LinkGAN: Linking GAN Latents to Pixels for Controllable Image Synthesis
Supplementary Material

Jiapeng Zhu[†*1] Ceyuan Yang[†2] Yujun Shen[†3] Zifan Shi[*1] Bo Dai[2] Deli Zhao[4] Qifeng Chen[1]
[1]HKUST [2]Shanghai AI Laboratory [3]Ant Group [4]Alibaba Group

This paper proposes LinkGAN that explicitly links some latent axes to a region of an image or a semantic by utilizing an easy yet powerful regularizer. In this supplementary material, we first give the implementation details of our method in Sec. 1. Second, more results are given in Sec. 2, including comparisons with other methods and more quantitative and qualitative results of our methods. Third, we give an additional ablation study in Sec. 3 besides the one offered in the main text, *i.e.*, the problem of image inconsistency after resampling. Fourth, we give some discussions in Sec. 4.

1. Implementation Details

We use the official Pytorch implementation of StyleGAN2 [5] and official Pytorch implementation of EG3D [1] to validate our method. We keep all the parameters untouched except our newly added regularizer during training. We followed the original codebase to compute FID, and for the masked MSE, we calculated it on 10,000 images for each edit. The update frequency of our lazy regularization is 8. For how many axes we use to control the specific region, we list below: 1). For small regions, we use 64 axes, such as the eye, nose, mouth, and ear region on FFHQ or AFHQ. 2). For the larger region, such as the left region of the human face and the bottom part of the church in Fig.3 of the main text, we use 128 axes. Also, for linking the whole image of Fig.5 in the main text along with two (Fig. 5, Fig. 7) in this *Supplementary Material*, each part has a size of 128 since we evenly split the latent space. 3). When the partition size becomes bigger, such as half of the image, we use 256 axes, and for the semantic control (church, sky, and car) in Fig.4 of the main text, we use 256 axes as well. For the loss weight λ_1 and λ_2, we list as below: 1). For the latent segment with 64 axes, we set λ_1 equal to 0.04 and λ_2 equal to 0.01. 2). For the latent segment with 128 axes, we set λ_1 equal to 0.03 and λ_2 equal to 0.01. 3). For the latent segment with 256 axes, we set λ_1 equal to 0.02 and λ_2 equal to 0.02. The perturbation strength α is set to one in all the experiments.

Training time. Recall that we are just finetuning the generator from an official checkpoint. Hence, building such a link will not take much time, which usually takes $4 \sim 8$ hours, depending on the dataset. However, achieving a better FID requires a longer training time. We also do an experiment that trains a GAN from scratch on the FFHQ dataset and then involves the regularizer when the FID is lower than 10, in which we get similar results regarding the controllability and the generation performance compared to finetuning. In such cases, the train time is roughly equal to the original StyleGAN when getting the smallest FID value for each training.

2. More results

Comparison with existing methods. In our main text, we give the quantitative comparison results with some existing methods (Tab.2 in the main text). Here, we show the corresponding qualitative results. Fig. 1, Fig. 2, and Fig. 3 give the comparison results with ReSeFa [10] and StyleCLIP [6] on eyes, nose, and mouth regions. It is noteworthy that the ReSeFa is re-implemented on our fine-tuned model, and the text prompt we used in StyleCLIP when editing these three regions are: "extremely big eyes without any change in the background", "crooked nose without any change in the background", and "open mouth without any change in the background". From these three figures, we can see that our method can achieve better control precision compared to the other two. For instance, when editing the eyes, from the heatmaps, we can observe that the outlines of the face for ReSeFa and StyleCLIP are obviously changed. The same phenomenon also occurs when editing the nose and mouth regions. On the contrary, our method has negligible changes in the regions when editing a specific region thanks to our explicit link. For Stylespace [8], we use the code released here to manipulate eyes, nose, and mouth. And the results are shown in Fig. 4, we can see from this figure, the outline of the face is also affected. Hence, both quantitative (Tab.2 in the main text) and qualitative results demonstrate the precise control ability of our method.

More results of our methods. We first give more quantitative results when our regularizer is added to different regions and datasets. Tab. 1 and Tab. 2 report the FID on a single region and multiple regions, respectively. From Tab. 1, we can see that on some datasets and regions,

