# AMT: All-Pairs Multi-Field Transforms for Efficient Frame Interpolation
## – *Supplementary Materials* –

Zhen Li* Zuo-Liang Zhu* Ling-Hao Han Qibin Hou Chun-Le Guo† Ming-Ming Cheng
VCIP, CS, Nankai University
{zhenli1031, nkuzhuzl}@gmail.com, lhhan@mail.nankai.edu.cn
{houqb, guochunle, cmm}@nankai.edu.cn

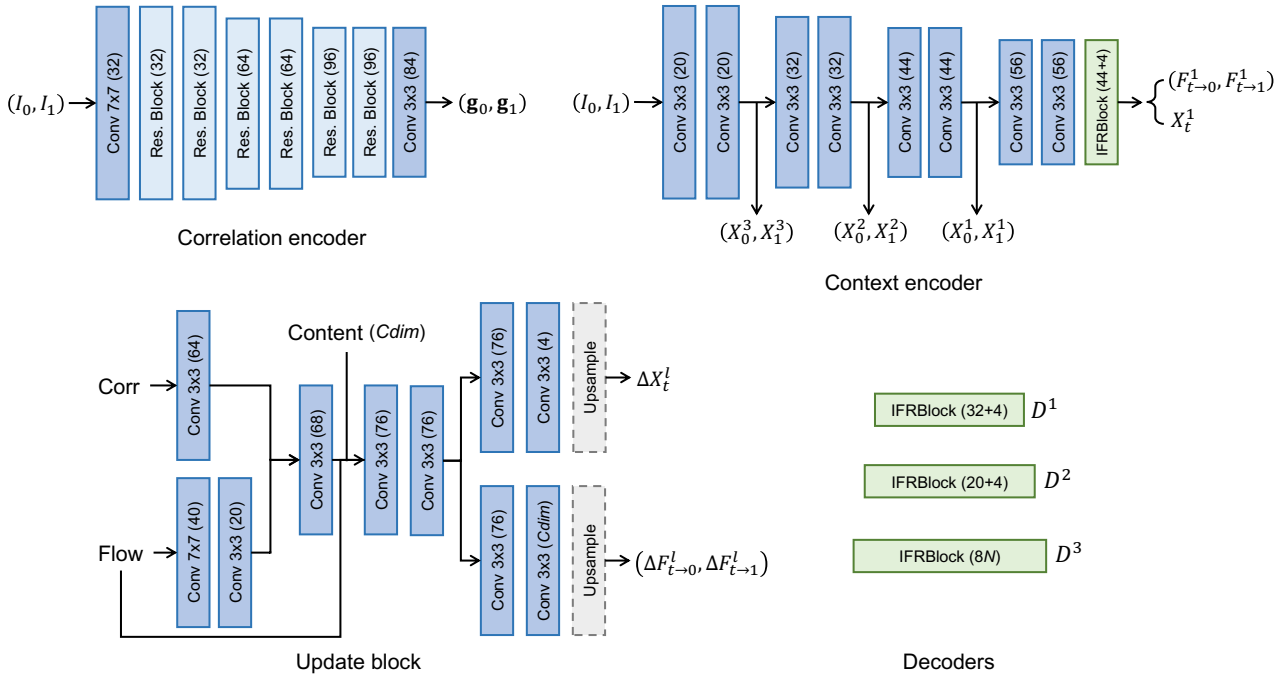



Figure 1. Architecture details of the AMT-S. The number in parentheses denotes the output channels. $N$ represents the number of output groups. IFRBlock denotes the decoder proposed in IFRNet [9].

## 1. Architecture Details

We build three models with different sizes, termed AMT-S, AMT-L, and AMT-G. For reproducibility, the architecture details of them are shown in Fig. 1, Fig. 4, and Fig. 5, respectively. We employ standard residual blocks [3] and instance normalization [21] in the correlation encoder. The lookup radius is set to 3. For each update block, a bilinear upsampling layer follows each head on upper levels (*i.e.,* $l$>1). The IFRBlock represents the decoder proposed in IFRNet [9], which jointly estimates the bilateral flows and the intermediate feature. To further improve performance, we upsample the correlation feature in the case of AMT-G to align its spatial resolution with the current interpolated feature, facilitating updates in the high-resolution space. The code is available at https://github.com/MCG-NKU/AMT.

## 2. Multi-Frame Interpolation

For the multi-frame setting, we use GoPro dataset [14] for training and evaluate our model on the test partition of GoPro dataset [14] and Adobe240 dataset [18]. Here, we aim at $8\times$ interpolation, synthesizing 7 intermediate frames with two input frames. The other training settings and loss functions are consistent with those in our main paper. Following recent frame interpolation works [6, 9], we inject a temporal embedding vector into the network for $8\times$ interpolation. The elements in this vector are all set to $t$ according to the current time step, where $t \in \{1/8, 2/8, ..., 7/8\}$.


Figure 2. Qualitative results of our AMT-S and IFRNet-B [9] on Adobe240 [18]. The time steps are $1/4$ and $1/2$ from top to bottom.

| Method | GoPro [14] | | Adobe240 [18] | |
|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM |
| DVF [12] | 21.94 | 0.776 | 28.23 | 0.896 |
| SuperSloMo [7] | 28.52 | 0.891 | 30.66 | 0.931 |
| DAIN [1] | 29.00 | 0.910 | 29.50 | 0.910 |
| IFRNet-B [9] | 29.97 | 0.922 | 31.93 | 0.936 |
| AMT-S (Ours) | **30.20** | **0.927** | **32.04** | **0.938** |

Table 1. Quantitative comparison for $8\times$ interpolation.

We compare our AMT-S with DVF [12], SuperSloMo [7], DAIN [1], and IFRNet-B [9]. The results of $8\times$ interpolation are shown in Tab. 1. Our method obtains the best PSNR and SSIM results on both evaluation datasets, indicating the effectiveness of the proposed AMT for the task of multi-frame interpolation. Fig. 2 and Fig. 6 visually compare our method and IFRNet-B on the Adobe240 dataset. Here, we visualize the cases for $1/4$ and $1/2$ time steps. It can be seen that our method can generate more temporally consistent results with fewer artifacts and more clear edges.

