Supplementary Material for On the Difficulty of Unpaired Infrared-to-Visible Video Translation: Fine-Grained Content-Rich Patches Transfer

A. Overview

This supplementary material is organized as follows. § B introduces the detail information about the *Infrared-Adverse* dataset. § C describes the process of our CPTrans via pseudocode. § D illustrates the implementation details including CPTrans and other approaches in the experiments. We further evaluate our CPTrans on the Viper dataset as reported in § E. § F shows more visualization results. Finally, we discuss and analyze our limitation in § G.

B. InfraredCity-Adverse

The unpaired infrared-to-visible video translation task aims to provide clear visible videos for most vision applications under various conditions, especially for adverse weather conditions. However, most existing infrared-related datasets (e.g., IRVI [7], InfraredCity [12]) focus on clear days and nights while lacking data under severe weather conditions, which leads to an incomprehensive evaluation of the translation algorithm. Therefore, to compensate for the lack of current benchmarks, we extend the existing dataset InfraredCity to more challenging adverse weather conditions (rain and snow), dubbed as *InfraredCity-Adverse*[1]. Here we build the dataset with infrared videos, which is captured by the binocular infrared color camera (DTC equipment) on raining and snowing scenes, and visible videos from the InfraredCity dataset.

Rain. Rain is one of the most common adverse condition in the real world. We capture infrared videos in rainy city streets, which contains various complex objects (e.g., buildings, cars). Although infrared sensors are capable of stable imaging in diverse weather conditions, the contents of the videos in rainy scenes are inconsistent with those on clear days. There are noises such as raindrops and fog in the rainy weather, resulting in a data distribution irregular with that of clear weather. Thus, on the proposed rain scene, all competitive methods need to overcome the huge semantic gaps and noises to generate explicit visible videos. Example videos are displayed on our *github*[1]. The total number of infrared frames on the rain scene is *21356*. To reduce the negative effect of the repetition of similar frames, we build the raining scene dataset via the selection by interval sampling. Following [12], we design three data types: *Single*, *Double*, and *Triplet* for the rain scene to be in line with the input requirements of most image/video translation approaches. Detailed in Tab. 4.

Table 4. The structure of InfraredCity-Adverse.

Scene	Data Type	Infrared Frame
Rain	*Single*	4262
	Double	2143
	Triplet	1421
Snow	*Single*	2834
	Double	1419
	Triplet	946

Snow. Similar to rainy days, the environment of snowy days is also different from the one in clear weather, which is caused by the presence of noises such as snow trails and snowflakes. Our snow dataset captures a real snowy day scenario and contains noises mentioned above, which facilitates the evaluation on the performance of translation algorithms in real snowy scenarios. Example videos could be obtained on our *github*[1]. The total number of infrared frames on the snow scene is *14226*. Similarly, we perform the sampling strategy on the snow dataset and design the three data types: *Single*, *Double*, and *Triplet*. Detailed in Tab. 4.

Comparison. IRVI [7] and InfraredCity-Lite [12] are recently released datasets for unpaired infrared-to-visible video translation. As for the IRVI dataset, the infrared data is collected during the day. It has distinct infrared videos without any noise. Similarly, the infrared videos of the InfraredCity-Lite dataset are collected on clear and overcast nights without any environmental noise. Thus, their data distributions differ from those in adverse scenarios. Our released InfraredCity-Adverse includes infrared data from raining and snowing scenes, which can evaluate the method's performance in realistic adverse weather. We hope the provided dataset can encourage the further research on the infrared-related area.

[1]The code and dataset are available at https://github.com/BIT-DA/I2V-Processing

C. Algorithm CPTrans

We illustrate the training process with unpaired data in Alg. 1. When the training is finished, we only need to utilize the generator G for inference.

