POSITIONAL ENCODING FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Transformers (DiTs) have emerged as the dominant architecture for visual generation, powering state-of-the-art image and video models. By representing images as patch tokens with positional encodings (PEs), DiTs combine Transformer scalability with spatial and temporal inductive biases. In this work, we revisit how DiTs organize visual content and discover that patch tokens exhibit a surprising degree of independence: even when PEs are perturbed, DiTs still produce globally coherent outputs, indicating that spatial coherence is primarily governed by PEs. Motivated by this finding, we introduce the Positional Encoding Field (PE-Field), which extends positional encodings from the 2D plane to a structured 3D field. PE-Field incorporates depth-aware encodings for volumetric reasoning and hierarchical encodings for fine-grained sub-patch control, enabling DiTs to model geometry directly in 3D space. Our PE-Field–augmented DiT achieves state-of-the-art performance on single-image novel view synthesis and generalizes to controllable spatial image editing.

1 INTRODUCTION

Diffusion Transformers (DiTs) (Peebles & Xie, 2023) have rapidly emerged as the dominant architecture in visual generation, forming the backbone of recent state-of-the-art image and video models such as Flux.1 Kontext (Labs et al., 2025), Qwen-Image (Wu et al., 2025a), CogVideo (Yang et al., 2024), and Wan (Wan et al., 2025). By encoding images into sequences of patch tokens and applying 2D positional encodings (PEs) (Vaswani et al., 2017), DiTs leverage the scalability of Transformers while preserving the spatial inductive biases necessary for visual synthesis. This design has enabled remarkable progress, supporting high-fidelity image generation and temporally coherent video synthesis (where additional temporal PEs are employed).

Despite their empirical success, the internal mechanisms by which DiTs organize and compose visual content remain relatively underexplored. In this work, we begin with a simple yet striking observation: patch tokens in DiTs exhibit a surprising degree of independence. When positional encodings are reassigned, the model still produces globally coherent output, though with patches reorganized according to the altered PEs. This suggests that spatial coherence in DiTs is primarily enforced by positional encodings rather than by explicit token-to-token dependencies and that manipulating PEs alone can induce structured reconfiguration of spatial content. This property offers a new avenue for spatially controllable generation, where images can be reorganized according to PEs transformation without modifying the token content itself.

Building on this insight, we focus on single-image novel view synthesis (NVS) and extend the positional encodings of DiTs beyond the 2D image plane into a structured 3D field, which we term the Positional Encoding Field (PE-Field). The PE-Field introduces two key innovations: First, we extend standard 2D RoPE (Su et al., 2024) to a 3D depth-aware encoding, embedding tokens in a volumetric field that supports reasoning across viewpoints. Second, we design a hierarchical scheme that subdivides tokens into finer sub-patch levels, allowing different sub-vectors to capture spatial information at varying granularities. Together, these designs transform DiTs into a geometry-aware generative framework that reasons directly in a 3D positional encoding field. As a result, our approach achieves state-of-the-art results in novel view synthesis (NVS) from a single image, and naturally generalizes to spatial editing tasks, where manipulating the PE-Field enables structured control of image content at both global and local levels.

Our contributions are as follows: **1)** We show that DiTs can reorganize image content purely through positional encodings, revealing a previously underexplored property that enables structured spatial editing. **2)** We introduce a depth-augmented positional encoding field that embeds tokens into a 3D space, enabling volumetric reasoning and geometric consistency. **3)** We extend DiTs with multi-level positional encodings, allowing fine-grained spatial control at sub-patch granularity. **4)** Our PE-Field–augmented DiT achieves state-of-the-art results on novel view synthesis (NVS) from a single image, and further generalizes to spatial image editing tasks.

2 Related Works

2.1 Novel view synthesis

Novel view synthesis (NVS) is a widely studied and discussed problem which can be broadly divided into two categories: methods based on multiple input images and those based on a single input image. In this work, we focus on the latter. The simplest approach is to directly use a feed-forward model (Hong et al., 2024; Jin et al., 2025) to generate novel views from an input image. Such methods typically rely on learning intermediate, general 3D representations from data. For example, early works adopt multi-plane representations (Zhou et al., 2018; Han et al., 2022; Tucker & Snavely, 2020), PixelNeRF (Yu et al., 2021) employs NeRF (Mildenhall et al., 2020) as the 3D representation, LRM (Hong et al., 2024) uses tri-plane representations, and 3D-GS (Kerbl et al., 2023) has also been adopted by methods such as PixelSplat (Charatan et al., 2024). Other methods (Wiles et al., 2020; Rombach et al., 2021; Rockwell et al., 2021; Park et al., 2024) incorporate additional results from monocular reconstruction to provide an explicit geometric structure, where warping into the target view is used which is then followed by inpainting to synthesize novel views.