## 3. Limitation

Although our method has shown remarkable performance, the 4D correlation volume computed from all pairs of pixels makes it hard to adapt to very high-resolution inputs under a resource-constrained environment. This is because the computational complexity of constructing correlation volumes is quadratic to the image resolution. The possible ways to alleviate this problem include computing each correlation value only when it is looked up [19] or factorizing the 4D correlation volume to two 3D correlation volumes [22].

## 4. Discussions with RAFT

Teed and Deng [19] proposed RAFT, which iteratively performs lookups on multi-scale 4D correlation volumes for updating flow fields. Given its impressive results, current state-of-the-art flow estimation methods [5, 8, 22, 23, 25] all derive from such architecture design. Besides, it inspires the development of stereo matching [11] and scene flow [20]. However, the RAFT-like design paradigm is not well investigated in frame interpolation.

To better model large motions for frame interpolation, we build AMT based on RAFT. However, AMT involves many novel and task-specific designs beyond it. To better illustrate our model, we detail the differences between our AMT and RAFT from the following perspectives:

**Volume Design:** RAFT constructs a unidirectional correlation volume because it only needs to predict the optical flow along one direction. For frame interpolation, we hope to model the dense correspondences on both directions for updating bilateral flows. We thus construct bidirectional correlation volumes. We have verified the effectiveness of the bidirectional correlation volumes in Tab. 2a of the main paper.

**Context Encoder:** In RAFT, the context encoder extracts the content feature only from the first input frame. Because of the characteristics of frame interpolation, in our AMT, the context encoder takes the image pair as input. It outputs the initial intermediate feature, the initial bilateral flows, and the pyramid features from the input pair. This design is

| Case | Vimeo90K [24] | SNU-FILM [2] | | FLOPs (G) |
|---|---|---|---|---|
| | | Hard | Extreme | |
| Single-scale Pred. | 35.94 | 30.52 | 25.26 | 124 |
| ConvGRU | 35.99 | 30.58 | 25.27 | 132 |
| Tied Weights | 35.93 | 30.56 | 25.22 | 121 |
| Convex Upsampling | 35.99 | 30.56 | 25.28 | 123 |
| Original Model | 35.97 | 30.60 | 25.30 | 121 |

Table 2. Investigation on RAFT-like [19] designs. The default setting is marked in gray.

also inspired by recent one-stage frame interpolation methods [9, 17].

**Correlation Lookup:** The lookup operation can be directly performed in RAFT for the identical coordinate system between the correlation volume and predicted flow field. To solve the coordinate mismatch issue caused by the invisible frame, we propose to scale the bilateral flows before the lookup operation. Besides, we retrieve bidirectional correlations instead of the unidirectional ones in RAFT. We use the initial bilateral flows $(F^1_{t\to 0}, F^1_{t\to 1})$ as the initial starting point, while RAFT uses zero instead. The lookup strategy is investigated in Tab.2b of the main paper.

**Predict and Update Manner:** While RAFT predicts and updates the flow prediction at a single resolution, we predict and update the bilateral flows in a coarse-to-fine manner. We also provide a variant of our AMT to verify the design, which only predicts the flow fields at a single resolution before feeding into the last decoder. Tab. 2 shows that this variant performs worse than the original one. This indicates that predicting multi-scale flows are important for frame interpolation. Besides, we also investigate the effectiveness of the cross-scale update in Tab. 2d of the main paper.

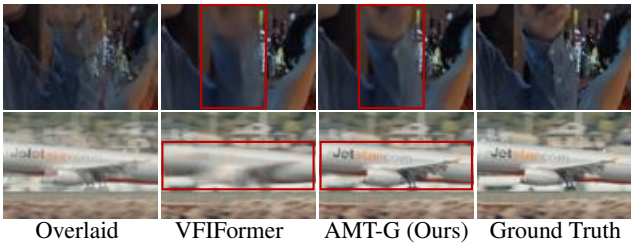
**Update Block:** In the design of the update block, our AMT differs from RAFT in five aspects: 1) While RAFT regards the feature extracted from the visible frame as the content guidance, we use the interpolated intermediate feature representing the invisible frame instead. 2) RAFT only has one head in update block for regressing a flow residual, while we have two heads for jointly predicting content and flow residuals. The two aspects mentioned above have been discussed in Tab. 2c of the main paper. 3) We stack two convolutional layers instead of a cumbersome ConvGRU unit in RAFT to handle the content and motion features. We also investigate a variant that equips with a ConvGRU unit in each update block. As shown in Tab. 2, this variant shows a comparable performance in contrast to the original one, but it has more computational costs. We thus choose to stack two convolutional layers for efficiency. 4) The weights of update blocks are not shared across levels in our AMT. However, weight tying is beneficial to RAFT. Tab. 2 demonstrates that the model with untied weights performs better than that with tied weights. 5) We employ bilinear upsampling instead of convex sampling in RAFT for upscaling the flow fields. As shown in Tab. 2, the two upsampling operators have similar performance, but convex upsampling will incur more computation costs. Thus, we choose the bilinear upsampling in our AMT.

**Final Objective:** RAFT is designed for flow estimation and is optimized only with flow regression loss. However, our AMT is introduced for frame interpolation and is supervised with both task-oriented flow distillation loss and distortion-oriented content losses. We need to consider not only the fidelity of estimated flows but also the diversity for meeting the requirement of task-oriented flows. We thus output multiple flow pairs rather than a single flow field in RAFT. Besides, occlusion reasoning and residual hallucination also need to be considered for faithful content generation.

## 5. Discussion about Multi-Field Refinement

Some works [4, 15, 16] also attempt to predict multiple flow pairs for preparing intermediate content candidates. Specifically, BMBC [15] predicts six bilateral motions through the bilateral motion network and optical flow approximation. ABME [16] generates four bilateral flow fields based on asymmetric motion assumption. After obtaining warped candidate frames and context features, the two works rely on a dynamic filter and even a cumbersome synthesis network to generate the final intermediate frame. Thus, they are inefficient for practical usage. In contrast, our AMT is more efficient, as shown in Tab. 1 of the main paper. We generate multiple flow fields in a single forward pass instead of multiple inference steps in BMBC and ABME. Besides, we obtain the intermediate candidates only in the image domain rather than the feature domain and stack two lightweight convolutional layers for fusing these candidates.