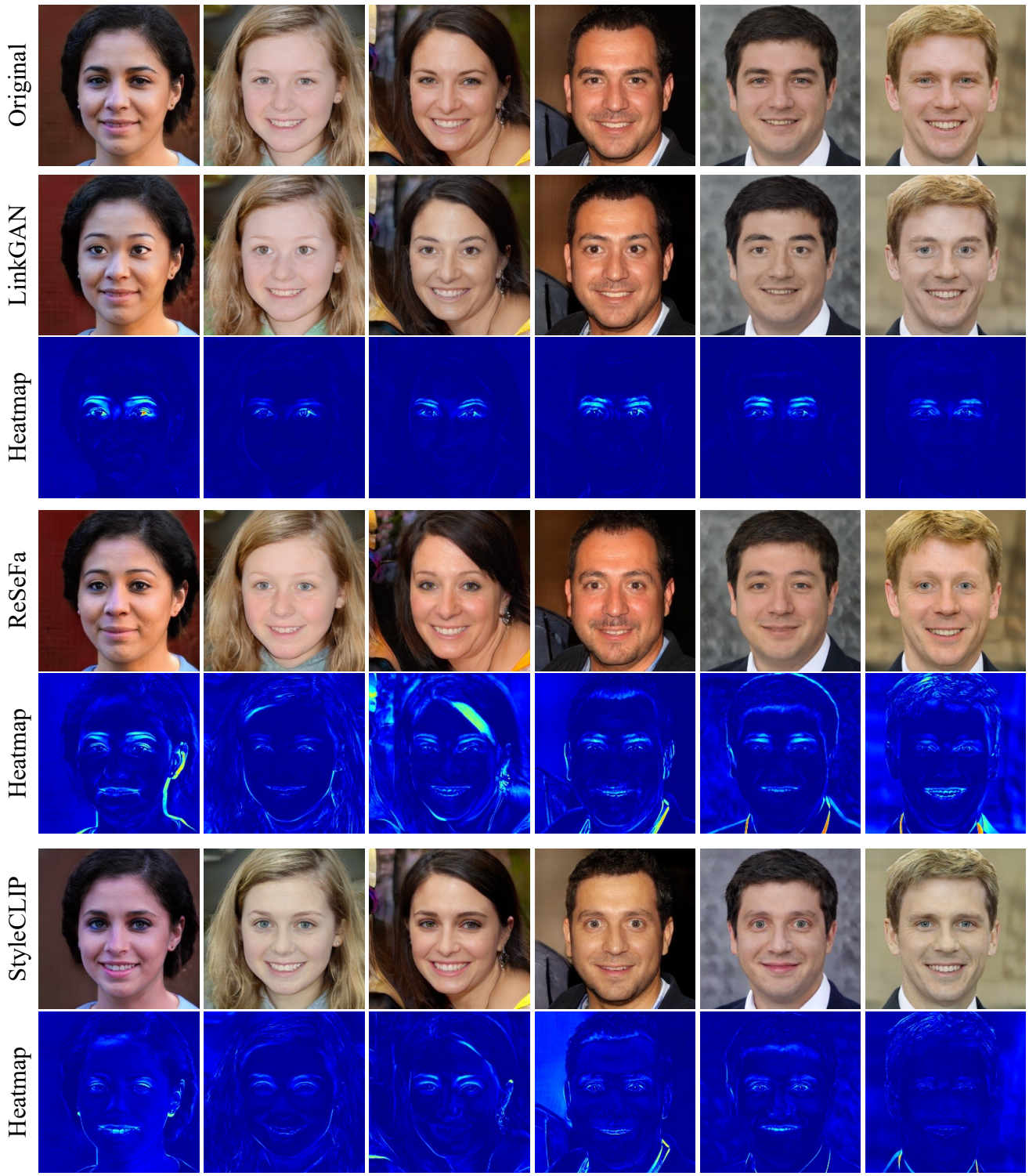

Figure 1. **Qualitative comparison** when manipulating the eyes region with ReSeFa [10] and StyleCLIP [6]. As we can see from the heatmaps, LinkGAN achieves more precise control within the regions of interest.

the FID slightly deteriorates (*e.g.*, the regions on FFHQ), and on some datasets and regions, FID is comparable, even lower than without adding our regularizer (*e.g.*, FFHQ on EG3D). We believe that a higher FID is caused by the loss of diversity because our model requires the output image to be realistic before and after local editing (*i.e.*, through