Recently, with the breakthrough of diffusion-based generative models, an increasing number of works have investigated the use of diffusion models for NVS, including GeNVS (Chan et al., 2023), Zero-1-to-3 (Liu et al., 2023), ZeroNVS (Sargent et al., 2024), and CAT3D (Gao et al., 2024; Wu et al., 2025b). However, directly encoding camera pose conditions as text embeddings makes it difficult to precisely control viewpoint changes. Reconfusion (Wu et al., 2024) uses PixelNeRF (Yu et al., 2021) features as diffusion conditions, but consistency across views cannot be guaranteed. The paradigm of monocular reconstruction followed by warping and inpainting has also been adopted in diffusion-based methods (Zhang et al., 2024; Chung et al., 2023; Shriram et al., 2024; Yu et al., 2024; Cao et al., 2025), where diffusion is used for the inpainting stage. However, reprojection errors in the warped image may disrupt the semantics of the source image and are difficult to correct during inpainting. To address this issue, GenWarp (Seo et al., 2024) proposes to use warped 2D coordinates as input instead of directly warping the image, and this idea has been extended to videos in later work (Seo et al., 2025). However, since view transformation inherently occurs in 3D space, relying solely on 2D coordinates remains ambiguous, and these methods require training additional branches to handle coordinate input. Many video-based models (Sun et al., 2024; Huang et al., 2025; Chen et al., 2025; Ren et al., 2025; Zhang et al., 2025; Song et al., 2025; Liang et al., 2025) incorporate camera control to achieve NVS, but when only the target view is required, generating intermediate frames is unnecessary. CausNVS (Kong et al., 2025) also explores an autoregressive approach for novel view synthesis.

2.2 DiTs for image generation and editing

Diffusion Transformers (DiTs) were first introduced by (Peebles & Xie, 2023), who replaced the commonly used U-Net backbone in diffusion models (Rombach et al., 2022) with a pure Transformer architecture. This design leveraged the scalability and flexibility of Transformers while retaining the generative power of diffusion, and has since become the foundation of many state-of-the-art image and video generation models. Building on DiT, subsequent works such as Stable Diffusion 3 (SD3) (Esser et al., 2024), Flux.1 Kontext (Labs et al., 2025), Qwen-Image (Wu et al., 2025a), CogVideo (Yang et al., 2024), and Wan (Wan et al., 2025) have established DiT as the main backbone for large-scale generative modeling. Owing to its flexible architecture, DiT can be naturally extended by incorporating the tokens of a context image directly into the input sequence, enabling end-to-end image editing within the same generative framework. This simple yet effective strategy has been widely adopted in current mainstream editing models (Labs et al., 2025; Wu et al., 2025a), demonstrating the versatility of DiTs for controllable generation tasks. In contrast, we propose equipping DiTs with a 3D-aware hierarchical positional encoding field, enabling controllable and geometry-aware generation and editing solely through transformations on positional encodings.

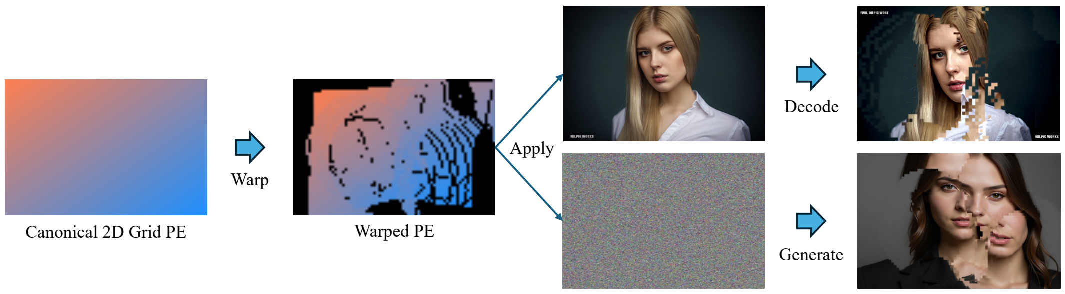

Figure 1: Illustration of DiT patch-level independence. When positional encodings (PEs) of image tokens or noise tokens are reassigned, the decoded or generated outputs still produce semantically meaningful images. The resulting structures follow the positional encoding reassignment, while boundaries between patches remain visually distinct.

