Anything in Any Scene: Photorealistic Video Object Insertion

Chen Bai, Zeman Shao, Guoxiang Zhang, Di Liang, Jie Yang, Zhuorui Zhang, Yujian Guo,
Chengzhang Zhong, Yiqiao Qiu, Zhendong Wang, Yichen Guan, Xiaoyin Zheng, Tao Wang, Cheng Lu

XPeng Motors

{chenbai, zemans, guoxiangz, liangd2, yangj23, zhangzr, guoyj4, chengzhangz, yiqiaoq
zhendongw, guanyc, xiaoyinz, taow, luc}@xiaopeng.com

Abstract

Realistic video simulation has shown significant potential across diverse applications, from virtual reality to film production. This is particularly true for scenarios where capturing videos in real-world settings is either impractical or expensive. Existing approaches in video simulation often fail to accurately model the lighting environment, represent the object geometry, or achieve high levels of photorealism. In this' paper, we propose Anything in Any Scene, a novel and generic framework for realistic video simulation that seamlessly inserts any object into an existing dynamic video with a strong emphasis on physical realism. Our proposed general framework encompasses three key processes: 1) integrating a realistic object into a given scene video with proper placement to ensure geometric realism; 2) estimating the sky and environmental lighting distribution and simulating realistic shadows to enhance the light realism; 3) employing a style transfer network that refines the final video output to maximize photorealism. We experimentally demonstrate that Anything in Any Scene framework produces simulated videos of great geometric realism, lighting realism, and photorealism. By significantly mitigating the challenges associated with video data generation, our framework offers an efficient and cost-effective solution for acquiring high-quality videos. Furthermore, its applications extend well beyond video data augmentation, showing promising potential in virtual reality, video editing, and various other video-centric applications. Please check our project website https://anythinginanyscene.github.io for access to our project code and more high-resolution video results.

1. Introduction

The image and video simulation has exhibited success in various applications, ranging from virtual reality to film production. The capability to generate diverse and high-quality visual content through realistic image and video simulation holds the potential to advance these fields, introducing new possibilities and applications. Although the images and videos captured in real-world settings are invaluable for their authenticity, they often suffer from the limitation of long-tail distribution. This results in common scenarios being over-represented, while rare yet crucial situations are under-represented, presenting a challenge known as the out-of-distribution problem. Traditional methods of addressing these limitations through video collection and editing prove impractical or excessively costly due to the inherent difficulty in encompassing all possible situations. The significance of video simulation, especially through the integration of existing videos with newly inserted objects, becomes paramount in overcoming these challenges. By generating large-scale, diverse, and realistic visual content, video simulation contributes to the enhancement of applications in virtual reality, video editing, and video data augmentation.

However, generating a realistic simulated video with consideration of physical realism is still a challenging open problem. Existing methods often exhibit limitations by concentrating on specific settings, particularly indoor environments [9, 26, 45, 46, 57]. These methods may not adequately address the complexities of outdoor scenes, including diverse lighting conditions and fast-moving objects. Methods relying on 3D model registration are constrained in integrating only limited classes of objects [12, 32, 40, 42]. Many approaches neglect essential factors such as modeling the lighting environment, proper object placement, and achieving photorealism [12, 36]. Failed cases are illustrated in Figure 1. Consequently, these limitations significantly constrain their applications in fields that need highly scalable, geometrically consistent, and realistic scene video simulation, such as autonomous driving and robotics.

In this paper, we propose a comprehensive framework Anything in Any Scene for the photorealistic video object insertion that addresses these challenges. The framework is

(a) The inserted car has an inconsistent shadow to another car because of the wrong lighting environment estimated.

(b) The car is in the air because of a wrong placement location determined.

(c) The inserted car in the scene has a significant difference in texture compared to another car, which makes the image lack photorealism.

Figure 1. Examples of simulated video frame with wrong lighting environment estimation, false object placement position, and unrealistic texture style, which make the image lack physical realism

designed to have universal applicability, and is adaptable to both indoor and outdoor scenes, ensuring physical accuracy in terms of geometric realism, lighting realism, and photorealism. Our goal is to create video simulations that are not only beneficial for visual data augmentation in machine learning but also adaptable to various video applications, such as virtual reality and video editing.

The overview of our Anything in Any Scene framework is shown in Figure 2. We detail our novel and scalable pipeline for building a diverse asset bank of scene video and object mesh in Section 3. We introduce a visual data query engine designed to efficiently retrieve relevant video clips from visual queries using descriptive keywords. Following this, we present two methods for generating 3D meshes, leveraging existing 3D assets as well as multi-view image reconstructions. This allows the insertion of any desired object without limitation, even if it is highly irregular or semantically weak. In Section 4, we detail our approach for integrating objects into dynamic scene video with a focus on maintaining physical realism. We design an object placement and stabilization method described in Section 4.1, ensuring the inserted object is stably anchored across continuous video frames. Addressing the challenge of creating realistic lighting and shadow effects, we estimate sky and environmental lighting and generate realistic shadows during the rendering process, as described in Section 4.2. The resulting simulated video frames inevitably contain unrealistic artifacts that differ from real-world captured videos, such as imaging quality discrepancies in noise level, color fidelity, and sharpness. We adopt a style transfer network to enhance the photorealism in Section 4.3.

The simulated videos produced from our proposed framework reach a high degree of lighting realism, geometrical realism, and photorealism, outperforming the others both qualitatively and quantitatively as shown in Section 5.3. We further showcase in Section 5.4 the application of our simulated videos in the training perception algorithm to verify its practical value. The Anything in Any Scene framework is able to create a large-scale, low-cost video

dataset for data augmentation with time efficiency and realistic visual quality, which alleviates the burden of video data generation and potentially ameliorates the long-tail distribution and out-of-distribution challenges. With its generic framework design, the Anything in Any Scene framework can easily incorporate improved models and new modules, such as an improved 3D mesh reconstruction method, further enhancing video simulation performance.

Our main contributions can be summarized as follows:
1. We introduce a novel and scalable Anything in Any Scene framework for video simulation, capable of integrating any object into any dynamic scene video.
2. Our framework uniquely focuses on preserving geometric realism, lighting realism, and photorealism in video simulations, ensuring high-quality and realistic outputs.
3. We conducted extensive validations, demonstrating the ability of the framework to produce realistic video simulations, significantly expanding the scope and potential application in this field.

2. Related Work

Image Synthesis and Editing: Encompassing tasks from image inpainting to style transfer has attracted significant attention in both academic and industry communities. The traditional methods are mostly based on pixels, patches, and low-level image features, often lacking high-level semantic information. Specifically, the image inpainting methods replicate pixels or patches for image recovery [2, 3, 10, 19, 27]. The non-parametric-based texture synthesis methods re-sample the pixels of a given source texture to generate photorealistic textures [13, 28]. The style transfer methods, such as image analogies [21], perform example-based stylization using patches.

Deep learning networks, particularly Generative Adversarial Networks (GAN) [17], have demonstrated significant capabilities in computer vision and image processing tasks, achieving impressive success in image generation. Various GANs, such as MGANs [30], SGAN [25], and PS-GAN [4], have shown remarkable proficiency in the task
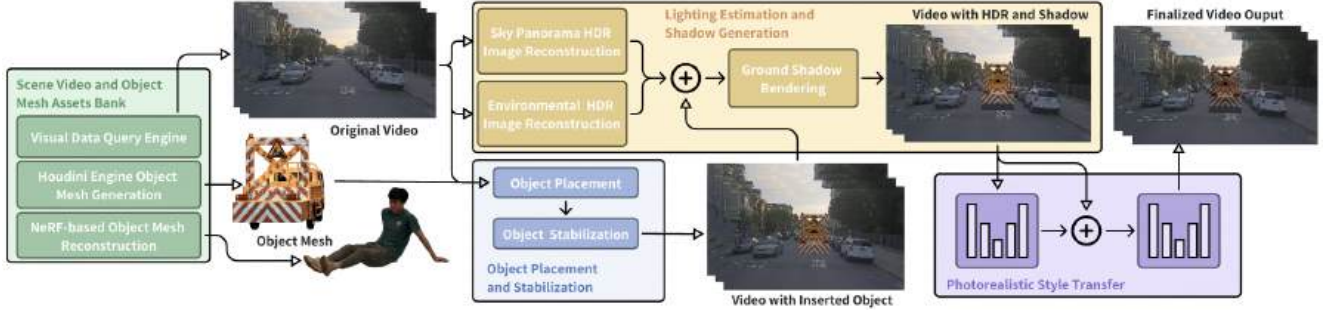
Figure 2. Overview of proposed Anything in Any Scene framework for photorealistic video object insertion

of texture synthesis. Additionally, GANs have been successfully applied to contextual image inpainting [38] and multi-scale image completion [59]. The pix2pix [24] and cycleGAN [62] leverage GAN architecture to train generative models for style transfer. The images generated by GANs tend to be less blurred and exhibit higher realism, aligning closely with distributions of training image data.