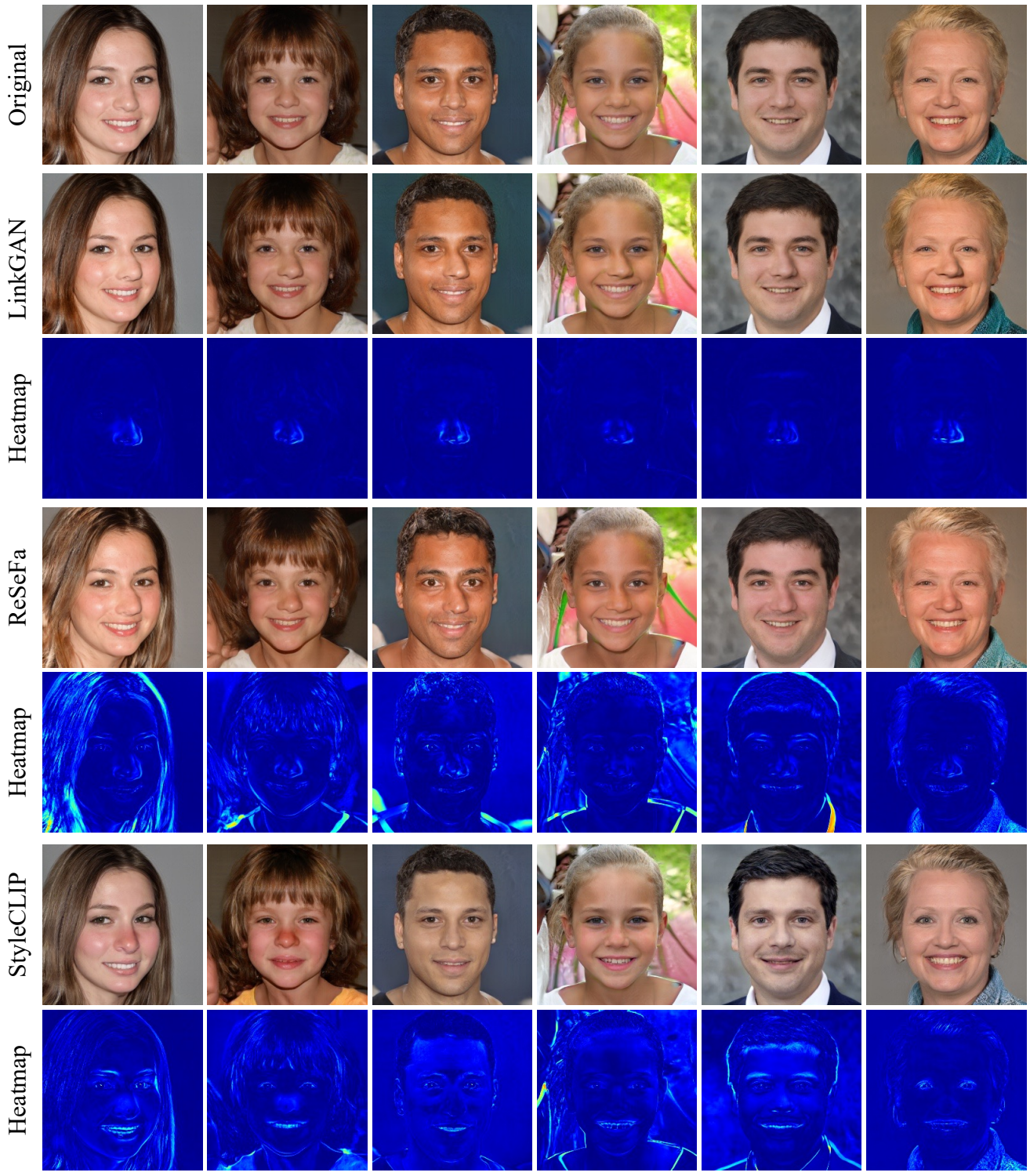

Figure 2. **Qualitative comparison** when manipulating the nose region with ReSeFa [10] and StyleCLIP [6]. As we can see from the heatmaps, LinkGAN achieves more precise control within the regions of interest.

partially resampling the latent code). Such a hypothesis can be confirmed, to some degree, by the higher precision (*i.e.*, image quality) and lower recall (*i.e.*, diversity) shown in Tab. 3. When more regions are linked, FID further deteriorates, as shown in Tab. 2. It is sensible that with more constraints, more diversity will be lost.