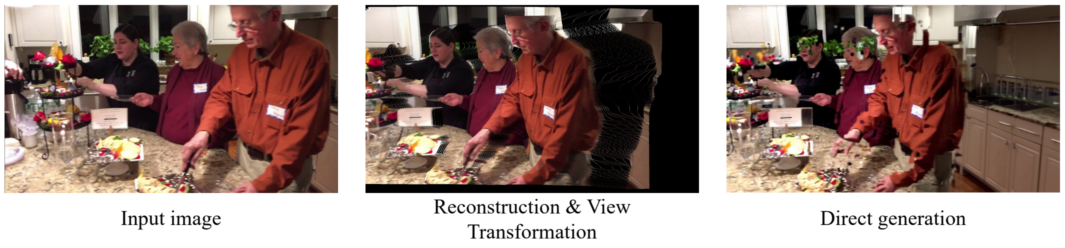

Figure 2: Illustration of our direct novel view synthesis (NVS) Results. We apply 2D positional encodings (PEs) derived from 3D reconstruction and view transformation directly to the source-view image tokens. Using these modified tokens as image conditions in DiT enables direct generation of a relatively accurate novel-view image.

3 Method

3.1 Token manipulation for view synthesis

Patch-level independence in DiT-based generative models. DiT-based architectures model image generation by patchifying the input and representing each patch as a token with a 2D positional encoding (PE). While tokens collectively reconstruct the image, we find that each token mainly encodes its local patch and retains a degree of independence. As shown in Figure 1 (Top), reassigning tokens' PEs leads to images reorganized according to the new layout, with clear patch boundaries indicating independent decoding. This independence also appears during denoising: as shown in Figure 1 (Bottom), reassigning PEs of noise tokens still yields globally coherent results (e.g., a face) but with block-wise discontinuities aligned with the modified positions. These findings suggest that global coherence is largely enforced by PEs, enabling the possibility of spatial editing by manipulating token positions through their PEs without altering token content.

Towards novel view synthesis via token manipulation. In this work, we mainly want to leverage these findings to address novel view synthesis (NVS) problem from a single image. A straightforward solution is to perform single-view 3D reconstruction followed by view transformation and inpainting, but this pipeline is often prone to errors (Seo et al., 2024). Instead, we directly manipulate DiT's image token positions: conditioned on the source reconstruction and target camera pose, we reassign positional encodings so that tokens migrate to their new projected locations. This allows recomposing image content under novel viewpoints within the DiT generative process, avoiding errors from direct image-space warping. As shown in Figure 2, this approach demonstrates a partial but effective ability to perform NVS, but artifacts remain due to: (1) resolution mismatch—positional grids from patch tokens (e.g., 16×16 pixels) are coarser than dense 3D reconstructions, limiting alignment precision. The manipulation can only rearrange image content at the patch level, but it cannot alter the content within each patch. and (2) depth ambiguity—multiple 3D points may project to the same token location. Without explicit mechanisms to disambiguate depth, generated tokens

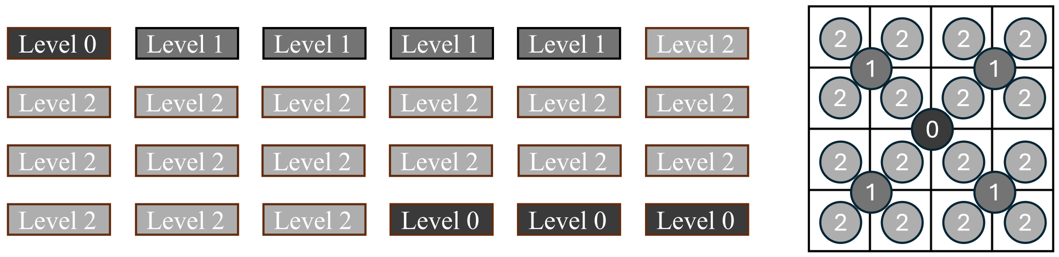

Figure 3: Illustration of hierarchical RoPE allocation in Flux (24 heads). Each rectangle on the left represents the subvector computed by one head, with colors indicating the RoPE level. Black denotes the original patch-level RoPE ($l = 0$), covers a 256 pixels patch. Level $l = 1$ corresponds to 64 pixels, and level $l = 2$ to 16 pixels. The square on the right represents a patch corresponding to one token, illustrating how different levels of positional encodings map to their respective 2D spatial locations, where $l = 2$ corresponds to a $1/16$-sized patch.

can collapse into inconsistent local structures. To adapt DiTs for NVS through positional encoding transformations, we introduce two key modifications to the existing PE design, extending it into a structured 3D field representation.