Video Synthesis and Editing: Transitioning from image to video synthesis requires addressing additional challenges, particularly maintaining temporal consistency.

Unconditional video synthesis methods, such as [44], [52], and [53], take a random noise as input and model both spatial and temporal correlation to generate video. However, they often result in constrained motion patterns in output video sequences. In contrast, conditional video synthesis methods employ conditional GAN [37] to train a generative model for video generation based on input content. In [55] and its following work [56], the generative network is conditioned on the previous frame of the source video for each subsequent frame generation. [34] take this approach further by considering all previously generated frames, achieving improved long-term temporal consistency in their video synthesis.

Additionally, the automatic video synthesis methods proposed in [29] and [23] insert the object's video into another video using spatial and temporal information. Recently, the GeoSim framework proposed in [7] has achieved impressive results in car insertion into a given real-world driving scene video, though its application to less common objects and diverse types of scene video remains limited. Our work seeks to bridge this gap, expanding the potential for any object insertion in any scene video.

3. Scene Video and Object Mesh Assets Bank

Our goal with the Anything in Any Scene framework is to generate large-scale and high-quality simulation videos by composition of dynamic scene videos and objects of interest. To achieve this, an assets bank of both scene videos and object meshes is required for simulated video composition.

In order to efficiently locate target videos for composi-

tion from a large-scale video assets bank, we proposed a visual data query engine that is used to retrieve the relevant scene video clips for simulated video composition based on the given visual clue descriptors. The mesh model of the target object is required before its insertion into an existing video clip. We introduced the 3D mesh generation of the target object by using the Houdini Engine from existing 3D assets and a NeRF-based 3D reconstruction from multi-view images, which enables theatrically unlimited classes of objects to be inserted into the existing scene video.

Detailed descriptions of our mesh assets bank can be found in supplementary materials.

4. Realistic Video Simulation

To achieve video simulation with geometric realism, lighting realism, and photorealism, our proposed framework consists of the following three main components:

1. Object Placement and Stabilization (Section 4.1)
2. Lighting and Shadow Generation (Section 4.2)
3. Photorealistic Style Transfer (Section 4.3)

4.1. Object Placement and Stabilization

Inserting an object into a background video for video composition requires the object placement location determined for each frame in the video sequence. We designed and proposed a novel object placement method with the consideration of occlusion with other existing objects in the scene, which is described in Section 4.1.1.

However, placement locations that are independently estimated from each single frame could yield unrealistic movement tracks since the video temporal information has not been considered. To address this issue, we propose an object placement stabilization method in Section 4.1.2 to correct the placement location in each frame. We employ optical flow tracking between consecutive frames to ensure the inserted object behaves realistically across the continuous video frames.
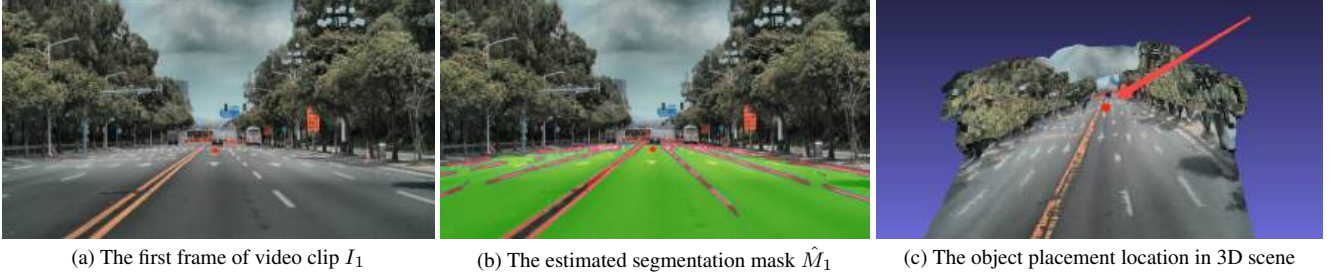
3

(a) The first frame of video clip I_1 (b) The estimated segmentation mask $\hat{M}_1$ (c) The object placement location in 3D scene

Figure 3. Example of driving scene video for object placement. The red point in each image is the location for object insertion.

4.1.1 Object Placement

Suppose there are $N + T$ continuous frames, the first N frames are the target frames that we aim to integrate the inserted object into, and the last T frames are used as reference for object placement. We assume the world coordinate of camera location in the frame I_{N+T} is the origin $O_w = [0, 0, 0, 1]$, and the camera coordinate system is aligned with the world coordinate system at this frame I_{N+T}. We place the inserted object at the location of origin in the world coordinate which is the same location as the camera itself in the frame I_{N+T}. To determine the pixel coordinates for object placement in the first N consecutive frames, we project the origin from the world coordinate to the pixel coordinate based on the camera intrinsic matrix $\mathbf{K}$ and the camera pose including rotation matrix $\mathbf{R}_n$ and translation vector $\mathbf{t}_n$ at each frame I_n. The placement pixel coordinate $\tilde{o}_n$ at the frame I_n is determined by:

$$\tilde{o}_n = \mathbf{K}[\mathbf{R}_n|\mathbf{t}_n]O_w \qquad (1)$$

The placement of the inserted object within a video clip should avoid occlusion with other existing objects in the scene. We estimated the semantic segmentation mask $\hat{M}_n$ for each frame I_n by using off-the-shelf models. The pixel $\hat{M}_n(\tilde{o}_n)$ denotes the category at the pixel location $\tilde{o}_n$, representing the origin in the world coordinate projected into the pixel coordinate in the frame I_n. This predicted category serves as a reference to determine whether the projected point location for object insertion is occluded by other objects in the scene.

We show an example of a driving scene in Figure 3. The first frame of the video clip and its associated estimated segmentation mask are shown in Figure 3a and Figure 3b. The red point in Figure 3c is the origin of the world coordinate and also the camera location in the frame I_{N+T}, we placed the object at this location. As the estimated segmentation mask shown in Figure 3b, the green region indicates the road area and the red region indicates the road lane. After the object placement location is projected back from the world coordinate to the pixel coordinate, the placement is located in the road area as indicated in the semantic segmentation, which is a plausible place to insert a road vehicle

in a driving scene video.

4.1.2 Object Placement Stabilization

Firstly, we select a 3D point with world coordinate $P_w = [X, Y, Z, 1]$, and follow the Equation 1 to project it from the world coordinate into the pixel coordinate $\tilde{p}_n$ in each frame I_n of the first $N + 1$ frames. We then estimate the optical flow between each two consecutive frames and obtain the selected 3D point P_w pixel coordinate $\hat{p}_n$ in the frame I_n through the image warping of $\tilde{p}_{n+1}$ and the estimated optical flow. The object placement stabilization can be interpreted as the optimization of camera pose for each frame I_n. Specifically, we optimize the camera pose rotation matrix $\mathbf{R}_n$ and translation vector $\mathbf{t}_n$ at each frame I_n by minimizing the 3D-to-2D projection error of $\hat{p}_n$ with the comparison to $\tilde{p}_n$. To achieve a better performance in placement stabilization, we select M points and optimize the rotation matrix $\mathbf{R}'_n$ and translation vector $\mathbf{t}'_n$, which can be expressed as:

$$
\begin{aligned}
(\mathbf{R}'_n, \mathbf{t}'_n) &= \arg\min \sum_{i=1}^{M} (\hat{p}_n - \tilde{p}_n)^2 \\
&= \arg\min_{(\mathbf{R}_n, \mathbf{t}_n)} \sum_{i=1}^{M} (\hat{p}_n - \mathbf{K}[\mathbf{R}_n|\mathbf{t}_n]P_w)^2
\end{aligned} \qquad (2)
$$

Lastly, we update the rotation matrix and translation vector in Equation 1 by $\mathbf{R}'_n$ and $\mathbf{t}'_n$, and calculate the updated object placement pixel coordinate $\tilde{o}_n$ for each frame I_n.

