations*, 2019. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

[12] Sam Kwong, Jialu Huang, and Jing Liao. Unsupervised image-to-image translation via pre-trained stylegan2 network. *IEEE Transactions on Multimedia*, 2021. 18, 19

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conf. Computer Vision*, pages 740–755. Springer, 2014. 2

[14] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. In *Machine Learning for Creativity and Design NeurIPS Workshop*, 2020. 18

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int'l Journal of Computer Vision*, 115(3):211–252, 2015. 2

[16] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. In *Proc. European Conf. Computer Vision*. Springer, 2020. 2, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int'l Conf. Learning Representations*, 2015. 2

[18] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 8012–8021, 2019. 2, 17