3.2 Multi-level positional encodings for sub-patch detail modeling

In the current DiT architecture, each image patch is represented as a single token, i.e., a one-dimensional vector $\mathbf{x}_i \in \mathbb{R}^d$, which is fed into the transformer for computation. Within the transformer, multi-head self-attention (MHA) is applied by projecting $\mathbf{x}_i$ into multiple subspaces (heads), $h \in \{1, \dots, H\}$ with per-head dimension d_h (typically $d_h = d/H$) enabling the model to capture diverse relationships across tokens. Current mainstream DiT models, such as Flux and SD3, first obtain queries, keys, and values by linear projections of the hidden states: $Q = XW_Q, K = XW_K, V = XW_V, X \in \mathbb{R}^{B\times T\times d}$. The results are then reshaped into H heads with per-head dimension $d_h = d/H$: $Q, K, V \in \mathbb{R}^{B\times T\times d} \to \mathbb{R}^{B\times H\times T\times d_h}$. For each head, attention is computed as $\text{head}^{(h)} = \text{softmax}\left(\frac{Q^{(h)}K^{(h)\top}}{\sqrt{d_h}}\right) V^{(h)}$. Finally, the outputs of all heads are concatenated and projected back to dimension d. However, all heads share the same positional encodings (specifically RoPEs (Su et al., 2024)), which are tied to patch-level locations. Thus, although each token is divided across multiple heads for modeling, it still encodes the holistic content of an entire patch, without explicitly capturing finer-grained details within the patch.

We argue that this design limits the transformer's ability to capture sub-patch structures that are crucial for tasks involving fine spatial transformations, such as novel view synthesis. Our goal is not to discard the different correspondences already learned by different heads at the patch level, but rather to enrich them with intra-patch detail modeling. To this end, we build directly on the head-splitting structure of MHA, augmenting it with multi-level hierarchical positional encodings so that each head's subspace captures not only patch-level information but also finer-grained details, while remaining highly compatible with the original architecture since the finer-level PEs differ little from the original ones.

Concretely, we retain a subset of heads that use the original patch-level RoPE ($l_h = 0$) to preserve the pretrained global structure, while other heads adopt finer-grained RoPEs derived from higher resolution grids (see Figure 3). At level $l_h = 0$, each positional encoding corresponds to the original patch-level RoPE (e.g., one token covers 16×16 pixels). When moving to higher levels, the positional grid resolution is increased: each step doubles the resolution along both axes, so the effective cell size shrinks by a factor of 2 per axis (i.e., by 4 in area). Let $\{\text{RoPE}^{(l_h)}\}_{l_h=0}^{M-1}$ denote the hierarchy of positional encodings, where larger l_h corresponds to higher spatial resolution (doubling per axis per level). Queries and keys in head h are rotated by the level-specific RoPE: $\mathbf{Q}_h = \text{RoPE}^{(l_h)}(Q^{(h)}), \mathbf{K}_h = \text{RoPE}^{(l_h)}(K^{(h)})$. We automatically choose the number of levels M from the total number of heads H in the pretrained architecture:

$$M = \big\lfloor \log_4(3H+1) \big\rfloor, \qquad W = \tfrac{4^M - 1}{3},$$

where W is the cumulative geometric series $1 + 4 + \cdots + 4^{M-1}$, which represents the total number of hierarchical heads that can be accommodated under the current architecture. Each head index $h \in \{1, \ldots, H\}$ maps directly to a level via the rule that exactly matches the geometric quotas $1{:}4{:}16{:}\cdots$ whose total sums to W, and falls back to the original RoPE ($l = 0$) for surplus heads:

$$l_h = \begin{cases} \big\lceil \log_4\big(3h+1\big) \big\rceil - 1, & h \le W, \\ 0, & h > W, \end{cases} \quad \text{clipped to } [0, M-1].$$

Any heads beyond the geometric budget W default to $l = 0$ to minimize disruption of pretrained patch-level priors. Taking Flux as an example, we divide each sub-vector into three levels: In Flux, there are 24 heads in total. The first head corresponds to $l = 0$, i.e., the original patch-level RoPE. Heads 2–5 are assigned to $l = 1$, and heads 6–21 to $l = 2$. The remaining heads 22–24 cannot be allocated under this scheme and are therefore reassigned back to $l = 0$. As illustrated in Figure 3, different colors indicate different PE levels. The coarsest level corresponds to a 16×16-pixel patch, while the finest level corresponds to a 4×4-pixel patch. This hierarchical design enables flexible spatial transformations: direct manipulations of sub-patch RoPE yield local geometric adjustments in the reconstruction while preserving pretrained patch-level correspondences.

3.3 Depth-aware rotary positional encoding

In standard 2D RoPE, the horizontal (x) and vertical (y) coordinates are encoded independently. Each axis is assigned a dedicated subspace of the embedding vector, within which a 1D RoPE is applied. Concretely, the token vector is partitioned into two segments, one modulated by the RoPE corresponding to the horizontal coordinate x and the other by the RoPE for the vertical coordinate y. This factorized scheme ensures that the dot product of two rotated queries and keys encodes relative displacements along both axes, while keeping the rotations invertible and dimensionally consistent.