We also adjust X and Y values of the selected 3D point P_w to ensure that the projected 2D point can be tracked in consecutive frames based on the estimated optical flow. For example in the driving scene view, we shifted the selected 3D points by adjusting the Y value so that the projected 2D points are the corner points of the white road lane.

4.2. Lighting Estimation and Shadow Generation

One important key to creating a realistic simulated video with an integrated object is to generate accurate lighting and shading effects for the inserted object. The position and luminance of the lighting in the scene, such as the sun

Original Image   HDR Image   Lighting Distribution

Figure 4. Examples of original sky image, reconstructed HDR image, and its associated sun lighting distribution map

for the outdoor scene and the environment for the indoor scene, affect the inserted object's visual appearance during the rendering process.

To simulate an accurate lighting and shading effect during the rendering process, we first introduced a High Dynamic Range (HDR) panoramic image reconstruction method in Section 4.2.1. Lastly, we rendered the shadow of the inserted object based on the estimated position of the main lighting source in Section 4.2.2.

4.2.1 HDR Panoramic Image Reconstruction

The Low Dynamic Range (LDR) images captured by the consumer camera are usually over-saturated due to the extremely high brightness of the main lighting compared to surrounding environmental lighting, which makes it much more difficult to estimate the position and luminance distribution of the main lighting. To address this issue, we first use an image inpainting network to infer the surround view of lighting distribution for rendering. We then adapt a sky HDR reconstruction network to identify the lighting source position and generate the HDR panoramic image.

Panorama Image Inpainting: The image captured by the consumer camera has a limited Field of View (FOV), which leads to missing lighting in the rendering process. We address this task by translating it into an inpainting task which infers a panorama image from a limited FOV image. Furthermore, we aim to infer the surround view image by using a diffusion model [22, 50]. We proposed to use an image-to-image diffusion model which is a conditional diffusion model that converts samples from a standard Gaussian distribution into samples from a data distribution through an iterative denoising process conditional on an input. In our task, we adapt an existing model [43] and make it conditional on the input image to generate a panoramic image.

Luminance Distribution Estimation: The HDR image reconstruction method proposed in [47] utilizes a Generative Adversarial Network (GAN) to train encoder-decoder networks that model the sun and sky luminance distribution. The input is a single outdoor LDR panoramic image and a U-Net [41] architecture network with ResNet [20] as its backbone is used to estimate the sky region luminance
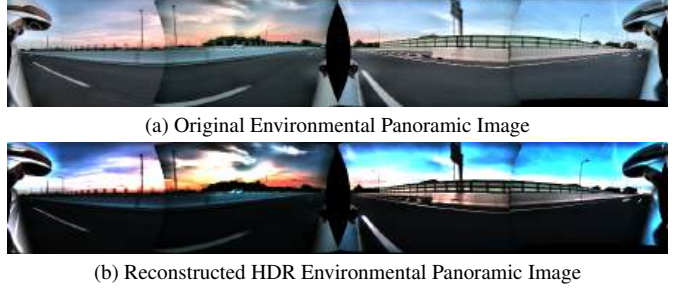

(a) Original Environmental Panoramic Image

(b) Reconstructed HDR Environmental Panoramic Image

Figure 5. Examples of Original and Reconstructed HDR Environmental Panoramic Image

distribution L_{sky}.

Another modified VGG16 network [49] is employed to estimate the sun position probability map $x_{i,j}$ which represents the probability at pixel (i, j) in the input LDR panoramic image containing the sun. The output feature maps from the CNN blocks the VGG are concatenated together as input fed into the convolutional layers for encoding the sun radiance map which is the Dirac delta function expressed by:

$$\delta(x_{i,j}, \tau, \beta) = \frac{\tau}{\beta\sqrt{\pi}} exp(-\frac{(1 - x_{i,j})^2}{\beta}) \qquad (3)$$

where τ and β are the transmittance and sharpness values of the sky. The sun radiance map is then merged with sun regions to generate the sun region luminance distribution L_{sun}. The L_{sun} and L_{sky} are applied to an inverse tone mapping operation and blended to generate the final output HDR map L.

We adapt this method in our lighting estimation module and follow the same process as described in [47] that uses GAN to re-train the network for generating HDR map L. We then applied L to the inserted object in the video frame.

Environmental HDR Image Reconstruction: As for the outdoor scenario, the sun as the main lighting is not the only one that can affect the visual appearance of the inserted object, we also need to consider the environmental lighting due to the diffuse reflection in order to achieve more realistic rendering outcomes. To reconstruct the environmental HDR image, we collect multiple side-view LDR images of the scene and recover them into HDR images by using an existing model to learn the continuous exposure value representations [6]. We followed the same process to estimate the camera extrinsic parameters for each side-view image and stitch them into one HDR panoramic image (Example of the environmental HDR image as shown in Figure 5). Thus we obtained the estimated environmental light distribution from the multiple side-view images, then we can apply it to the inserted object rendering process.

(a) Image with generated shadow   (b) Associated segmentation mask

Figure 6. Example of generated shadow for the inserted object

4.2.2 Object Shadow Generation

Since we've estimated the location and distribution of the main lighting source, *i.e.* sun for outdoor scene and light for indoor scene, we rendered the shadow of the inserted object by the 3D graphics application Vulkan [54] which offers higher performance and more efficient computing resource usage. Furthermore, we integrated the ray tracing into the Vulkan application for a better performance of realistic rendering [16]. Examples of the generated shadow for the inserted objects are shown in Figure 6.

4.3. Photorealistic Style Transfer

The simulated videos inevitably contain unrealistic artifacts, such as inconsistent illumination and color balancing, which are not included in videos captured in the real-world scenario. To address this issue, we proposed to use an image inpainting network that faithfully transfers the style to enhance the photorealism of simulated video sequences.

Specifically, we adapt the coarse-to-fine mechanism proposed in [61], which is originally designated to inpaint missing regions in an image. We utilized the coarse network and refinement network in [61], both of them consist of dilated convolution layers to generate the refined image based on the input image. We modified the input configuration for the two networks. The coarse network takes an image with black pixels filled in the foreground region, a binary mask indicating the foreground region, and a foreground image of the inserted object with black pixels filled in the background region. The refinement network takes the same input as the coarse network along with output from the coarse network, and it generates final refined image results.

To train the generative model, we adopt the same training strategy proposed in [61] which uses the WGAN [1] loss, and its objective function can be expressed as:

$$\min_{G} \max_{D \in \mathcal{D}} \mathbb{E}_{x \sim \mathbb{P}_r}[D(x)] - \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[D(\tilde{x})] \qquad (4)$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions, $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution implicitly defined by $\tilde{x} = G(z)$, and z is the input to the generator.

We added the gradient penalty term proposed in [18] to improve the WGAN and applied it to pixels in the foreground region. Thus the penalty function can be expressed

as:

$$\lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}(||\nabla_{\hat{x}} D(\hat{x}) \odot (1 - m)||_2 - 1)^2 \qquad (5)$$

where $\hat{x}$ sampled from the straight line between points sampled from the distribution $\mathbb{P}_r$ and $\mathbb{P}_g$, and m is the input binary mas of the foreground region.

5. Experimental Evaluation

In this section, we describe the evaluation details of our proposed method for video simulation. We introduce evaluation metrics in Section 5.1 to quantify performance. The video datasets covering both indoor and outdoor scenes used for validation are listed in Section 5.2. We perform an ablation analysis to evaluate the effectiveness of each module of our framework in Section 5.3. Lastly, we showcase the application of the framework in downstream perception tasks in Section 5.4.

5.1. Evaluation Metrics

We adopt the following two evaluation metrics used in [7] to assess the quality of simulated videos generated by our proposed framework. We report the average values for each metric across all video frames in a dataset.

Human Score: This metric measures the percentage of participants who prefer the results from one method over those from the baseline method in a human A/B test. Detailed descriptions of human study can be found in supplementary materials. Additionally, the complete set of video pairs and GUI application used in this study is available on our website at https://anythinginanyscene.github.io. We encourage peer researchers to download and review these video comparisons, or to conduct their own human studies for verification of our results.