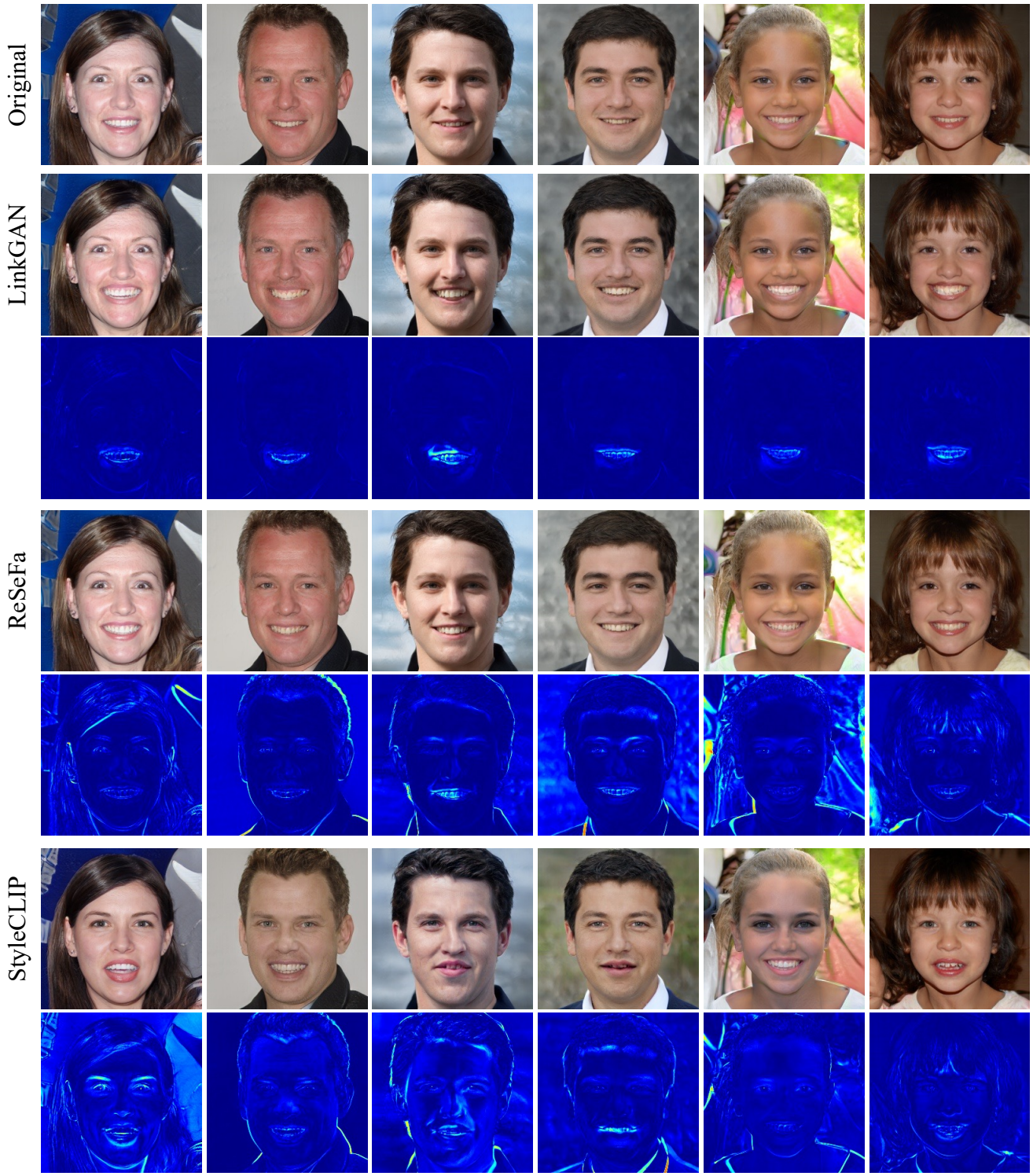

Figure 3. **Qualitative comparison** when manipulating the mouth region with ReSeFa [10] and StyleCLIP [6]. As we can see from the heatmaps, LinkGAN achieves more precise control within the regions of interest.

Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9 show more qualitative results on FFHQ [4], AFHQ [2], LSUN-car, church, and bedroom [9]. The linked regions in those figures are diverse and vary from a single fixed region to a dynamic change one, to multiple regions, and even to the whole images. For example, on human faces, as

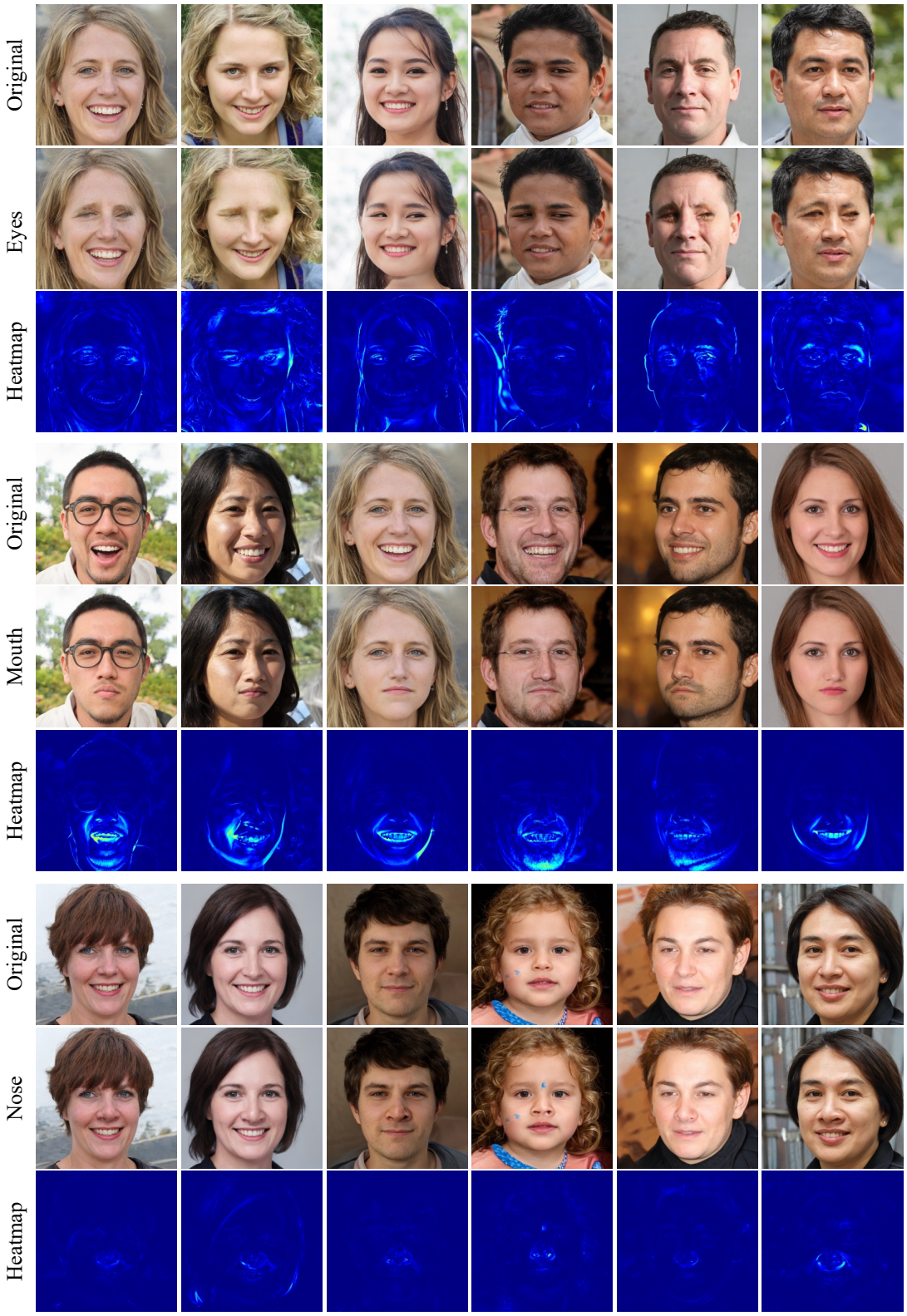

Figure 4. **Qualitative results** of StyleSpace [8] when manipulating on different regions along with corresponding heatmaps.

shown in Fig. 5, we can link either complicated or non-special semantics (*e.g.*, half of the faces, or just a cube of background) or link multiple regions to multiple latent fragments. Results on the other datasets (*e.g.*, Fig. 6, Fig. 7,