M2M-VFI [4] is most relevant to our multi-field refinement. It also generates multiple flows in one step and prepares warped candidates in the image domain. However, there are five key differences between our multi-field refinement and M2M-VFI. First, our method generates the candidate frames by backward warping rather than forward warping in M2M-VFI. Second, while M2M-VFI predicts multiple flows to overcome the hole issue and artifacts in overlapped regions caused by forward warping, we aim to alleviate the ambiguity issue in the occluded areas and motion boundaries by enhancing the diversity of flows. Third, M2M-VFI needs to estimate bidirectional flows first through an off-the-shelf optical flow estimator and then predict multiple bilateral flows through a motion refinement network. On the contrary, we directly estimate multiple bilateral flows in a one-stage network. In this network, we



Overlaid VFIFormer AMT-G (Ours) Ground Truth

Figure 3. Qualitative comparison between AMT-G with VFIFormer. Our method recovers more clear structure and edges.

first estimate one pair of bilateral flows at the coarse scale and then derive multiple groups of fine-grained bilateral flows from the coarse flow pairs. Fourth, M2M-VFI jointly estimates two reliability maps together with all pairs of bilateral flows, which can be further used to fuse the overlapping pixels caused by forward warping. As shown in Eqn. (5) of the main paper, we estimate not only an occlusion mask but a residual content for cooperating with each pair of bilateral flows. The residual content is used to compensate for the unreliable details after warping. This design has been investigated in Tab. 2e of the main paper. Fifth, we stack two convolutional layers to adaptively merge candidate frames, while M2M-VFI normalizes the sum of all candidate frames through a pre-computed weighting map.

## 6. More Visual Results

In this section, we provide additional visual results on two benchmark datasets, including Vimeo90K [24] and SNU-FILM [2], to further show the superiority of the proposed AMT. The comparison methods include CAIN [2], AdaCoF [10], ABME [16], RIFE [6], IFRNet(-B/-L) [9], and VFIFormer [13]. For a fair comparison, we also divide these methods into two groups according to the computational cost. As shown in Fig. 3, 7-12, our AMT synthesizes the object with large motions more faithfully and generates plausible textures with fewer artifacts.

## 7. Broader Impact

As presented in this paper, our AMT can synthesize faithful non-existent frames between two visible frames. Given its reliable synthesis results, our method may be abused to forge or tamper with videos.

## References


[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, 2019. 2

[2] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, 2020. 3, 4, 8, 9, 10, 11, 12

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[4] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, 2022. 3

[5] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *ECCV*, 2022. 2

[6] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022. 1, 4, 7, 9, 11

[7] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018. 2

[8] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 2

[9] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

[10] Hyeongmin Lee, Taeoh Kim, Tae young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, 2020. 4, 7, 9, 11

[11] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *3DV*, 2021. 2

[12] Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 2

[13] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *CVPR*, 2022. 4

[14] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1, 2

[15] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *ECCV*, 2020. 3

[16] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *CVPR*, 2021. 3, 4, 8, 10, 12

[17] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *ECCV*, 2022. 3

[18] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017. 1, 2, 6

[19] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 3

[20] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *CVPR*, 2021. 2

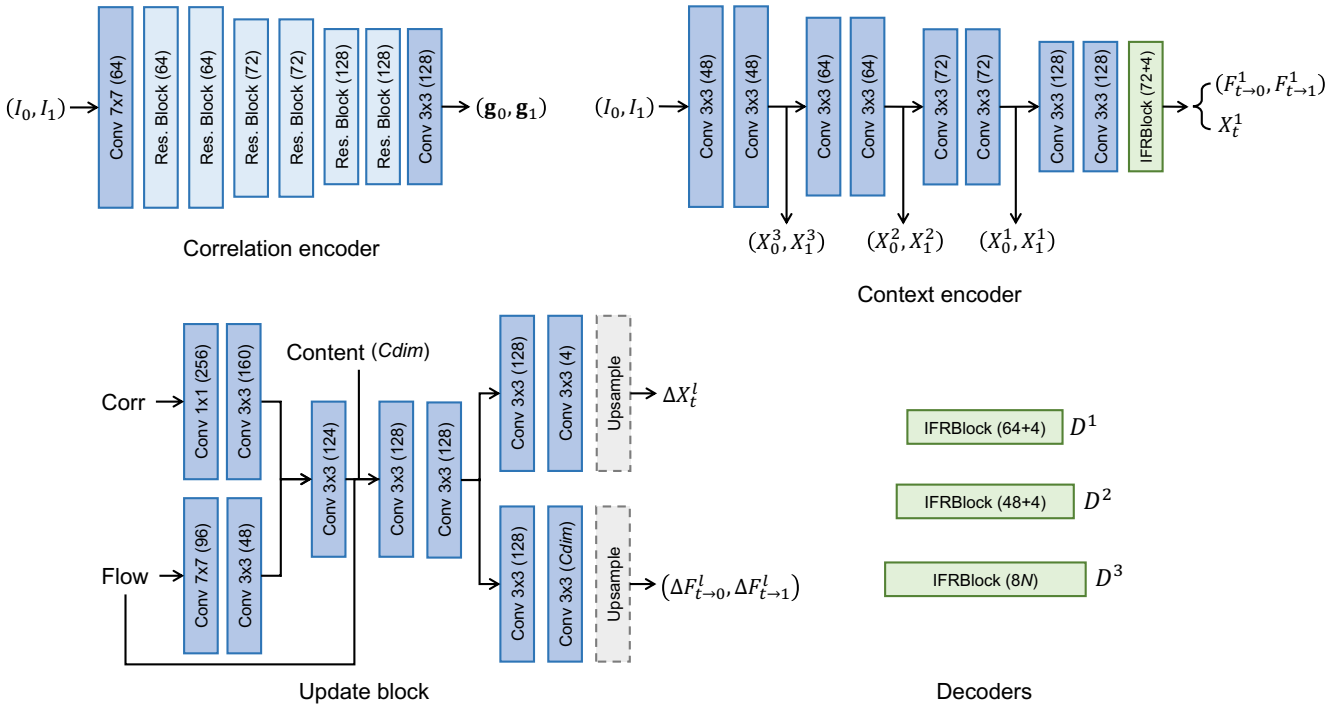



Figure 4. Architecture details of the AMT-L. The number in parentheses denotes the output channels. $N$ represents the number of output groups. IFRBlock denotes the decoder proposed in IFRNet [9].