To allow DiT to leverage positional encodings for reasoning about depth relationships between tokens that overlap in the 2D projection, following the above principle, we extend RoPE to include a third spatial axis for depth, which refers to the distance of each pixel's corresponding 3D point from the camera along the optical axis (that is, its z coordinate in the camera coordinate system). In addition to the subspaces for (x, y), we introduce another subspace for the depth z. Each coordinate (x, y, z) thus has its own 1D RoPE encoding, applied to a disjoint part of the embedding vector:

$$\mathbf{Q}^{(h)} = \big[\, \mathrm{RoPE}_x^{(l_h)}(\mathbf{Q}_x^{(h)}),\ \mathrm{RoPE}_y^{(l_h)}(\mathbf{Q}_y^{(h)}),\ \mathrm{RoPE}_z^{(l_h)}(\mathbf{Q}_z^{(h)}) \,\big],$$
$$\mathbf{K}^{(h)} = \big[\, \mathrm{RoPE}_x^{(l_h)}(\mathbf{K}_x^{(h)}),\ \mathrm{RoPE}_y^{(l_h)}(\mathbf{K}_y^{(h)}),\ \mathrm{RoPE}_z^{(l_h)}(\mathbf{K}_z^{(h)}) \,\big],$$

where $\mathbf{Q}_x^{(h)}, \mathbf{Q}_y^{(h)}, \mathbf{Q}_z^{(h)}$ (and $\mathbf{K}_x^{(h)}, \mathbf{K}_y^{(h)}, \mathbf{K}_z^{(h)}$) denote the corresponding vector segments allocated to each axis. This extension yields a 3D spatial RoPE that encodes relative offsets not only in the image plane but also along the depth axis, enabling the Transformer to model volumetric correspondences and maintain geometric consistency across viewpoints.

3.4 Overall architecture and training objective

These two components together form a new 3D field–based positional encoding, which we apply to the DiT architecture to jointly process noise tokens and source-view image tokens, resulting in our NVS-DiT model. As illustrated in Figure 4, noise tokens are placed on a regular 2D grid with depth initialized to zero, while source-view image tokens are projected into the target camera view via monocular reconstruction and view transformation. Each image token is assigned a hierarchical 3D positional encoding (x, y, z) that captures its detailed target spatial location and depth. Tokens projected outside the valid grid are discarded, and empty positions are filled with noise tokens, which are progressively refined by the transformer to generate geometrically consistent content. This design enables the model to integrate observed image evidence with generative completion, achieving novel view synthesis within the DiT framework.

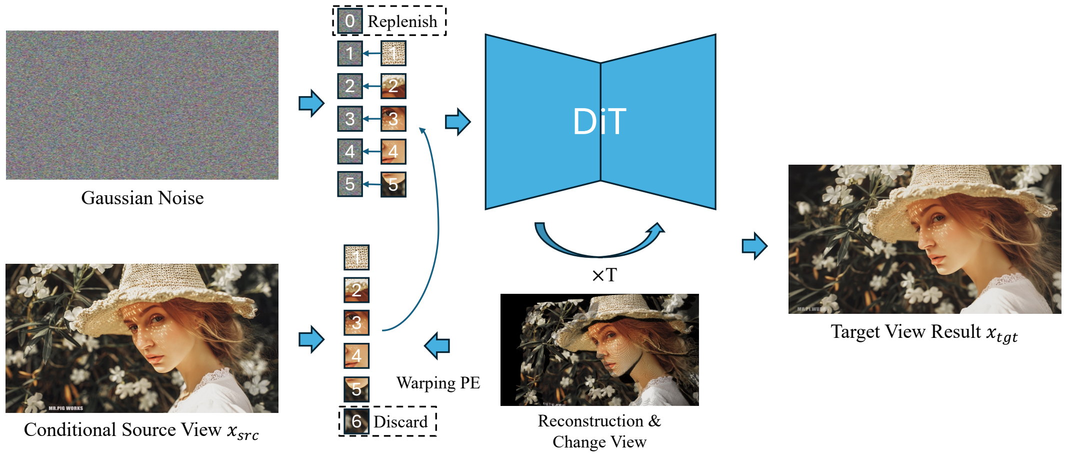
0 Replenish
1
2
3
4
5
Gaussian Noise
DiT
×T
Target View Result x_{tgt}
1
2
3
4
5
6 Discard
Warping PE
Conditional Source View x_{src}
Reconstruction & Change View

Figure 4: The transformer takes both noise tokens and source-view image tokens. Noise tokens are placed on a 2D grid with depth set to zero, while image tokens are assigned hierarchical PEs according to their projected positions from monocular reconstruction and view transformation, with depth values taken from the reconstruction. Tokens projected outside the grid (e.g., index 6) are discarded, and empty grid locations without image tokens (e.g., index 0) are filled by noise, which is refined to generate plausible content.