Frechet Inception Distance (FID): This metric quantifies the realism and diversity of the generated images by comparing the distribution of generated images with that of groundtruth images. Lower scores indicate greater similarity, with a zero score implying identical image sets.

5.2. Evaluation Data

To demonstrate the performance of our method for realistic video composition of various scene videos and objects, we validate our method using both outdoor and indoor scene video datasets and diverse inserted object items.

Outdoor Scene Video: PandaSet [58] is a multi-modal dataset capturing self-driving scenes in various conditions, including different times of day and weather. We utilized 95 out of all 103 video clips from this dataset, each containing 8 seconds of frames sampled at 10 Hz.

Indoor Scene Video: ScanNet++ [60] is a large-scale dataset of indoor scenes created by 3D scanning real environments The dataset includes DSLR images, RGB-D sequences, and semantic and instance annotations, providing

| (a) DoveNet | (b) StyTR2 | (c) PHDiffusion | (d) Ours |

Figure 7. Qualitative comparison of the simulated video frame from PandaSet dataset using different style transfer networks.

a comprehensive resource for evaluating our methods. We provide the experimental results of the indoor scene video dataset in the supplementary materials.

Object Mesh Assets: We used the methods introduced in Section 3 to generate 3D object meshes, focusing on various objects, including different types of vehicles and pedestrian models.

5.3. Experimental Results

To assess the performance of various style transfer networks, we compared different methods: a CNN-based method DoveNet [8], transformer-based method StyTR2 [11], diffusion model-based method PHDiffusion [33], and our method introduced in Section 4.3. For the human study, we use our framework without the style transfer module as the baseline for comparison. We summarize the result of the comparison in Table 1. Our transfer network achieved the lowest FID at 3.730 and the highest human score at 61.11%, outperforming the alternative methods.

Ablation Studies: To investigate the effectiveness of each key module, we conducted ablation studies and evaluated the performance. We removed one module from our framework at a time: placement (w/o placement), HDR image reconstruction (w/o HDR), shadow generation (w/o shadow), and style transfer (w/o style transfer). In this human study, the w/o style transfer method served as the baseline, and was compared to all other ablation methods. The results are summarized in Table 2. The absence of placement, HDR, and style transfer modules resulted in higher FIDs. Notably, adding shadows significantly enhanced the perceived realism for human observers, though this improvement was not proportionately reflected in the FID score. This discrepancy suggests a potential gap between computational assessments of perceptual quality and human judgment, as also noted in previous research [7]. Our proposed method achieved a human score above 50%, and the others scored below 50%, highlighting the contribu-

Method	Human Score(%)	FID
Proposed method	**61.11**	**3.730**
StyTR2 style transfer	58.89	4.091
PHDiffusion style transfer	47.22	4.554
DoveNet style transfer	47.78	3.999
w/o style transfer	N/A	4.499

Table 1. Experimental results for different style transfer networks plugged into our Anything in Any Scene framework.

Method	Human Score(%)	FID
Proposed method	**61.11**	3.730
w/o placement	25.56	4.327
w/o HDR	43.05	3.793
w/o shadow	37.78	3.485
w/o style transfer	N/A	4.499

Table 2. Experimental results for ablation analysis of modules in our Anything in Any scene framework. Note that the baseline w/o style transfer method theoretically has a human score of 50%

tion of each module in our proposed framework.

Qualitative comparison: In Figure 7, we provide a qualitative comparison of sample video frames using different style transfer networks applied to the outdoor scene dataset PandaSet. Figure 7a, 7b and 7c show images refined by DoveNet, StyTR2, and PHDiffusion, respectively. The inserted object in these images exhibits a color tone that is not consistent with the scene's lighting and weather conditions. Conversely, the image refined by our proposed method as shown in Figure 7d demonstrates the best visual quality among the four, aligning with the results reported in Table 1 that show our method outperforming others in both FID and human study scores. This indicates that an improved style transfer network can significantly enhance photorealism within our Anything in Any Scene framework.

Furthermore, we evaluate the visual quality of videos generated by the Anything in Any Scene framework by

| (a) w/o placement | (b) w/o shadow | (c) w/o HDR | (d) w/o style transfer | (e) Ours |

Figure 8. Qualitative comparison of the simulated video frame from PandaSet dataset under various rendering conditions.

removing one module at a time, using the outdoor scene dataset PandaSet as a reference. This evaluation is visually illustrated with two comparison samples in Figure 8. In Figure 8c and Figure 8d, we observe that the inserted object exhibits color textures that are inconsistent with the surrounding environment and other objects in the scene. Furthermore, Figure 8b highlights an instance where the inserted object lacks a generated shadow. This absence creates a visual effect where the object appears as if it is in the air, highlighting the importance of shadow rendering for realistic simulation. In contrast, Figure 8e shows the visual quality of videos generated by our framework, where the inserted object displays a high degree of consistency with the scene in terms of geometry, lighting, and overall photorealism. This demonstrates the capability of the Anything in Any Scene framework to achieve realistic integration of objects into diverse scene settings.

5.4. Downstream Perception Evaluation

Real-world datasets often exhibit a long-tailed class distribution, where a few common classes are over-represented, while a majority of classes are under-represented. This imbalance poses significant challenges for deep learning models, leading to biases towards common classes during training and worse performance on rare classes during inference.

To address this problem, we investigate the usage of the Anything in Any Scene framework to generate synthetic images containing rare cases for data augmentation. We perform the evaluation on the CODA dataset [31], an amalgamation of image data from KITTI [15], nuScenes [5], and ONCE [35] datasets, including 1,500 real-world driving scenes and over 30 object categories

The goal of this task is to insert 9 different rare object categories into images from the CODA2022 validation dataset, with each category comprising less than 0.4% of total bounding boxes. We trained three models: YOLOX-S, YOLOX-L, and YOLOX-X [14], on a subset of 2930 images from the dataset, reserving another 977 images for test-

Method	Data	mAP	
YOLOX-S	Original	0.186	0.037 ↑
	Original + Ours	**0.223**	
YOLOX-L	Original	0.260	0.011 ↑
	Original + Ours	**0.271**	
YOLOX-X	Original	0.249	0.026 ↑
	Original + Ours	**0.275**	

Table 3. Performance of the YOLOX models trained on the original images from the CODA dataset compared to their performance when trained on a combination of original and augmented images using our Anything in Any Scene framework. We report the mAP that represents the mean for all 9 object categories.

ing. We then employed our Anything in Any Scene framework to augment these training images by inserting various objects into them. This process produced an augmented set of training images that replaced the original ones in the training dataset. We applied the same training strategy and re-train the models on the augmented training dataset.

We evaluate the performance of the three models by training them on both the original and the augmented datasets, followed by testing them on the same test dataset. The results, detailed in Table 3, indicate an improvement in mean Average Precision (mAP) for all three models. Specifically, there is an enhancement of 3.7% in mAP for YOLOX-S, 1.1% for YOLOX-L, and 2.6% for YOLOX-X.

6. Conclusion

In this work, we proposed an innovative and scalable framework, Anything in Any Scene, designed for realistic video simulation. Our proposed framework seamlessly integrates a wide range of objects into diverse dynamic videos, ensuring the preservation of geometric realism, lighting realism, and photorealism. Through extensive demonstrations, we have shown its efficacy in alleviating challenges associated with video data collection and generation, offering a cost-effective and time-efficient solution adaptable to a variety of scenarios. The applica-

tion of our framework has shown notable improvements in downstream perception tasks, particularly in addressing the long-tailed distribution issue in object detection. The flexibility of our framework allows for straightforward integration of improved models for each of its modules, our framework stands as a robust foundation for future explorations and innovations in the field of realistic video simulation.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. 6

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[3] Connelly Barnes, Fang-Lue Zhang, Liming Lou, Xian Wu, and Shi-Min Hu. Patchtable: Efficient patch queries for large datasets and applications. *ACM Transactions on Graphics (ToG)*, 34(4):1–10, 2015. 2

[4] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. *arXiv preprint arXiv:1705.06566*, 2017. 2

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 8

[6] Su-Kai Chen, Hung-Lin Yen, Yu-Lun Liu, Min-Hung Chen, Hou-Ning Hu, Wen-Hsiao Peng, and Yen-Yu Lin. Learning continuous exposure value representations for single-image hdr reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12990–13000, 2023. 5