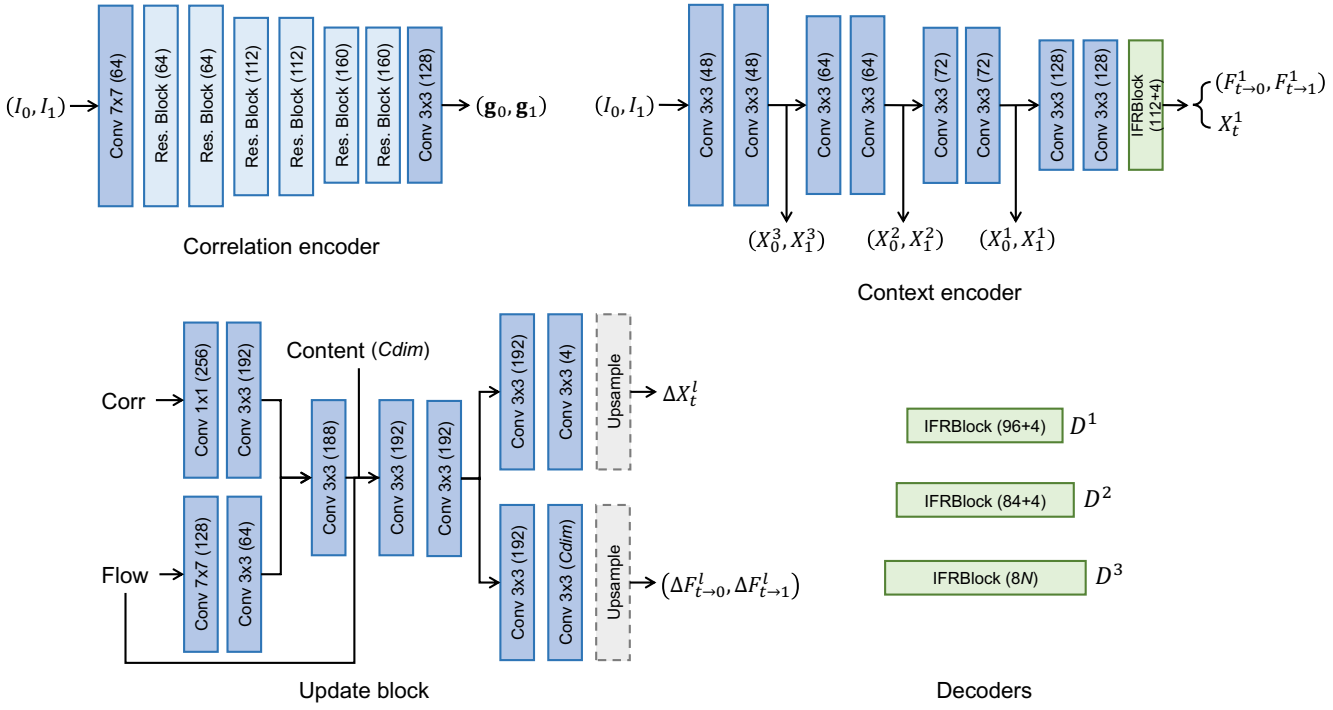



Figure 5. Architecture details of the AMT-G. The number in parentheses denotes the output channels. $N$ represents the number of output groups. IFRBlock denotes the decoder proposed in IFRNet [9].