To train the model, we leverage multi-view supervision under a rectified-flow (Liu et al., 2022) objective. Specifically, we adopt the rectified flow–matching loss:

$$\mathcal{L}_\theta = \mathbb{E}_{t\sim p(t),\, x_{tgt}, x_{src}^{trans-PE}}\left[\left\|v_\theta(z_t, t, x_{src}^{trans-PE}) - (\varepsilon - x_{tgt})\right\|_2^2\right],$$

where $x_{src}^{trans-PE}$ and x_{tgt} denote the image tokens of the source view with transformed PEs and the target view, respectively, obtained by the corresponding DiT's VAE encoder. z_t is the linearly interpolated latent between clean latent x_{tgt} and Gaussian noise $\varepsilon \sim \mathcal{N}(0, 1)$, defined as $z_t = (1-t)x_{tgt} + t\varepsilon$.

4 Experiments

4.1 Implementation details

Our model is built on Flux.1 Kontext (Labs et al., 2025), which generates images conditioned jointly on a text prompt and a reference image. This architecture naturally aligns with our design, as it already integrates reference-image tokens, providing a seamless foundation for incorporating our PE-Field framework. We remove its text input and condition solely on the reference image. To train our NVS model, we use two multi-view datasets, DL3DV (Ling et al., 2024) and MannequinChallenge (Li et al., 2020), both processed with VGGT (Wang et al., 2025) to obtain per-image depth maps and corresponding camera poses. Additional details are provided in the **Appendix**.

4.2 Comparisons with relevant methods

We mainly compare our approach with several baseline methods (listed in Table 1) in the single-image novel view synthesis setting. Experiments are conducted on three datasets, Tanks-and-Temples (Knapitsch et al., 2017), RE10K (Zhou et al., 2018), and DL3DV (Ling et al., 2024). In each case, a single input image is provided, and subsequent frames are generated under different target viewpoints. For methods that require depth or point cloud as conditional input, we uniformly use the predictions obtained from VGGT as input. We then calculated three metrics, PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and reported the average scores for all test samples in Table 1. Our method outperforms existing approaches across all metrics on all three datasets. Qualitative comparison with a subset of representative methods is presented in Figure 5.

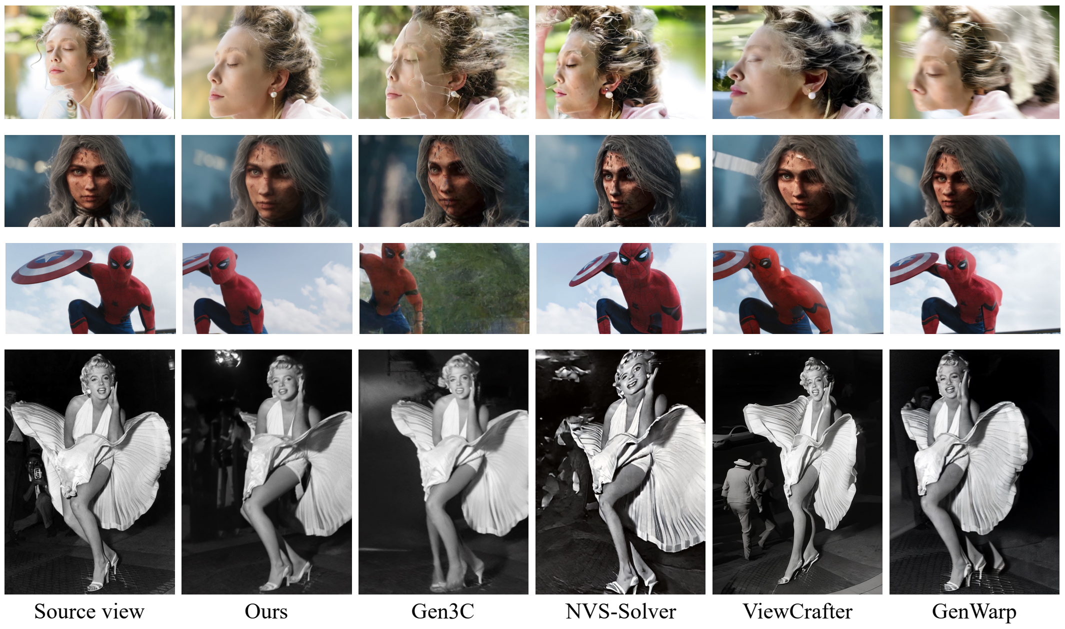

Figure 5: Visualization of novel view synthesis results where the source image (left) is rotated 30° to the right. Compared with other methods, our approach achieves accurate viewpoint transformation while preserving consistency with the source image and avoiding noticeable artifacts.