D. More Implementation Details

CPTrans. We adopt the $resnet\_9blocks$, a ResNet-based model with nine residual blocks, as the backbone for generator G. Additionally, we utilize the patch-wise discriminator D following [12], which is a variant of the PatchGAN. As for the discriminator, we remove its multiscale operation but perform enhancement on content-rich patches. Specifically, the y and $\tilde{y} = G(x)$ will be sent to a pre-trained ViT [4] encoder, and we select the token embeddings on the fifth layer of the encoder, dubbed as T_y^5 and $T_{\tilde{y}}^5$. Notably, we discard the class token embedding at the beginning and obtain $N = w \times h$ token embeddings on T_y^5 and $T_{\tilde{y}}^5$. Each token embedding of T_y^5 has a correspondence with a patch on the y, and it is similar for $T_{\tilde{y}}^5$ and $\tilde{y}$. Then, we send the T_y^5 and $T_{\tilde{y}}^5$ to MLP networks and access prediction scores $\{p_i\}_{i=1}^N$ and $\{\tilde{p}_j\}_{j=1}^N$. Besides, η_{ratio} and λ_{inc} are applied on the content-aware optimization module for controlling the enhancement of attentions to content-rich patches. We conduct extensive research and set $\eta_{ratio} = 40\%, \lambda_{inc} = 8.0$. Additionally, the γ_{stride} in content-aware temporal normalization is set as 20.0 for a distinct shift. λ_1 and λ_2 in Eq. (11) are set to 6.0 and 10.0, respectively. We train CPTrans for 100 epochs with the learning rate of 2×10^6 and the Adam optimizer, using a batch size of 1.

Other Methods. We utilize the official codes of Cycle-GAN[2] [5], CUT[3] [8], F/LSeSim[4] [13], Recycle-GAN[5] [2], UnsupRecycle[6] [10] and I2V-GAN[7] [7]. Similarly, the $resnet\_9blocks$ is applied as the generator for all methods. As described in their original papers, except ROMA, they use the 70×70 PatchGAN, which aims to classify whether 70×70 overlapping image patches are real or fake. ROMA proposes a new patch-wise discriminator with a multiscale operation for different sizes of receptive fields. The batch size of all methods is set as 1. The learning rate values are not the same for different methods, and we follow the settings in the respective papers. Since ROMA [12] and Mocycle-GAN [3] haven't released the code, we implement them ourselves following the training process and implementation from their original paper.

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

[3] https://github.com/taesungp/contrastive-unpaired-translation

[4] https://github.com/lyndonzheng/F-LSeSim

[5] https://github.com/aayushbansal/Recycle-GAN

[6] https://github.com/wangkaihong/Unsup_Recycle_GAN

[7] https://github.com/BIT-DA/I2V-GAN

Table 5. Segmentation score (%) of our CPtrans and other methods for video-to-labels translation on Viper.

Segmentation score (%) of Video-to-Labels							
Metric	**Method**	Day	Sunset	Rain	Snow	Night	All
AC	Cycle-GAN [5]	7.5	6.2	6.1	6.4	4.0	6.6
	MUNIT [13]	13.4	14.7	16.8	15.4	11.7	14.4
	Recycle-GAN [2]	13.6	13.8	11.5	13.6	6.9	13.4
	STC-V2V [3]	12.2	13.5	13.3	13.7	6.7	12.3
	Unsup$_{recycle}$ [10]	16.6	17.2	15.8	16.1	8.3	16.1
	Unsup$_{munit}$ [10]	18.4	19.8	19.1	18.5	10.1	18.3
	Ours	**20.0**	**25.6**	**21.4**	**19.4**	**19.5**	**20.1**
IoU	Cycle-GAN [5]	4.1	3.7	3.3	3.6	1.9	3.8
	MUNIT [13]	7.6	10.2	12.0	11.0	7.4	9.7
	Recycle-GAN [2]	9.0	10.9	8.1	9.6	3.6	9.5
	STC-V2V [3]	8.1	10.6	9.2	10.2	3.9	9.1
	Unsup$_{recycle}$ [10]	11.9	13.5	11.8	12.6	5.1	12.3
	Unsup$_{munit}$ [10]	13.3	15.6	13.9	14.2	6.5	13.7
	Ours	**14.1**	**20.4**	**14.3**	**14.7**	**12.8**	**14.4**
MP	Cycle-GAN [5]	29.7	32.9	34.9	28.6	17.9	28.3
	MUNIT [13]	39.9	66.5	70.5	64.2	50.7	56.4
	Recycle-GAN [2]	46.9	69.8	60.1	56.9	31.9	52.1
	STC-V2V [3]	41.0	67.5	63.0	61.8	36.1	52.5
	Unsup$_{recycle}$ [10]	56.3	76.9	71.8	67.9	48.8	63.2
	Unsup$_{munit}$ [10]	58.7	78.6	**74.1**	**71.1**	44.8	64.3
	Ours	**63.4**	**81.8**	73.0	67.2	**61.5**	**65.7**