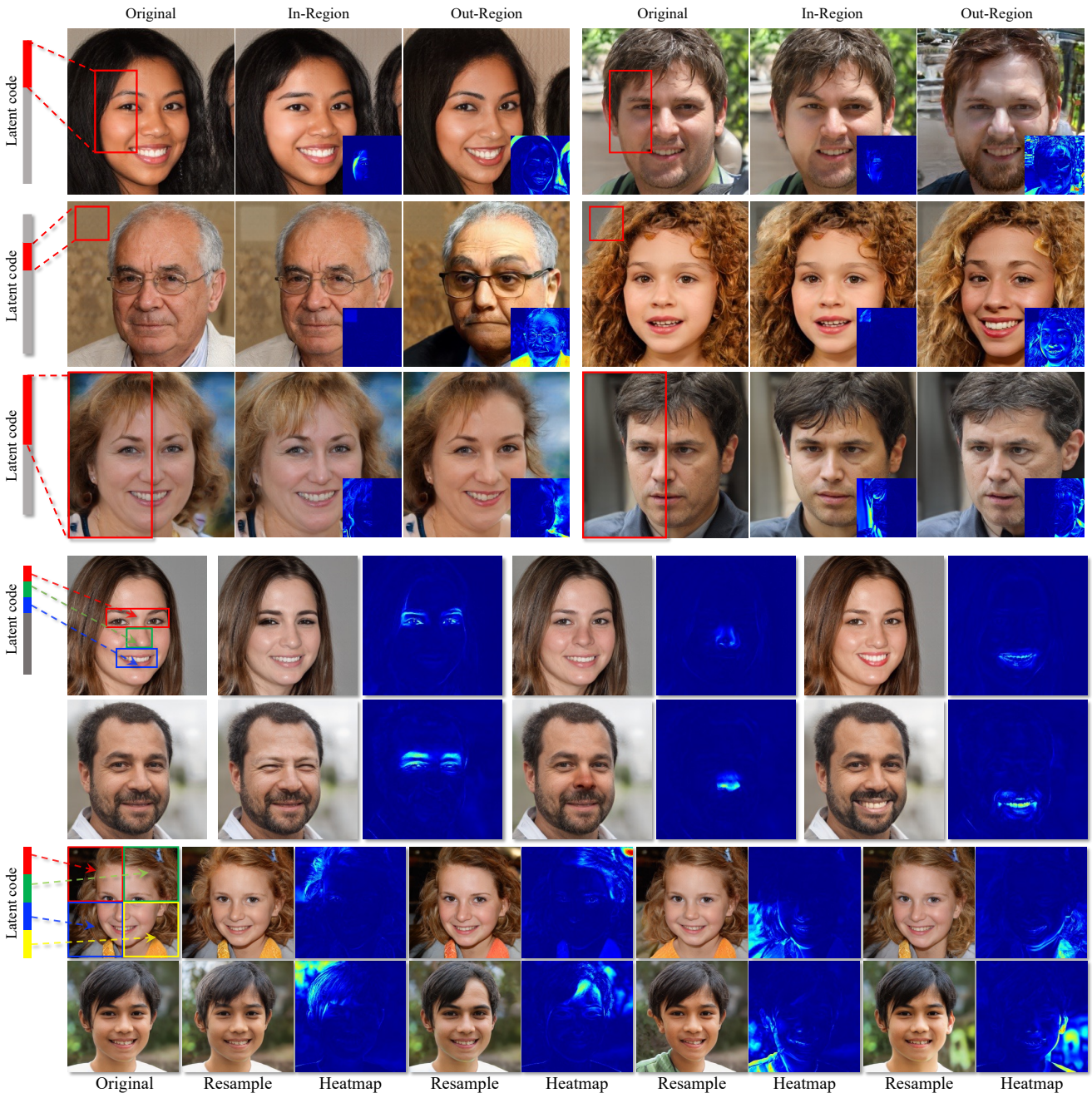

Figure 5. **Linking latents to some fixed regions** on human faces, which are pre-selected before training and shared by all instances. Linked latent subspaces and regions are highlighted with the same colors, the heatmaps reflect the change of pixel values after in-region resampling and out-region resampling. The results on the top group show linking a single region, while the results on the bottom show we can link multiple regions.

Fig. 8) also demonstrate the success of our approach in linkage building regarding different regions or semantics. Also, Fig. 9 gives the results on the 3D generative model EG3D [1] trained on AFHQ [2], which is another evidence to demonstrate the generalization ability of our regularizer.

3. Ablation Study

Here we conduct an ablation study, which is the emergence of the inconsistency after resampling. We do the study on the eye region of AFHQ [2] dataset since it has multiple classes, and we can observe this inconsistency more clearly. Fig. 10 and Fig. 11 show some qualitative

Figure 6. **Linking latents to regions** on cars. The regions in the top group are pre-selected before training and shared by all instances, while the regions in the bottom group dynamically vary across instances. Linked latent subspaces and regions are highlighted with red fragments and boxes/contours, the heatmaps reflect the change of pixel values after in-region resampling and out-region resampling. We can see that LinkGAN can link arbitrary regions no matter whether they are fixed or dynamically vary.

results when interpolating between the resampled part and the original latent part. As we can see from these two figures, the inconsistency is obvious if we directly use a different latent code on the linked segments (*i.e.*, the images with one on its bottom). When we do the interpolating with the original latent code, the inconsistency can be alleviated, especially when we involve more than half of the content of the original latent code. For instance, in the last row of Fig. 10, when the color of the resampled cat is different from the original one, the resulting image shows severe inconsistency. When we mix some content from the original latent code, the inconsistency is relieved. Hence, interpolation is an effective way to remedy this inconsistency.