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast styliza-

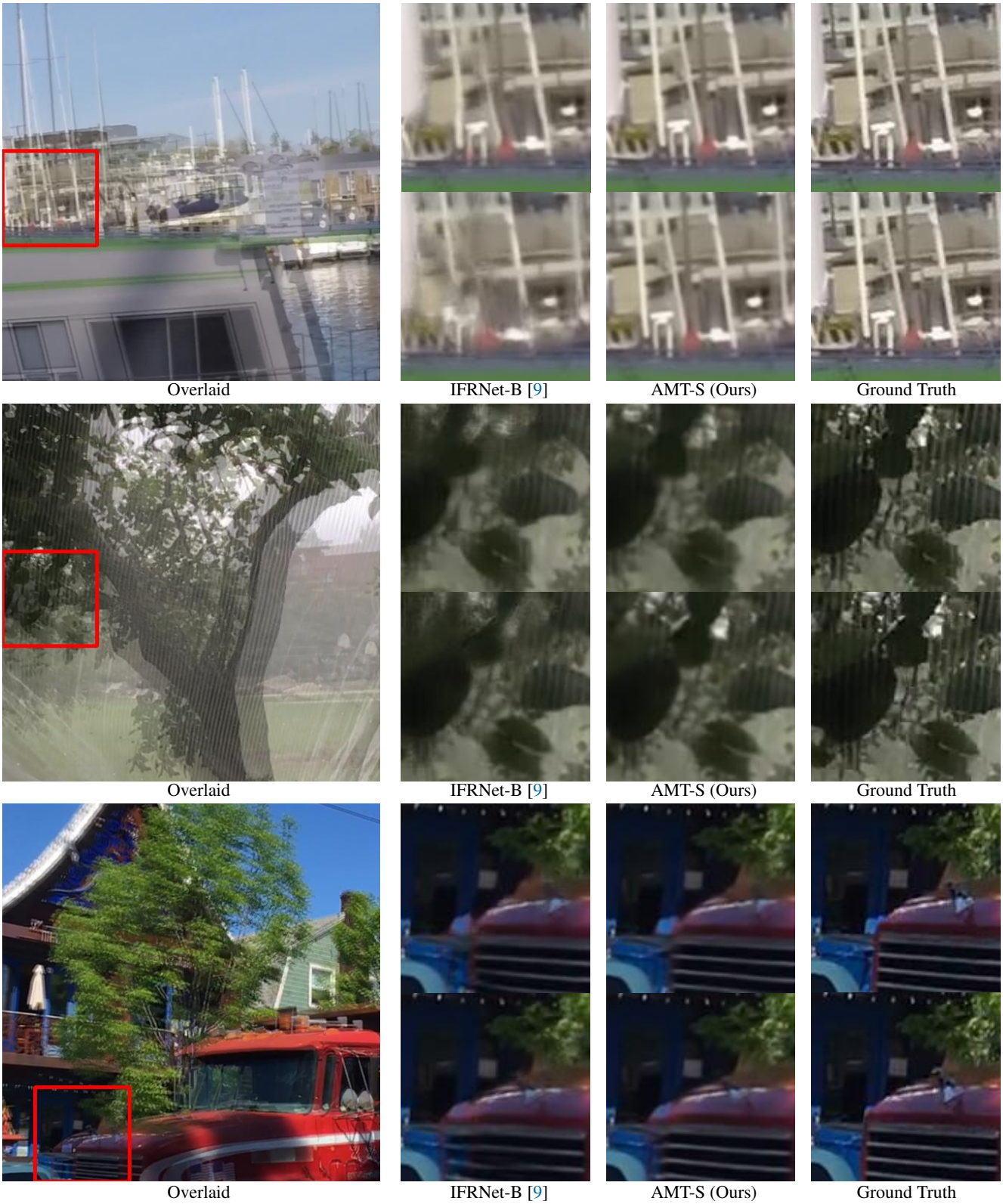



Figure 6. Qualitative results of AMT-S and IFRNet-B [9] on Adobe240 [18]. The time steps are $1/4$ and $1/2$ from top to bottom.

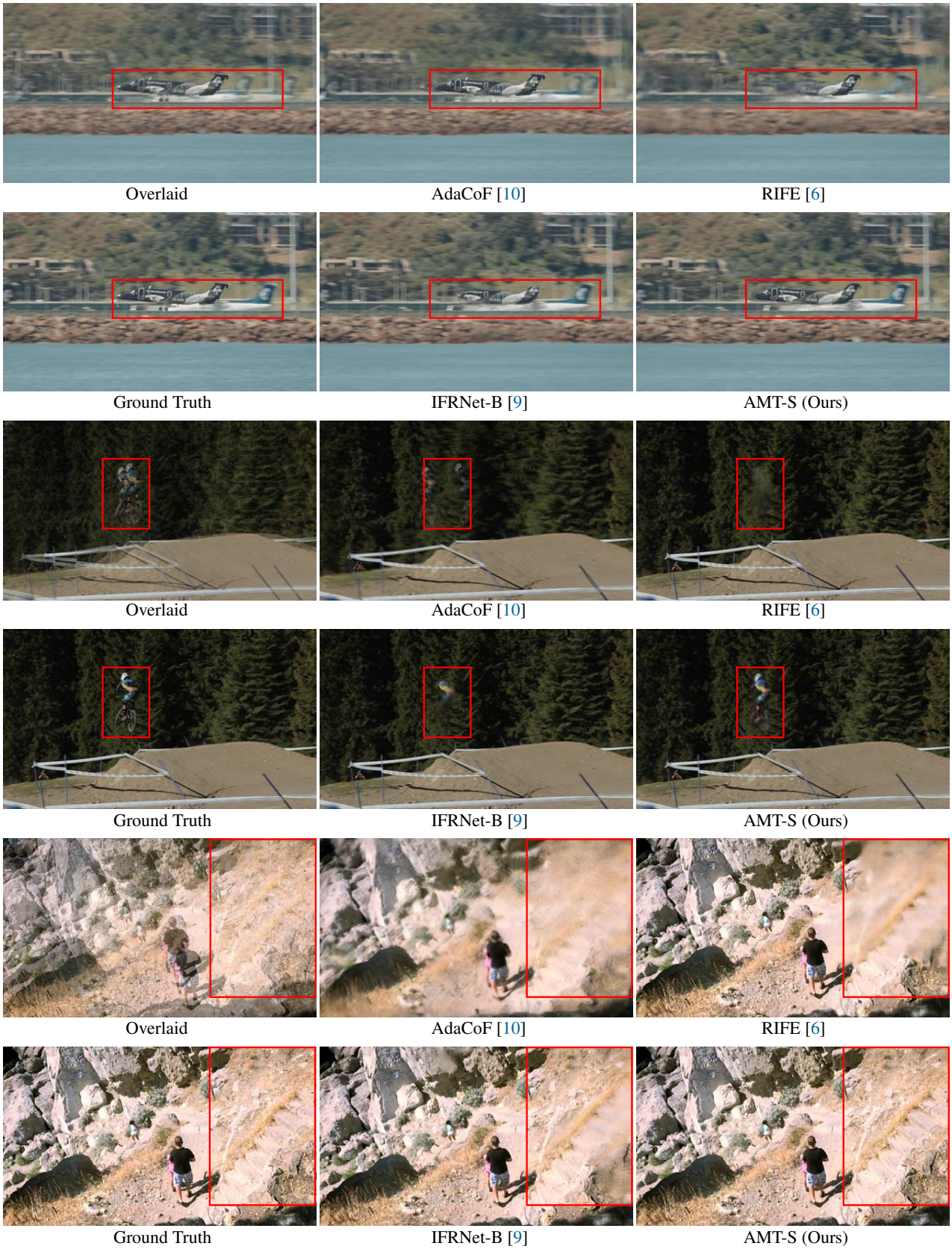



Figure 7. Visual comparison for the methods with low computational complexity on Vimeo90K dataset [24]