E. Evaluation on Viper

Viper [9] consists of virtual world data synthesized from a video game. Each frame is annotated with pixel-level semantic labels. According to different weather conditions, it is divided into five different scenes: day, sunset, rain, snow, and night. Following [2], we reduce the resolution of each frame from 1920×1080 to 256×256 and take 57 video clips for training and the other 22 clips for testing.

To further measure the translation capability of our CPTrans, we evaluate it on the Viper dataset, which also requires every pixel of the generated results to be correct. From Tab. 5, we can see that our method outperforms most other techniques. Especially in the night scene, we achieve improvements of 93.1% on AC, 96.9% on IoU, and 37.3% on MP, respectively, compared with Unsup$_{munit}$ [10].

F. More Visualization Results

F.1. Long-tail Effect and Content-rich Patches

We illustrate the visualization of pixel category distribution on dataset InfraredCity [12], which is the largest dataset for unpaired infrared-to-visible video translation, as shown in Fig. 9. It indicates that real-world training data usually exhibits long-tailed distribution.

Since gradients from different patches tend to vary [1, 6] in the optimization process, plus the existing long-tail effect on the real world, the optimization can be prejudiced against content-rich patches (i.e., minority pixels). We visualize the most deviated parts in Fig. 10, and it confirms the effectiveness of our CO module in optimizing content-rich patches. Our CPTrans encourages optimizing the model along the

Algorithm 1 Training process of CPTrans

Input: Source domain $\{x\}$; Target domain$\{y\}$; Models G, D; Feature extractor F; Max iteration: N_{iter}
Output: Translated results $\{G(x)\}$ ▷ $G(x)$ looks similar to the target domain while having the same content as x.

for $n = 1$ to N_{iter} **do**
  Sample x from $\{x\}$, y from $\{y\}$; ▷ x, y are unpaired data, and are randomly selected.
  Generate $\tilde{y} = G(x)$ via generator G;
  Obtain token embeddings T_s and T_t from x and $\tilde{y}$ via feature extractor F, respectively;
  Compute $\mathcal{L}_{cs}$ through Eq. (1);
  Compute the prediction scores $\{p_i\}_{i=1}^N, \{\tilde{p}_j\}_{j=1}^N$ from $y, \tilde{y}$, through the discriminator D, respectively;
  Obtain gradients $\{\nabla_{\theta_D} \log p_i\}_{i=1}^N, \{\nabla_{\theta_D} \log(1-\tilde{p}_j)\}_{j=1}^N$ via calculating $\nabla_{\theta_D}\mathcal{L}_{adv}^{patch}$ through Eq. (4);
  Calculate cosine similarities $\{\delta_i\}_{i=1}^N, \{\tilde{\delta}_i\}_{i=1}^N$ according to Eq. (5);
  Get the weight maps $\{w_i\}_{i=1}^N, \{\tilde{w}_j\}_{j=1}^N$ through Eq. (6);
  Utilize weight maps to calculate $\mathcal{L}_{co\text{-}adv}^{patch}$ through Eq. (7);
  Randomly sample z from a standard Gaussian distribution $\mathcal{N}(0, 1)$;
  Generate content-aware optical flow $F_{content}$ with z and $\tilde{w}$ through Eq. (9);
  Calculate $\mathcal{L}_{ctn}$ through Eq. (10);
  Compute $\mathcal{L} = \mathcal{L}_{co\text{-}adv}^{patch} + \lambda_1 \cdot \mathcal{L}_{cs} + \lambda_2 \cdot \mathcal{L}_{ctn}$ with SGD optimization;
end for