[7] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7230–7240, 2021. 3, 6, 7

[8] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020. 7, 3

[9] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016. 1

[10] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, pages II–II. IEEE, 2003. 2

[11] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022. 7, 3

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1

[13] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, pages 1033–1038. IEEE, 1999. 2

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 8, 5

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 8

[16] GitHub. Ray tracing examples and tutorials, 2023. https://github.com/nvpro-samples/vk_raytracing_tutorial_KHR/tree/master [Accessed: (October 16, 2023)]. 6

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 6

[19] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)*, 26(3):4–es, 2007. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[21] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 557–570. 2023. 2

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[23] Hao-Zhi Huang, Sen-Zhe Xu, Jun-Xiong Cai, Wei Liu, and Shi-Min Hu. Temporally coherent video harmonization using adversarial networks. *IEEE Transactions on Image Processing*, 29:214–224, 2019. 3

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 3

[25] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016. 2

[26] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 1

[27] Nikos Komodakis and Georgios Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007. 2

[28] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *Acm transactions on graphics (tog)*, 22 (3):277–286, 2003. 2

[29] Donghoon Lee, Tomas Pfister, and Ming-Hsuan Yang. Inserting videos into videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10070, 2019. 3

[30] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 702–716. Springer, 2016. 2

[31] Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. *arXiv preprint arXiv:2203.07724*, 2022. 8

[32] Wei Li, CW Pan, Rong Zhang, JP Ren, YX Ma, Jin Fang, FL Yan, QC Geng, XY Huang, HJ Gong, et al. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28):eaaw0863, 2019. 1

[33] Lingxiao Lu, Jiangtong Li, Junyan Cao, Li Niu, and Liqing Zhang. Painterly image harmonization using diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 233–241, 2023. 7, 3

[34] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 359–378. Springer, 2020. 3

[35] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Chunjing Xu, et al. One million scenes for autonomous driving: Once dataset. 2021. 8

[36] Mark Martinez, Chawin Sitawarin, Kevin Finch, Lennart Meincke, Alex Yablonski, and Alain Kornhauser. Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars. *arXiv preprint arXiv:1712.01397*, 2017. 1

[37] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[38] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 3

[39] QIU023. Guivideodisplayselector: A simple tkinter-based gui application for video comparison and selection., 2023. https://github.com/QIU023/GUIVideoDisplaySelector [Accessed: (November 8, 2023)]. 4

[40] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 1

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 5

[42] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 1

[43] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 5

[44] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 3

[45] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 1

[46] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 1

[47] Gyeongik Shin, Kyeongmin Yu, Mpabulungi Mark, and Hyunki Hong. Hdr map reconstruction from a single ldr sky panoramic image for outdoor illumination estimation. *IEEE Access*, 2023. 5

[48] SideFX. Unreal plug-in, 2023. https://www.sidefx.com/products/houdini-engine/plug-ins/unreal-plug-in/ [Accessed: (October 24, 2023)]. 1

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 5

[51] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. Delicate textured mesh recovery from nerf via adaptive surface refinement. *arXiv preprint arXiv:2303.02091*, 2023. 2

[52] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 1526–1535, 2018. 3

[53] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016. 3

[54] Vulkan. Vulkan, cross platform 3d graphics, 2023. https://www.vulkan.org/ [Accessed: (October 16, 2023)]. 6

[55] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 3

[56] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *arXiv preprint arXiv:1910.12713*, 2019. 3

[57] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 1

[58] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021. 6, 2

[59] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017. 3

[60] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 6, 2

[61] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 6, 3

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 3

Anything in Any Scene: Photorealistic Video Object Insertion

Supplementary Material

In this supplementary material, we include additional technical details and a broader range of quantitative and qualitative results of our proposed method. We first describe additional details on the assets bank in Section 7, the object placement in Section 8, the lighting estimation and shadow generation in Section 9, and the photorealistic style transfer in Section 10. We then introduce the details of how we conducted the human study to compare different simulated videos in Section 11.

Furthermore, we also present the quantitative validation results of our method using the indoor dataset ScanNet++ in Section 12, and further details on the downstream tasks we conducted are available in Section 13. We provide more details of the result of downstream task we performed in Section 13. To visually underscore the effectiveness of our approach, we include an extensive gallery of simulated videos generated by our framework alongside others for comparative analysis in Section 14.

Finally, we kindly suggest that reviewers view our supplementary video files (*sample_video_outdoor.mp4* for an outdoor scene and *sample_video_indoor.mp4* for an indoor scene) to better appreciate the capabilities of our simulation method through these representative examples.

7. Assets Bank Details

Our Anything in Any Scene framework aims to create large-scale simulation videos by integrating dynamic scene videos with objects of interest. This requires an asset bank of scene videos and object meshes. To facilitate this, we developed a visual data query engine for efficiently selecting scene videos based on visual descriptors. Additionally, we employ the Houdini Engine and Neural Radiance Fields (NeRF)-based reconstruction for 3D mesh generation, enabling the integration of diverse objects into these videos.

7.1. Visual Data Query Engine

In order to efficiently locate target videos for composition from a large-scale video assets bank, our method leverages a visual data query engine. This engine is designed to retrieve clusters of video clips that visually match the provided descriptive words. To handle large-scale image and video data with detailed visual features, we employ the Bag of Visual Words (BoVW) approach.

We first estimate semantic segmentation masks for each frame in the scene video assets bank. This segmentation breaks down each video frame into labeled regions of interest. Following this, we utilize the Scale Invariant Feature Transform (SIFT) algorithm to extract visual features

(a) A crashed sedan object mesh generated by Houdini engine

(b) A person object mesh reconstructed by NeRF-based method.

Figure 9. Examples of generated object mesh for video simulation

from these segmented regions. We detect key points in each frame and compute descriptors represented by feature vectors for the regions containing these key points. These descriptors are then clustered, with the centroid of each cluster representing a 'visual word' in the BoVW. The frequency of these visual words across the video dataset is used to build a frequency histogram for each video. Consequently, the BoVW representation allows us to effectively retrieve matching videos based on the occurrence and frequency of the given visual words, improving the process of selecting appropriate videos for our Anything in Any Scene framework.

7.2. Object Mesh Generation

The mesh model of a target object is required before its insertion into an existing video clip. We employ the following two methods to generate the object mesh models.

Houdini Engine for Object Mesh Generation To create visually appealing and physically accurate object meshes, we utilize the Houdini Engine [48] that leverages the physics-based rendering capabilities to enhance existing object mesh models with realistic physical effects The

Houdini Engine, known as a robust 3D animation procedural tool, can produce a wide range of physical effects such as deformation, animations, reflections, and particle visual effects. As an example is shown in Figure 9a, the Houdini engine can transform a truck model into a crashed one by applying deformation effects. Furthermore, it can simulate diverse realistic physical effects, such as smoke from a crashed car, using its particle visual system. This approach is particularly critical for creating object meshes that are challenging or expensive to capture in real-world scenarios.

NeRF-based Object Mesh Reconstruction In order to also cover the objects that are difficult to produce by the Houdini engine and generalize the asset bank to include arbitrary objects, we propose the complementary NeRF-based Object Mesh Reconstruction The impressive performance of Neural Radiance Fields (NeRF) in 3D reconstruction from multi-view images offers the potential to build an extensive 3D asset bank. In our work, we adopt an off-the-shelf method [51] that combines the advantage of both NeRF and mesh representation This method reconstructs the object mesh model from multi-view RGB images. An example of the reconstructed person object mesh is shown in Figure 9b, which features rich textures and detailed geometry suitable for following rendering processes.

8. Object Placement Details

In order to accurately position the inserted object within a scene video, the first step involves reconstructing the 3D point cloud representation of the captured environment. The object placement point is then determined in 3D space, guided by segmentation mask. During the 2D-to-3D projection process, we focus on estimating an appropriate placement plane for the inserted object. This plane is conceptualized as the best-fitting plane, represented by the equation:

$$Ax + By + Cz + D = 0 \tag{6}$$

based on the selected points (x, y, z).

For a more accurate estimation, we utilize multiple 3D points to determine the optimal fitting plane. As illustrated in Figure 11, we select several points within the road region (in yellow in the Figure 11) to estimate the ground plane where the object can be realistically inserted.