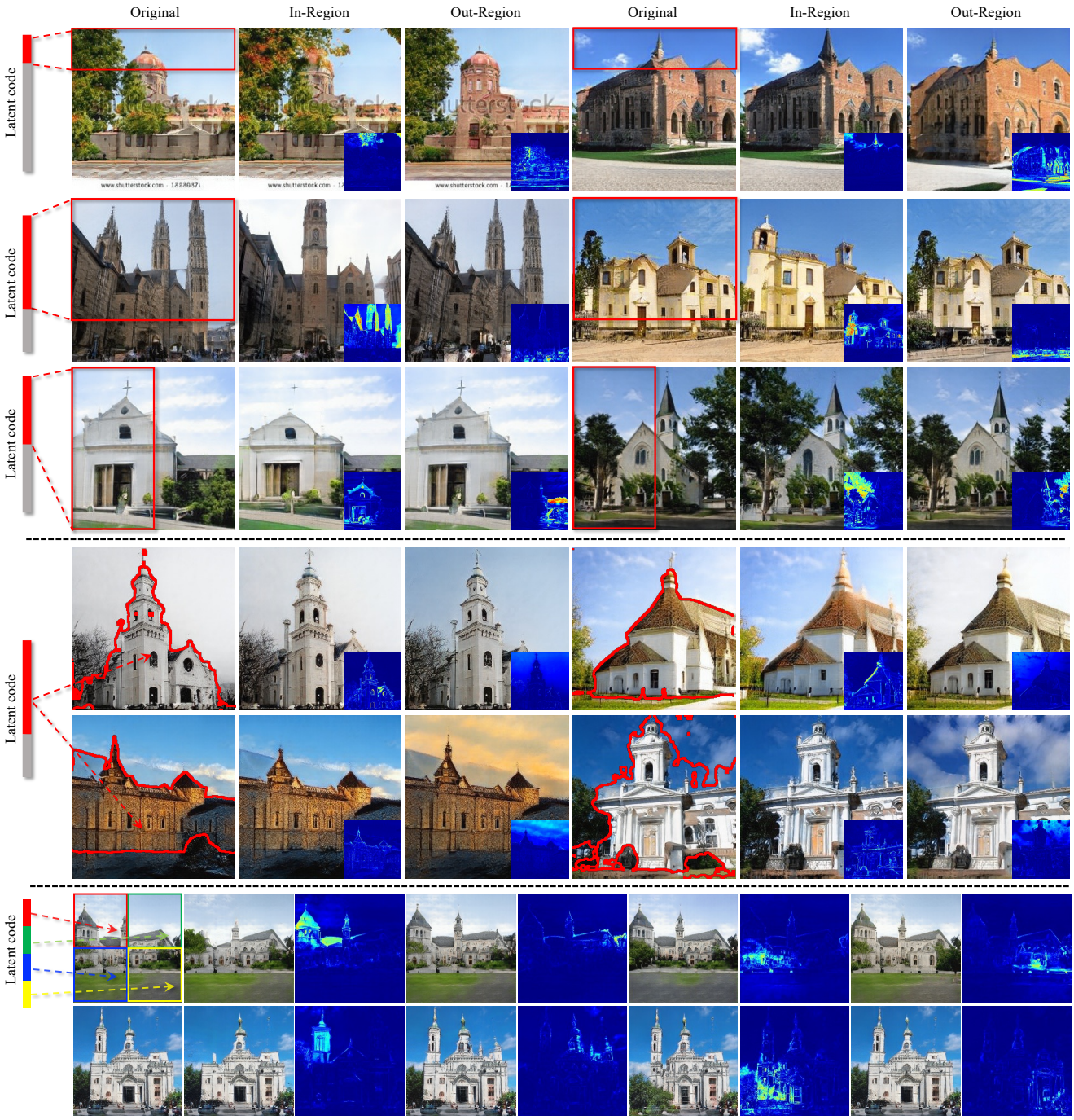

Figure 7. **Linking latents to regions** on church. The regions in the top group are pre-selected before training and shared by all instances, while the regions in the middle group dynamically vary across instances, and the bottom group shows we can link all the image regions to the latent space. Linked latent subspaces and regions are highlighted with different colors, the heatmaps reflect the change of pixel values after in-region resampling and out-region resampling. We can see that LinkGAN can link arbitrary regions no matter whether they are fixed, or dynamically vary, or even for the whole images.