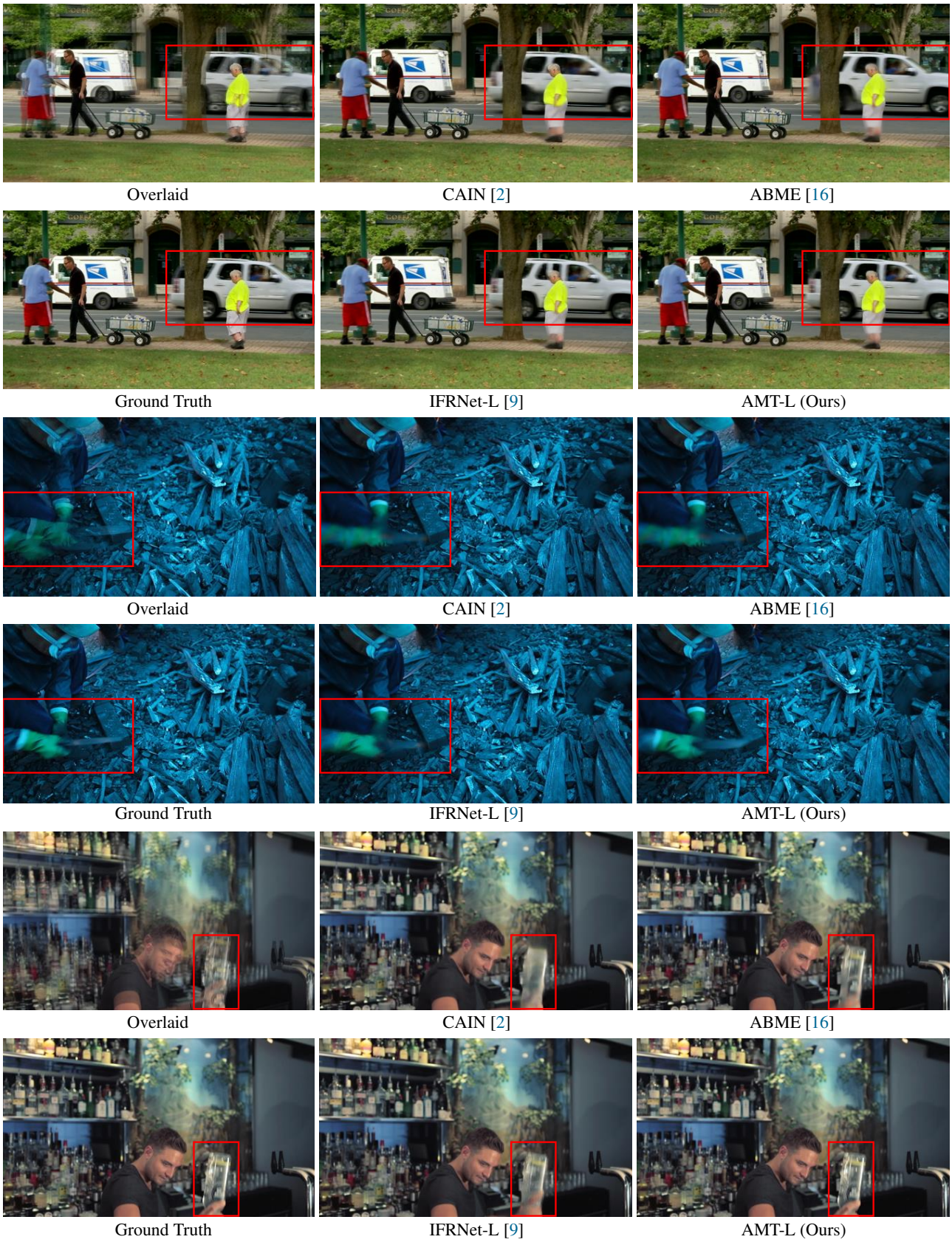



Figure 8. Visual comparison for the methods with relatively high computational complexity on Vimeo90K dataset [24].

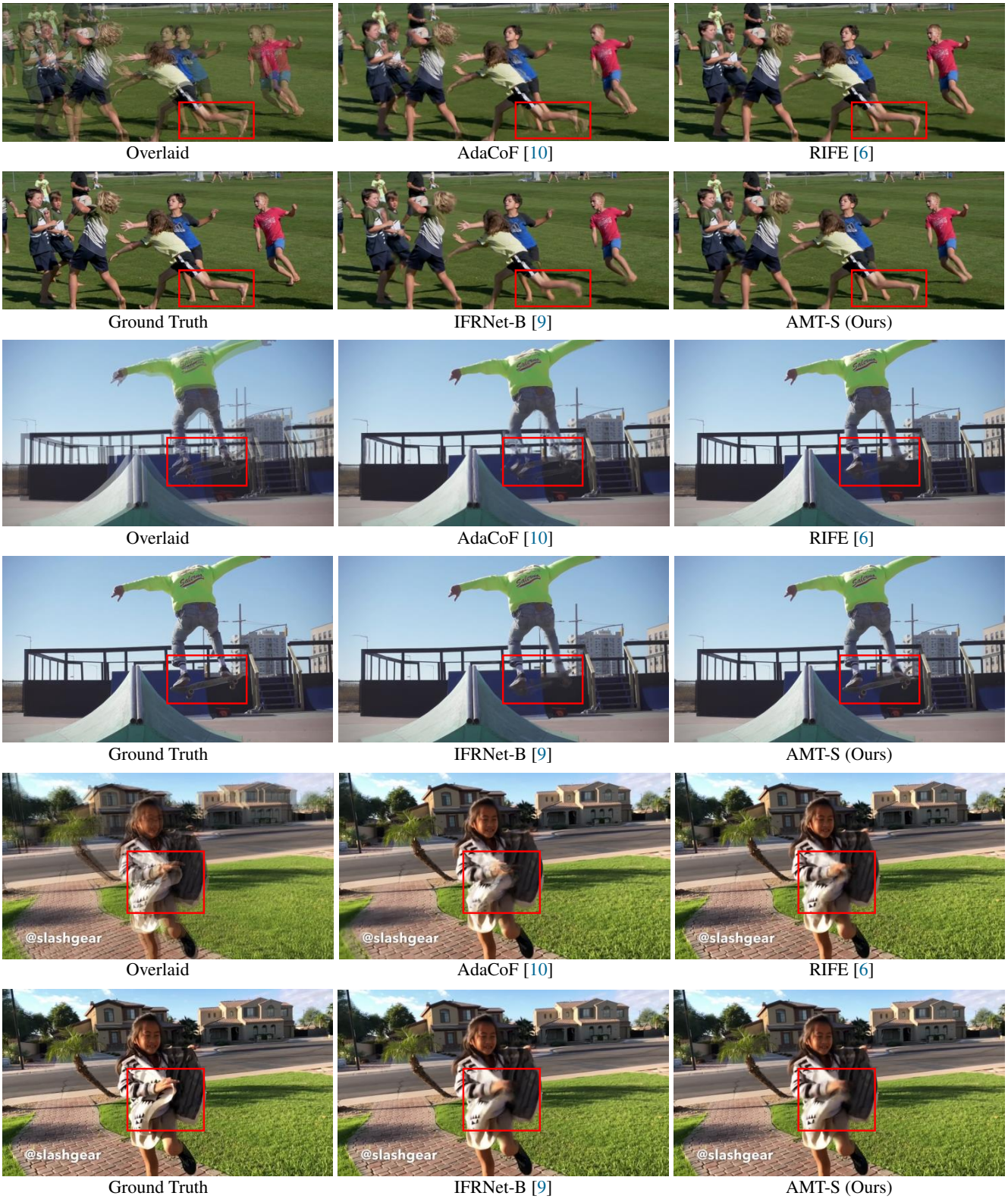



Figure 9. Visual comparison for the methods with low computational complexity on the Hard partition in SNU-FILM dataset [2].