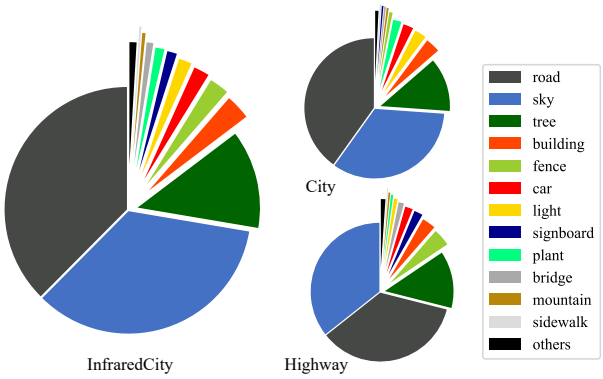

Figure 9. Left: Visualization of pixel category distribution on dataset InfraredCity [7]. We conduct semantic segmentation via a pre-trained SegFormer [11] on all visible video frames of InfraredCity and predict all pixels according to the predefined categories in ADE20K [14]. Right: We additionally illustrate the pixel category distributions of specific scenarios (i.e., City and Highway).

gradients of content-rich patches that are no longer the most deviated parts.

F.2. Qualitative Comparison

We display more qualitative comparisons in Fig. 11 with other methods. Our CPTrans yields visually pleasant outcomes with more details.

F.3. Visual Examples

We display more video translation results on our *github*[1], including the outputs of translation between infrared and visible videos, the results of translation between videos and labels, object detection videos, and video fusions.

G. Limitation

One potential limitation of our translation methods, which is shared with the existing SOTA, is the lack of diversity in the appearance of the generated videos. Due to the thermal imaging mechanism of infrared sensors, all video frames are covered by the gray-style appearance, and the data diversity mainly comes from the structure difference. For example, our CPTrans provides black looks for most generated cars while red appearance for most translated trucks. Similar structures in the generated results always have a limited appearance. How to make the generated results as diverse as real visible videos will be our further exploration in future research.

References

[1] Aleksandar Armacki, Dragana Bajovic, Dusan Jakovetic, and Soummya Kar. Gradient based clustering. In *ICML*, pages 929–947, 2022. 2

[2] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *ECCV*, pages 122–138, 2018. 2

[3] Y chen, Y Pan, T Yao, X Tian, and T Mei. Mocycle-gan: Unpaired video-to-video translation. In *ACM MM*, pages 647–655, 2019. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[5] Zhu Jun-Yan, Park Taesung, Isola Phillip, and Efros Alexei A. Unpaired image-to-image translation using cycle-