Another way to alleviate this inconsistency we explored is using the discriminator on the perturbed images. Namely, when finetuning, we could involve the discriminator in those perturbed images to hinder the generator from synthesizing those inconsistency perturbed images. We can see from the second column of Fig. 10 and Fig. 11, which

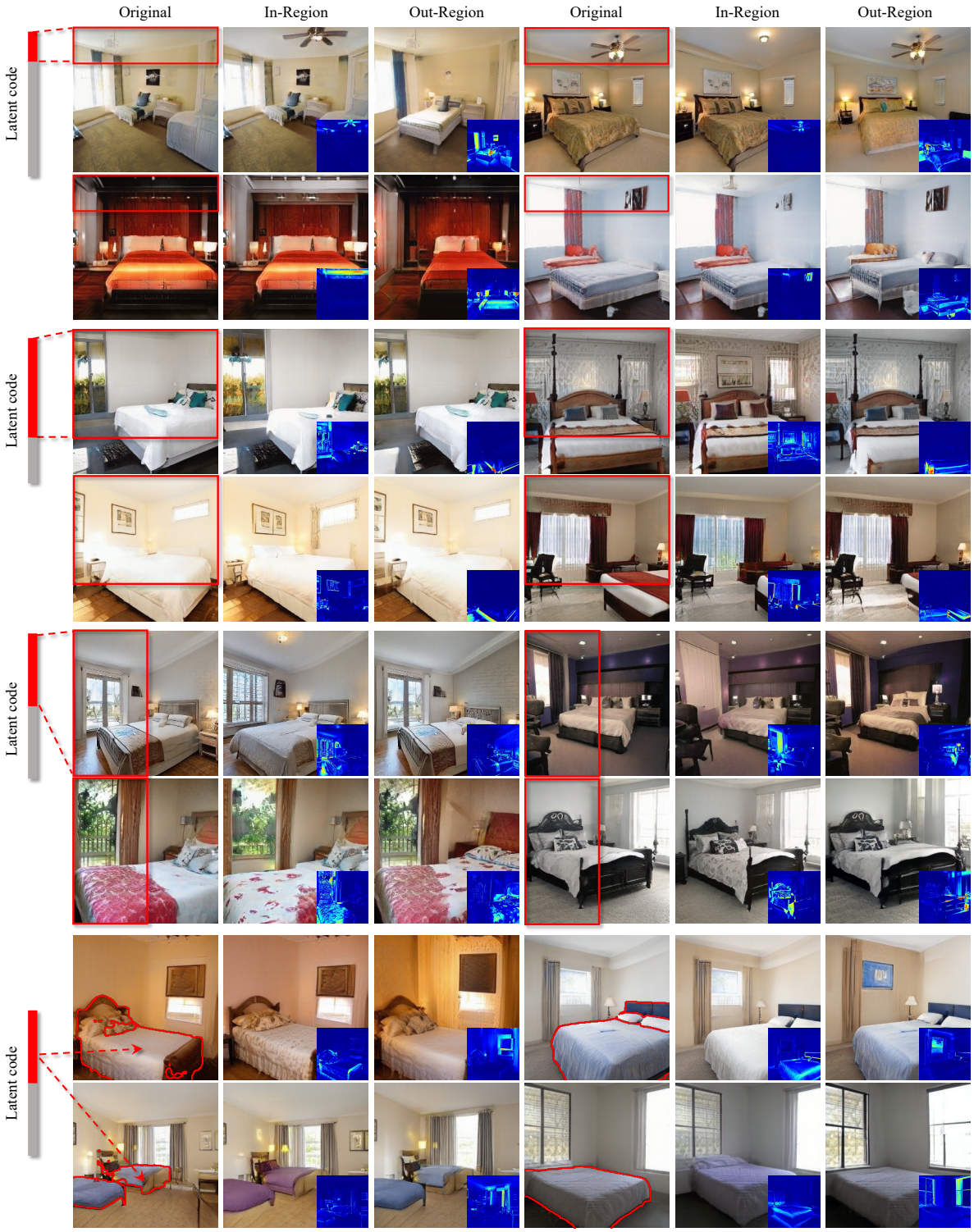

Figure 8. **Linking latents to regions** on bedroom. The regions in the top group are pre-selected before training and shared by all instances, while the regions in the bottom group dynamically vary across instances. Linked latent subspaces and regions are highlighted with red fragments and boxes/contours, the heatmaps reflect the change of pixel values after in-region resampling and out-region resampling. We can see that LinkGAN can link arbitrary regions no matter whether they are fixed or dynamically vary.

give a clear comparison. For instance, in Fig. 10 we can see some edges of the rectangle in the eye region

Figure 9. **Controllability on 3D-aware generative model**, *i.e.*, EG3D [1], under the cases of eyes, left ear, and right ear. We find that LinkGAN is well compatible with 3D-aware image synthesis and allows controlling both the appearance and the underlying geometry.

Table 1. Performance change after introducing our proposed regularizer into 2D and 3D baselines on a single region on different datasets, where the synthesis quality slightly drops but the controllability significantly improves.

Model	StyleGAN [5]													EG3D [1]	
Dataset	FFHQ			AFHQ		Church			Car			Bedroom		FFHQ	
Region	Left	Nose	Mouth	Eyes	Ear	Top	Left	Sky	Bottom	Left	Car	Top	Bottom	Nose	Mouth
w/o Linking	3.98			8.44		3.82			2.95			3.01		4.28	
LinkGAN (Ours)	5.54	5.14	5.11	10.52	9.85	4.27	4.61	4.40	2.88	3.07	2.93	3.49	3.72	4.17	4.21

clearly, even the replaced content is aligned. Instead, we can get much smoother resample results when involving the discriminator. For example, the edge of the rectangle is disappeared, even when the resampled content is not

Table 2. Performance change after introducing our proposed regularizer on StyleGAN [5] on multiple regions on different datasets. For FFHQ, we report links to two, three, and four regions, and for AFHQ and Church, we give the results of linking four regions (*i.e.*, the whole image is linked).