Settings for PandaSet and ScanNet++ Datasets: The two datasets we used, PandaSet [58] for outdoor scene and ScanNet++[60] for indoor scene, consist of footage captured by RGB cameras and depth sensors. These sensors record driving scenes for PandaSet and room scenes for ScanNet++. Our selection process for video clips from these datasets involves choosing those captured by a forward-facing camera, particularly focusing on clips where the camera exhibits motion, thus ensuring dynamic and varied frames for composition. Regarding the ScanNet++
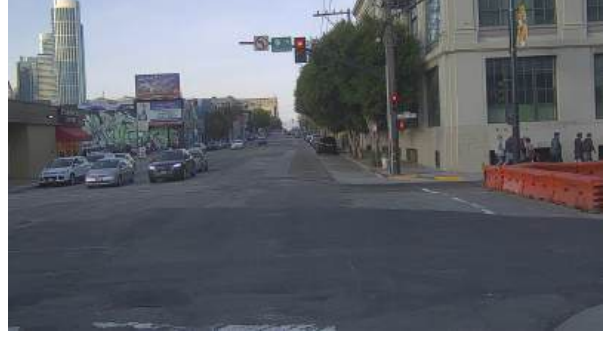

Figure 10. An example of an excluded video clip from the PandaSet. The camera is stable during the entire video clip because the camera is on a car waiting for the traffic light.
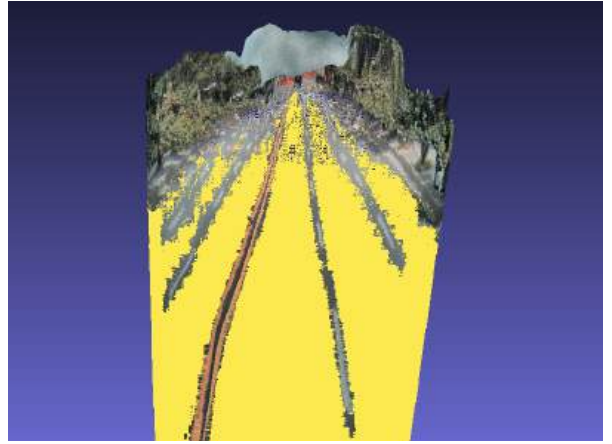

Figure 11. The projected 3D scene for object insertion. The yellow region is the plane that is available to place the inserted object.

dataset, we selected 5-second segments from each original video clip, down-sampling them from the original 60Hz to 20Hz. We ensured that a minimum of 20 frames from each clip were available, providing an adequate number of frames for effective video composition. We exclude video clips that suffer from low frame rate issues or where the camera remains stationary during the recording, such as the scenario shown in Figure10, where the camera is affixed to a stationary vehicle at a traffic signal.

Utilizing the depth information and segmentation data available in both dataset, we reconstruct the 3D point cloud for the scenes. This enables us to precisely select the object placement points from within the designated placeable areas in 3D space. For instance, in a driving scene from the PandaSet, as illustrated in Figure 11, the road region highlighted in yellow is identified as the appropriate location for inserting a car into the scene.
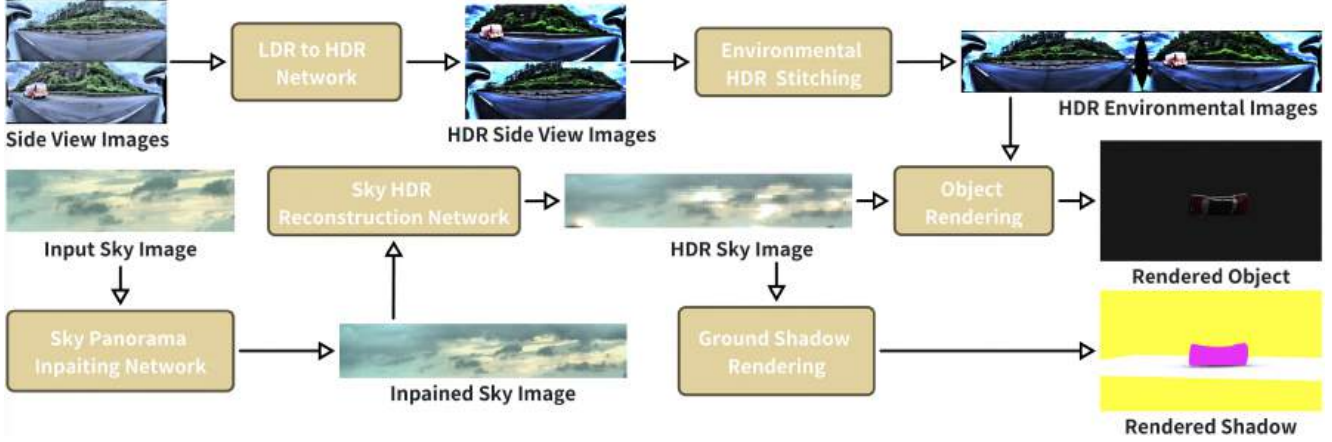
Figure 12. The overview of lighting estimation and shadow generation.

9. Lighting Estimation Detail

In Figure 12, we provide a comprehensive overview of the lighting estimation and shadow generation process. To ensure realistic shading effects on objects inserted during rendering, we estimate High Dynamic Range (HDR) images of both the sky and the surrounding environment.

For HDR sky image estimation, an image inpainting network initially infers a panoramic sky image. This is followed by employing a sky HDR reconstruction network to transform this panoramic sky image into an HDR one. In parallel, the estimation of HDR environmental images involves reconstructing HDR images from Low Dynamic Range (LDR) side-view images of the scene by using an LDR to HDR network. These images are then seamlessly stitched together to form an HDR panoramic environmental image.

Both the HDR sky and environmental images are integrated together to achieve realistic lighting effects on the inserted objects in the rendering process. Additionally, we leverage the estimated HDR sky image to render shadows for the inserted objects, utilizing the 3D graphics application Vulkan for this purpose.

10. Photorealistic Style Transfer Detail

We utilize the coarse-to-fine mechanism for photorealistic style transfer, and the overview of the network is shown in Figure 13 where both of the coarse network and refinement network consist of the dilated convolution layers. We concatenate an image with black pixels filled in the foreground region, a binary mask indicating the foreground region, and an image with black pixels filled in the background region as an input to the coarse network that outputs an initial coarse prediction. The refine network takes the composition of the coarse network's input and output, and it generates the final refined completed image.

We followed the same training strategy as described in [61], the coarse network is trained with the reconstruction loss, and the refinement network is trained with the reconstruction and GAN losses. We trained and finetuned the networks on the PandaSet dataset for the outdoor scenario. All input is concatenated together and then resized 256×256 as input image resolution.

We are also interested in the performance of different style transfer methods on the task of photorealistic style improvement in our proposed framework. Specifically, we investigated the usage of a CNN-based method DoveNet [8], a transformer-based method StyTR2 [11], and a diffusion model-based method PHDiffusion [33] compared to our method introduced in main paper.

DoveNet: a U-Net-based network is used as a generator to translate the visual domain of the inserted foreground region to match the background, and the GAN framework with two different discriminators is leveraged to train the generator for more realistic and harmonious image output. We follow the same process as described in [8], we resize the input images as 256×256 during both the training and testing stages.

StyTR2: a transformer is leveraged as an encoder to capture long-range dependencies of image features for style transfer. We set the style weight as 10.0 and content weight as 7.0 for the StyTR2 model, and we downscale the image to 512×512 and then randomly crop the image to 256 as the input image during the training stage.