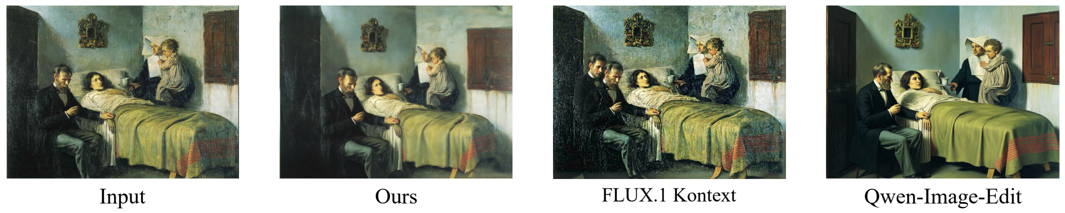

Figure 6: Comparison with prompt-based image editing methods. Our approach enables accurate control of rotation angles while maintaining consistency with the input image.

We observe that GEN3C often propagates reconstruction artifacts into the final results, leading to noticeable white streaks and irregular boundaries. NVS-Solver and ViewCrafter tend to introduce depth-warping errors, which negatively affect the geometric accuracy of the synthesized novel views. GenWarp produces unsatisfactory results due to the absence of depth information in its coordinate representation and the misalignment between its coordinate system and the input image. Due to space limitations, **more qualitative comparisons are provided in the Appendix**. It is worth noting that, unlike many video-based models listed here, our approach does not require generating intermediate frames between viewpoints, making it over an order of magnitude faster than video-based method to generate target view while still producing geometrically consistent results.

Beyond pose-conditioned approaches, recent image editing models such as Flux.1 Kontext (Labs et al., 2025) and Qwen-Image-Edit (Wu et al., 2025a) also demonstrate strong capabilities in viewpoint manipulation. We further compare our method with these prompt-based editing results, as illustrated in Figure 6. Flux is generally insensitive to prompts specifying spatial viewpoint changes, often producing only minor viewpoint variations while introducing noticeable artifacts. Qwen, on the other hand, achieves more pronounced spatial editing effects than Flux, but tends to alter the original image tokens. As shown in the rightmost example of Figure 6, the result appears overly smoothed and even alters the person's identity. Overall, it remains very challenging to precisely control viewpoint changes through prompts. More comparisons can be found in the **Appendix**.

Method	Tanks-and-Temples			RE10K			DL3DV		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ZeroNVS (Sargent et al., 2024)	13.14	0.327	0.516	15.23	0.540	0.386	14.17	0.441	0.481
CameraCtrl (He et al., 2024)	15.34	0.534	0.331	17.74	0.681	0.278	16.31	0.552	0.352
GenWarp (Seo et al., 2024)	16.45	0.513	0.377	15.30	0.538	0.371	15.81	0.531	0.382
NVS-Solver (You et al., 2024)	16.73	0.521	0.323	17.00	0.673	0.314	16.86	0.543	0.341
ViewCrafter (Yu et al., 2024)	17.18	0.589	0.346	17.75	0.681	0.315	17.24	0.571	0.329
DimensionX (Sun et al., 2024)	17.78	0.635	0.228	18.21	0.717	0.307	18.22	0.653	0.201
SEVA (Zhou et al., 2025)	17.61	0.621	0.235	17.58	0.688	0.334	18.01	0.638	0.214
MVGenMaster (Cao et al., 2025)	18.03	0.622	0.253	17.87	0.701	0.321	17.71	0.586	0.277
See3D (Ma et al., 2025)	18.35	0.641	0.244	18.24	0.735	0.293	18.41	0.631	0.215
Voyager (Huang et al., 2025)	18.61	0.669	0.238	18.56	0.723	0.264	18.84	0.636	0.227
FlexWorld (Chen et al., 2025)	18.91	0.675	0.236	18.03	0.691	0.282	18.67	0.645	0.218
GEN3C (Ren et al., 2025)	19.18	0.681	0.207	20.64	0.754	0.229	19.14	0.658	0.198
Original PE	20.03	0.683	0.221	20.17	0.752	0.233	19.92	0.667	0.201
w/o Depth	20.63	0.692	0.217	20.33	0.767	0.227	20.46	0.695	0.194
w/o Multi-Level	21.97	0.718	0.180	21.42	0.809	0.168	21.91	0.733	0.162
Ours	**22.12**	**0.732**	**0.174**	**21.65**	**0.816**	**0.162**	**22.23**	**0.742**	**0.154**

Table 1: Quantitative comparison of different methods on Tanks-and-Temples, RE10K, and DL3DV datasets. We report the average PSNR, SSIM, and LPIPS scores for novel view synthesis from a single input image.