Dataset	FFHQ			AFHQ	Church
w/o Linking	3.98			8.44	3.82
LinkGAN	5.91	6.12	6.28	12.38	4.77

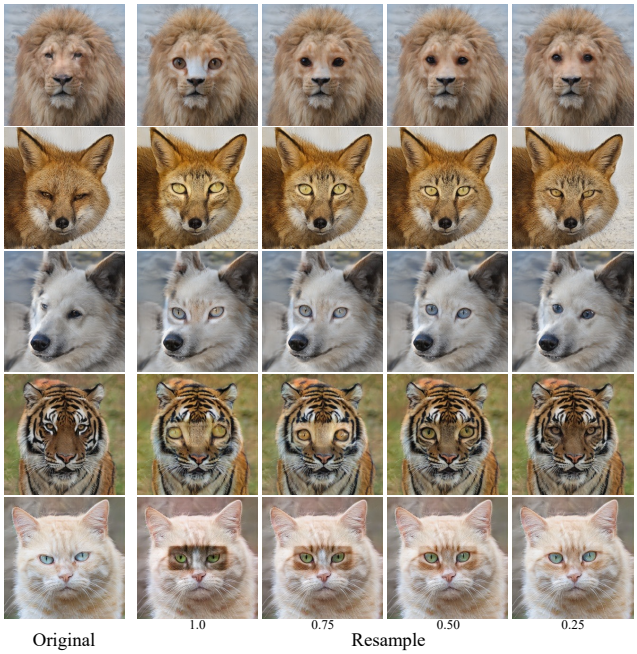

Figure 10. Ablation study on AFHQ [2] using different interpolation strength. The number under each column means the interpolation strength, *e.g.*, 1 means the content in the eyes region is totally from resampled latent code. In contrast, 0.25 means the content combines 0.75 original latent code and 0.25 of the resampled one. During training, the discriminator is not involved in the perturbed images.

well-aligned. Hence, involving a discriminator is another effective way to alleviate inconsistency.

4. Discussion and Conclusion

After linking an arbitrary region to some latent axes with the size of n, any perturbation with randomly sampled n dimension vector on the linked subspace results in the content change only in the linked image region, which can be viewed as a local semantic direction since it only influences the connected region. However, some of the sampled latent vectors can not generate realistic manipulation, and some can (*i.e.*, identical to the inconsistency phenomenon after resampling). Hence, we need to verify whether the randomly sampled vector can produce a meaningful manipulation posteriorly, just like other unsupervised methods [3, 7].

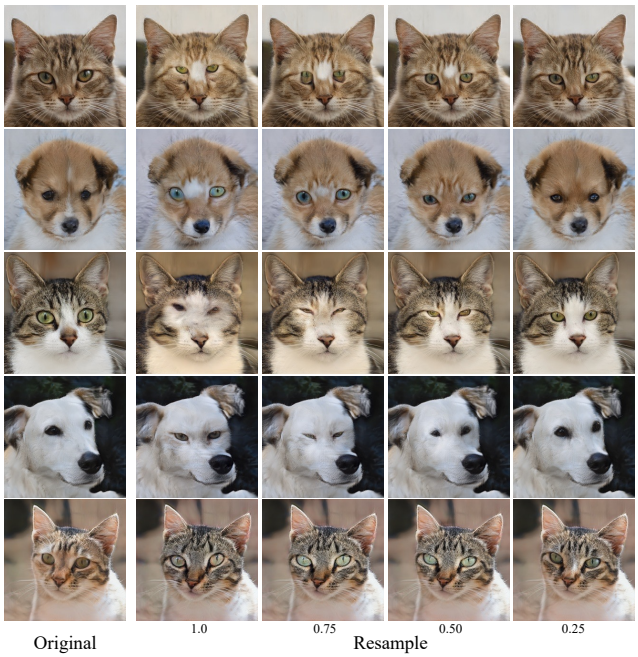

Figure 11. Ablation study on AFHQ [2] using different interpolation strength. The number under each column means the interpolation strength, *e.g.*, 1 means the content in the eyes region is totally from resampled latent code. In contrast, 0.25 means the content combines 0.75 original latent code and 0.25 of the resampled one. During training, the discriminator is involved in the perturbed images.

5. Ethical Considerations

LinkGAN can benefit vision and graphics applications, such as animation and content creation. However, it also poses a threat because generative models can be misused for DeepFake-related applications, *e.g.*, human face editing, and talking head generation. We hope that DeepFake detection algorithms can be developed to avoid such misuse.

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 6, 10

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 4, 6, 11

[3] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *NeurIPS*, 2020. 11

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 4

Table 3. Precision and Recall curve on different datasets trained on StyleGAN [5].

	FFHQ		AFHQ		Church		Car	
Metrics	Prec.↑	Recall↑	Prec.↑	Recall↑	Prec.↑	Recall↑	Prec.↑	Recall↑
w/o Linking	0.849	0.200	0.847	0.126	0.803	0.158	0.826	0.294
LinkGAN	0.864	0.165	0.844	0.021	0.813	0.086	0.838	0.252

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 10, 11, 12

[6] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Int. Conf. Comput. Vis.*, 2021. 1, 2, 3, 4

[7] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 11

[8] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 5

[9] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4

[10] Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen. Region-based semantic factorization in GANs. In *Int. Conf. Mach. Learn.*, 2022. 1, 2, 3, 4