PHDiffusion: a stable diffusion model is proposed to use two encoders to stylize foreground features. The features from both encoders are combined to finalize the style transfer process. We loaded the pre-trained Stable Diffusion model weights and all images are resized to 512×512 as input resolution for both training and testing.
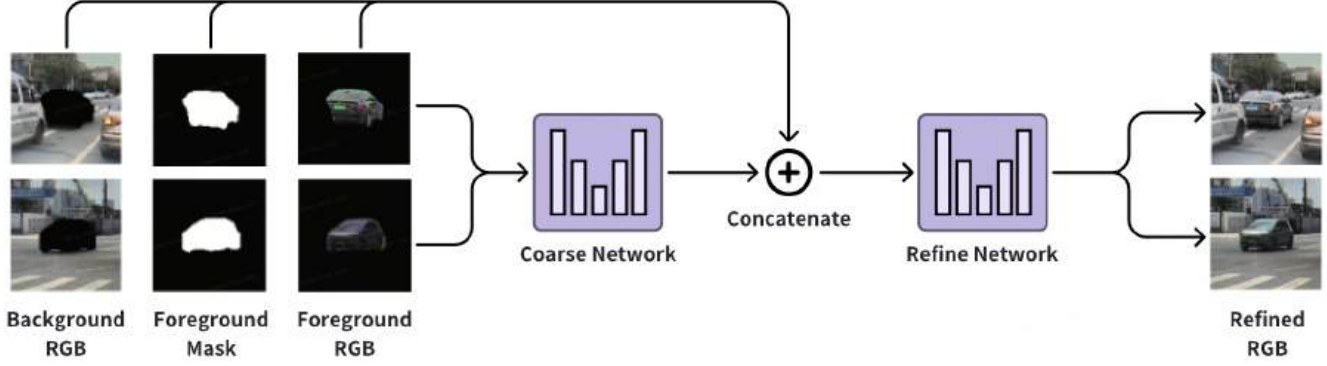
3

Figure 13. The overview of coarse-to-fine mechanism for photorealistic style transfer. The input is a background RGB image with a black foreground region, the inserted object foreground RGB image, and a foreground segmentation mask. The output is the refined RGB image.

11. Human Study Details

We validated the simulated videos generated by our proposed method through a human A/B test. We utilized a GUI application [39] designed and developed by ourselves, which allows users to compare two videos side by side, and select the preferred one between them. We provide instruction as follows to each human judge before they start to conduct the study:

You are participating in a study to assess the realism of videos created by computers. Each video features a scene with an object inserted. Your task is to compare two videos side by side and select the one that appears more realistic to you.

Please ensure that you are seated at an appropriate distance in front of the display screen, and familiarize yourself with controls, such as playing the video and going to the previous or next task.

For this study, realism is defined by how convincingly the object is integrated into the scene video. You can consider the following factors in your assessment:

1. *The consistency of the object with physical rules depicted in the scene.*
2. *The naturalness of lighting, shadows, and replications.*
3. *The believability of the object's interaction with its environment.*

Watch both videos in full at least once before making a decision, and feel free to view as many times as needed, focusing on the inserted object in the scene. Please select the video that you believe has better realism by pressing the corresponding "select" button.

The human judges use the application as shown in Figure 14, which provides multiple controls, such as navigation through all video pairs, video playback, video selection, video suspend, and selection view panel.

For the validation of each dataset, we conducted two separate human studies. The study for the outdoor dataset involved 24 human judges, while the study for the indoor

Method	Human Score(%)	FID
Proposed method	**57.92**	**10.537**
StyTR2 style transfer	53.33	11.145
PHDiffusion style transfer	36.25	12.004
DoveNet style transfer	44.58	10.832
w/o style transfer	N/A	11.901

Table 4. Experimental results of indoor Scene dataset ScanNet+ with different style transfer networks plugged into our Anything in Any Scene framework.

dataset had 16 participants. In validating the PandaSet dataset, we had a pool of 100 videos, from which 38 were randomly selected for the human study. In the case of the ScanNet++ dataset, out of the 52 available videos, 30 were randomly chosen for conducting the human study. Note that all videos were used in the calculation of the FID score. In each study, every judge was tasked with evaluating and labeling a total of 105 pairs of videos. In the first study, we compare the performance of different style transfer networks plugged into our proposed framework, covering DoveNet, StyTR2, PHDiffusion, and ours. As for another human study, we analyze the performance of our proposed method if we remove one of the key components. In both study settings, we set our proposed method without a style transfer process as the baseline.

The human score of method A can be computed as

$$\frac{\text{times of results by method A selected}}{\text{total times of results by method A and B selected}} \quad (7)$$

where method B is the baseline, and method A is the method for comparison. Suppose method A is also the base, theoretically baseline method has a human score of 50%. We provide the quantitative and qualitative results on the outdoor dataset PandaSet in the main paper, and the results on the indoor dataset ScanNet+ are detailed in Section 14.
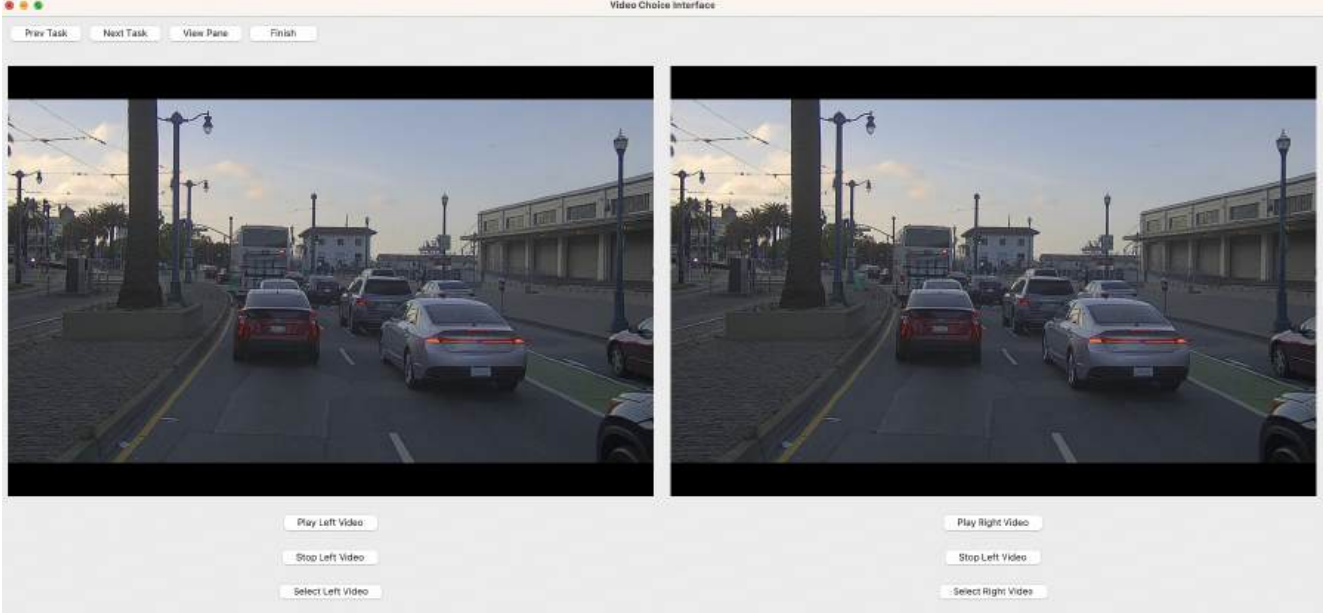
Figure 14. Example of the human study interface for comparing two videos quality. The human judge selects the right video because of its more realistic visual effect.

Method	Human Score(%)	FID
Proposed method	57.92	10.537
w/o placement	9.58	9.709
w/o HDR	32.92	10.824
w/o shadow	36.25	10.464
w/o style transfer	N/A	11.901

Table 5. Experimental results for ablation analysis of modules in our Anything in Any scene framework. Note that the baseline w/o style transfer method theoretically has a human score of 50%

12. Experimental Results of Indoor Scene

We followed the same experimental setup detailed in the main paper, and conducted a validation of our proposed method on an indoor scene. Similarly to the validation on the outdoor scene, we evaluated the performance of various style transfer networks by comparing the following methods: a CNN-based method DoveNet, transformer-based method StyTR2, diffusion model-based method PHDiffusion, and our method. In the human study, as described in 14, we designated our framework without the style transfer module as the baseline for comparison. The comparative results, summarized in Table 4, reveal that our style transfer network achieved the lowest Frechet Inception Distance (FID) score of 10.537 and the highest human score of 57.92%, surpassing the performance of the other methods.

Ablation Studies: We also performed ablation studies using indoor dataset ScanNet+ to assess the impact of individual modules on overall performance. Similarly. we re-moved one module from our framework: placement (w/o placement), HDR image reconstruction (w/o HDR), and style transfer (w/o style transfer). The results are detailed in Table 5. Similarly to the result of outdoor dataset PandaSet, the absence of HDR, and style transfer modules resulted in higher FIDs. The placement of objects in unrealistic locations significantly reduced their perceived realism among human observers. However, this decrease in realism was not accurately reflected in the FID scores. One primary reason is the nature of indoor scenes, which often have limited space. This can result in the inserted object being partially or completely out of the camera's field of view, impacting the assessment metrics. Our method consistently received a human score above 50%, while the others fell below 50%, emphasizing the contribution of each module to the efficacy of our system.