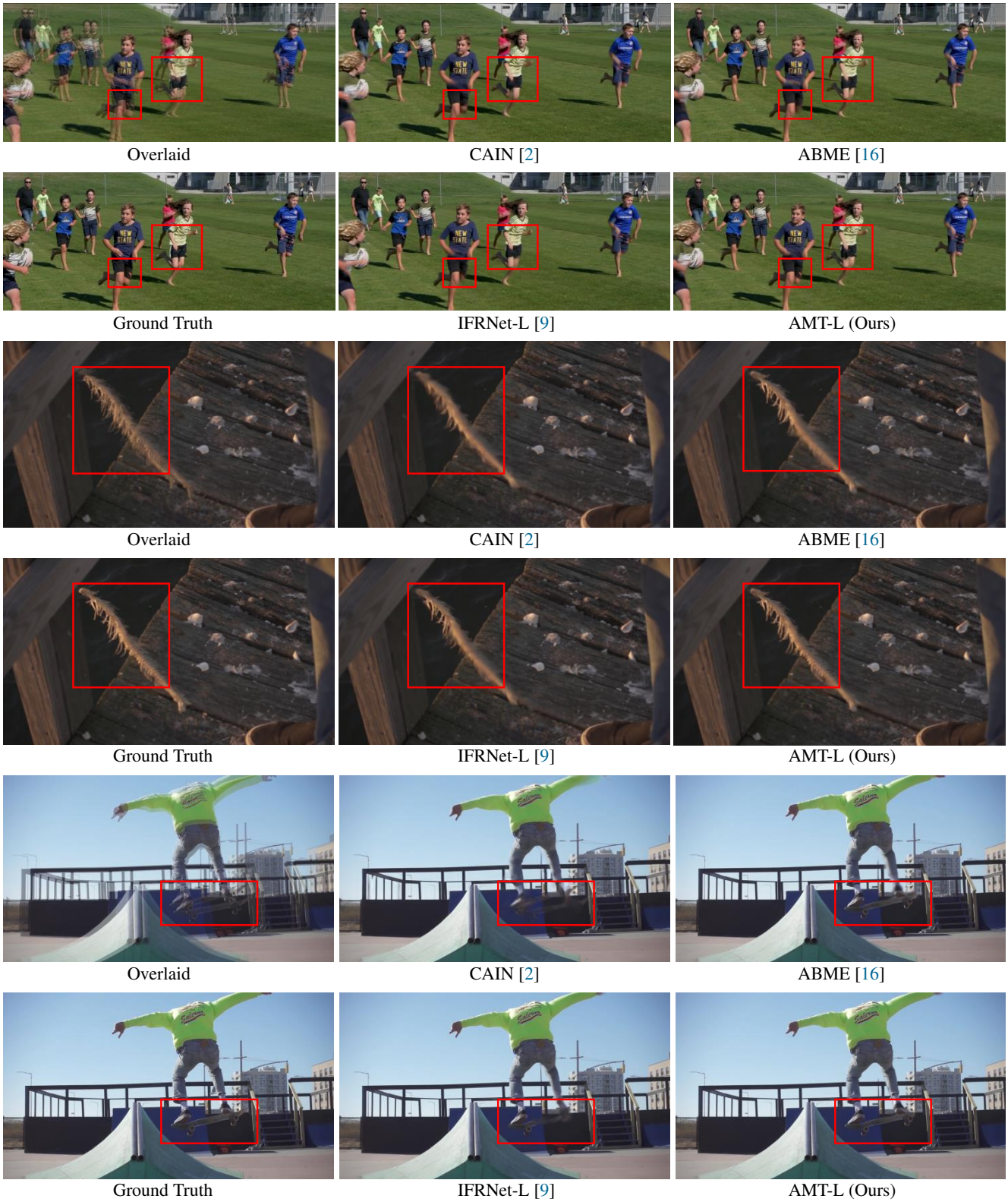



Figure 10. Visual comparison for the methods with relatively high computational complexity on the Hard partition in SNU-FILM dataset [2].