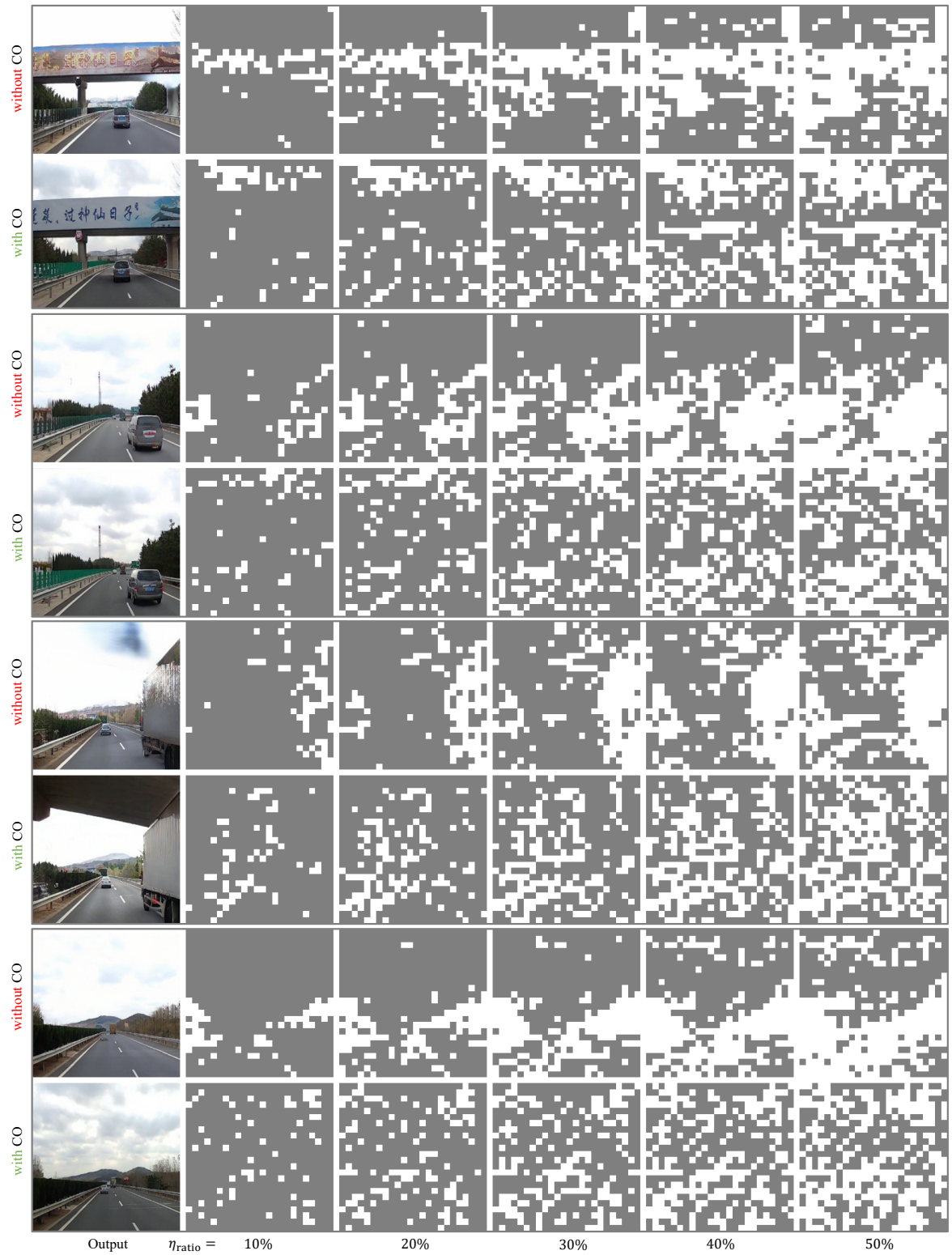

Figure 10. Visualization for the influence of our Content-aware Optimization for gradients. We select the **most deviated** η_{ratio} patches to display based on the cosine similarity to the final gradient. The visualization results indicate that the most deviated parts are usually the content-rich patches when trained without our CO module. Significantly, the bottom figure shows that the optimization is controlled by the major pixels (i.e., sky pixels). In contrast, our CPTrans encourages optimizing the model along the gradients of content-rich patches that are no longer the most deviated parts.

Figure 11. Qualitative comparisons with different methods. Our outputs show cleaner and more visual details compared to others on diverse scenes, especially the adverse ones (the bottom three scenes).

consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2

[6] Jongkyeong Kang and Seung Jun Shin. A gradient-based variable selection for binary classification in reproducing kernel hilbert space. *CoRR*, abs/2109.14282, 2021. 2

[7] Shuang Li, Bingfeng Han, Zhenjie Yu, Chi Harold Liu, Kai Chen, and Shuigen Wang. I2V-GAN: unpaired infrared-to-visible video translation. In *ACM MM*, pages 3061–3069,

2021. 1, 2, 3

[8] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, pages 319–345, 2020. 2

[9] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, pages 2232–2241, 2017. 2

[10] Kaihong Wang, Kumar Akash, and Teruhisa Misu. Learning temporally and semantically consistent unpaired video-to-video translation through pseudo-supervision from synthetic optical flow. In *AAAI*, pages 2477–2486, 2022. 2

[11] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, pages 12077–12090, 2021. 3

[12] Zhenjie Yu, Kai Chen, Shuang Li, Bingfeng Han, Chi Harold Liu, and Shuigen Wang. ROMA: cross-domain region similarity matching for unpaired nighttime infrared to daytime visible video translation. In *ACM MM*, pages 5294–5302, 2022. 1, 2

[13] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *CVPR*, pages 16407–16417, 2021. 2

[14] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, pages 5122–5130, 2017. 3