13. Downstream Task Details

We expanded our scope to include 25 object categories for insertion into images from the CODA2022 validation dataset. Similarly, we trained three models: YOLOX-S, YOLOX-L, and YOLOX-X [14]. Utilizing the "Anything in Any Scene" framework, we augmented the original training images by inserting a variety of objects, thereby generating a new set of training data. This enhanced dataset was then used to re-train the models, ensuring consistency with the original training strategy.

The performance of the models was evaluated by training on both the original and augmented datasets and then testing

Method	Data	**mAP**	Plastic Bag	Stone	Stroller	Traffic Light	Concrete Block
YOLOX-S	Original	0.321	0.302	0.020	0.218	0.193	**0.125**
	Original + Ours	**0.334**	**0.475**	**0.093**	**0.260**	**0.227**	0.108
YOLOX-L	Original	**0.394**	0.314	**0.105**	0.406	0.262	0.178
	Original + Ours	0.391	**0.336**	0.077	**0.474**	**0.318**	**0.309**
YOLOX-X	Original	0.395	**0.319**	0.110	0.307	0.246	**0.215**
	Original + Ours	**0.405**	0.311	**0.133**	**0.529**	**0.290**	0.202

Table 6. Performance of the YOLOX models trained on the original images from the CODA dataset compared to their performance when trained on a combination of original and augmented images using our Anything in Any Scene framework. We report the mAP represents the mean for all 25 object categories. We also report individual categories that has a significant improved AP in either one of the three models.

on a consistent test dataset. The results are presented in Table 6, where we detail the mean Average Precision (mAP) across all 25 object categories. We also highlight individual categories that showed significant AP improvement in any of the three models.

14. Qualitative Visualization

The quantitative experimental results show that human judges prefer our proposed method compared to the other which either has one key component missing or another photorealistic style transfer network used. We demonstrate more qualitative visualization for both outdoor dataset PandaSet and indoor dataset ScanNet+ as shown in the following.

Style Transfer Network: In Figure 15 and 17, we demonstrate the qualitative comparison of sample video frames generated by different style transfer networks using both the outdoor scene dataset PandaSet and the indoor scene dataset ScanNet+. The foreground region of images refined by DoveNet has significant grid pattern artifacts and is much blurry compared to the background regions. As for the refined images generated by StyTR2 and PHDiffusion, the color tone of the inserted object is not consistent with the weather and sunlight environment in the scene. The refined image generated by our proposed method has the best visual quality compared to the other three, and our proposed method achieved the best result in both FID and human study scores as reported in the quantitative result.

Ablation Analysis: We conducted an ablation analysis of each key rendering component including HDR image reconstruction, shadow generation, and style transfer. In Figure 16 and Figure 18, we show more qualitative comparisons of the simulated video frame with different rendering options using both the outdoor scene dataset PandaSet and the indoor scene dataset ScanNet+.

The inserted objects show inconsistent illumination a color balancing if there is no style transfer module or HDR reconstruction module involved in the video simulation process. The inserted objects with no rendered shadow seem to be off the ground.

Figure 15. Qualitative comparison of a simulated video frame using outdoor scene dataset PandaSet with different rendering options. (a) generated by the method without object placement; (b) generated by the method without HDR image reconstruction; (c) generated by the method without shadow generation (d) generated by the method without style transfer (e) generated by our proposed method including all rendering options.

Figure 16. Qualitative comparison of a simulated video frame using outdoor scene dataset PandaSet with different style transfer networks. (a) generated by DoveNet; (b) generated by StyTR2; (c) generated by PHDiffusion (d) generated by our proposed style transfer network

Figure 17. Qualitative comparison of a simulated video frame using indoor scene dataset ScanNet++ with different rendering options. (a) generated by the method without object placement; (b) generated by the method without HDR image reconstruction; (c) generated by the method without shadow generation (d) generated by the method without style transfer (e) generated by our proposed method including all rendering options.
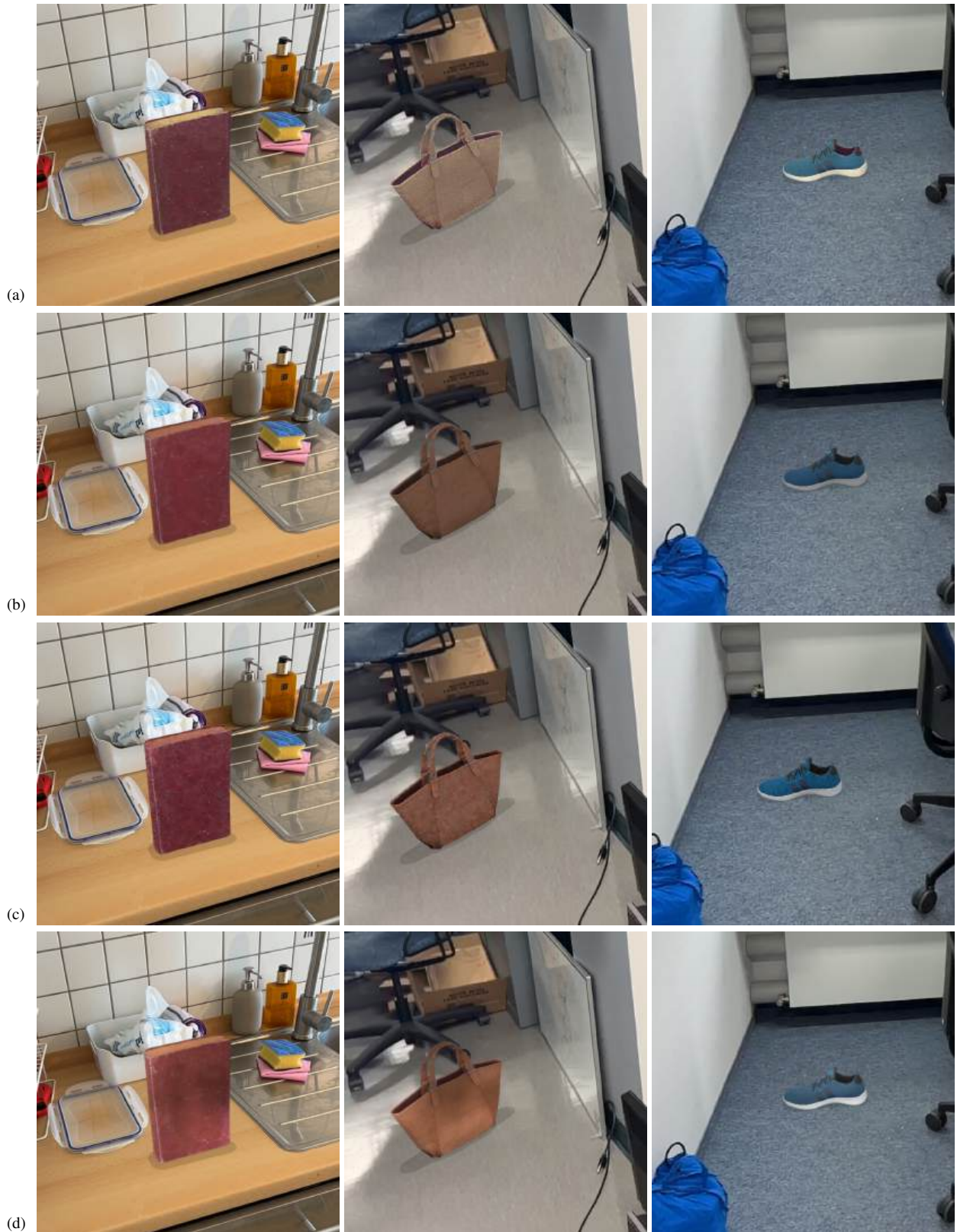
Figure 18. Qualitative comparison of a simulated video frame using indoor scene dataset ScanNet++ with different style transfer networks. (a) generated by DoveNet; (b) generated by StyTR2; (c) generated by PHDiffusion (d) generated by our proposed style transfer network