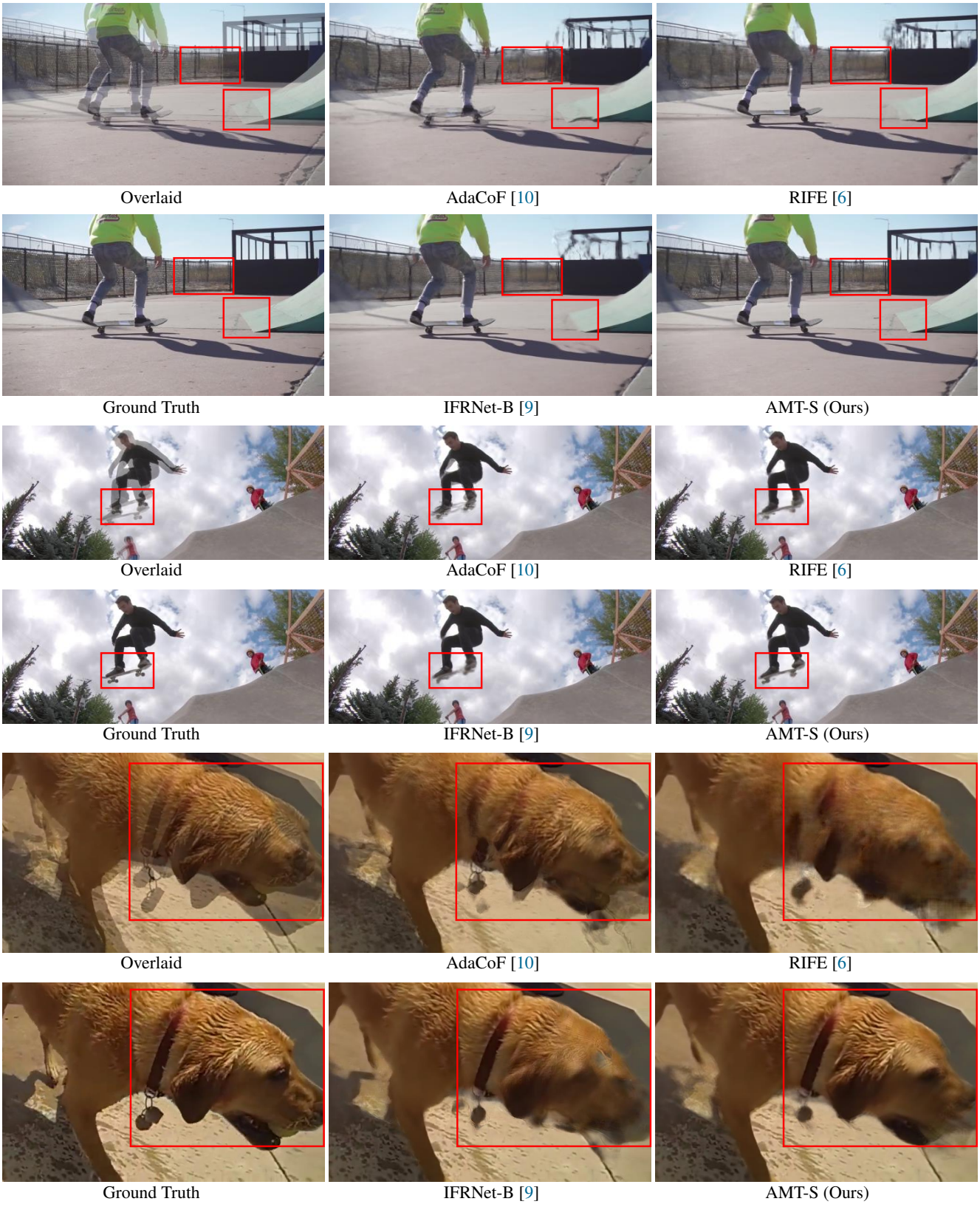



Figure 11. Visual comparison for the methods with low computational complexity on the Extreme partition in SNU-FILM dataset [2].

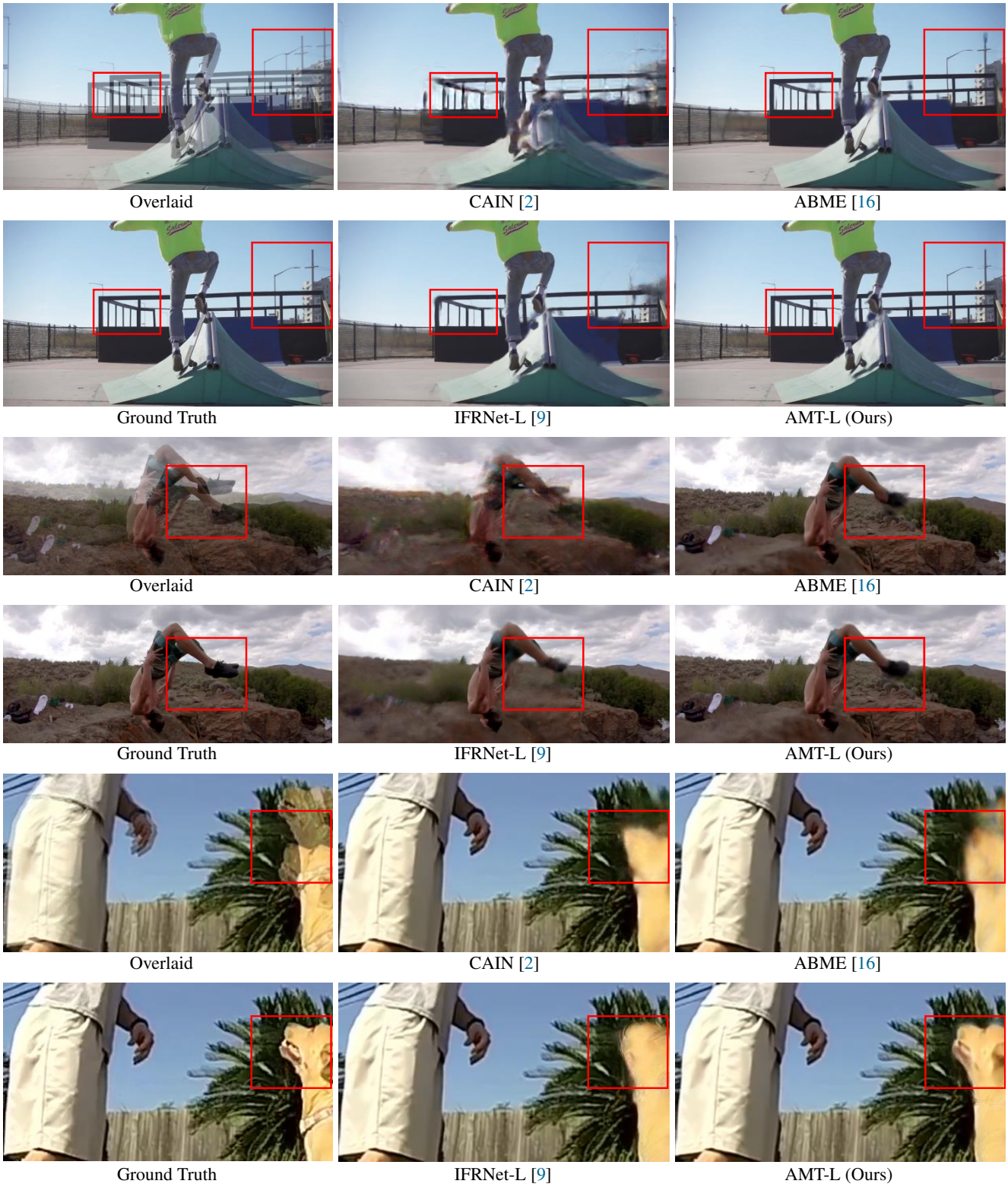



Figure 12. Visual comparison for the methods with relatively high computational complexity on the Extreme partition in SNU-FILM dataset [2].

tion. *arXiv preprint arXiv:1607.08022*, 2016. 1

[22] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1d attention and correlation. In *ICCV*, 2021. 2

[23] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022. 2

[24] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 2019. 3, 4, 7, 8

[25] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 2