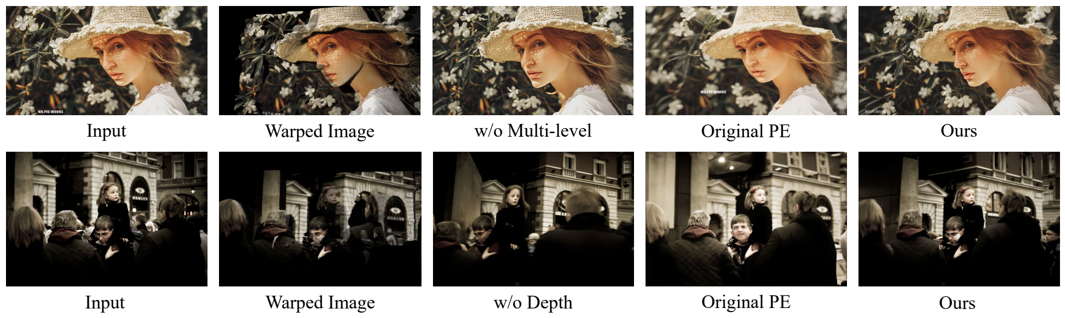

Figure 7: Ablation studies. Removing the detailed positional encoding or depth leads to different types of degradation in the generated results.

4.3 Ablation studies

We mainly analyze the effect of removing our two key components: the hierarchical detailed positional encodings and the additional depth-aware extension. The quantitative impact of removing each component can be observed in Table 1, while Figure 7 provides two illustrative cases. As shown in the top example of Figure 7, when the multi-level positional encoding (particularly the detailed level) is removed, undesirable distortions appear due to the mismatch between patch-level positional encodings and the reconstruction. When depth information is removed (see bottom example in Figure 7), the generated images suffer from severe spatial misalignment.

When applying our method to generate results under large viewpoint changes, the model is required to directly generate a substantial amount of unseen content, which increases the generation burden and may compromise consistency with the source image. To mitigate this issue, we decompose the transformation into multiple steps, in which the model only needs to complete a small portion of the missing content in each step. As shown in Figure 8, we divide the transformation of the target viewpoint into five steps. After each step, the newly generated content is fused back into the image tokens of the original viewpoint, and the fused tokens (or point cloud) are then transformed to the next intermediate viewpoint for subsequent generation. Compared to directly transforming to the target viewpoint in one step (rightmost result in Figure 8), this progressive strategy produces results that are more consistent with the source view. See **Appendix** for **quantitative** comparisons.

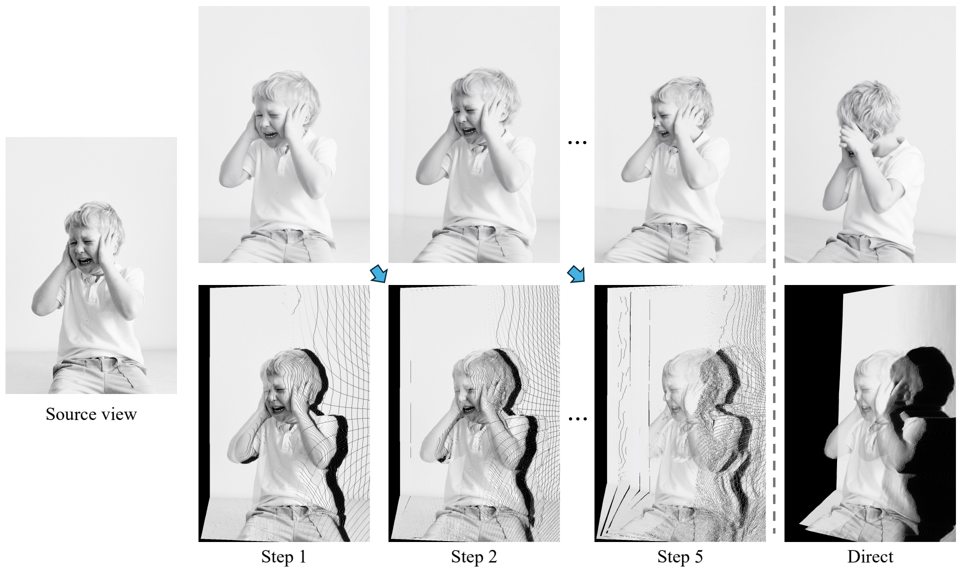

Figure 8: Multi-step generation. Left: input image. Top: generated results. Bottom: rotated point clouds. Right: direct one-step generation.

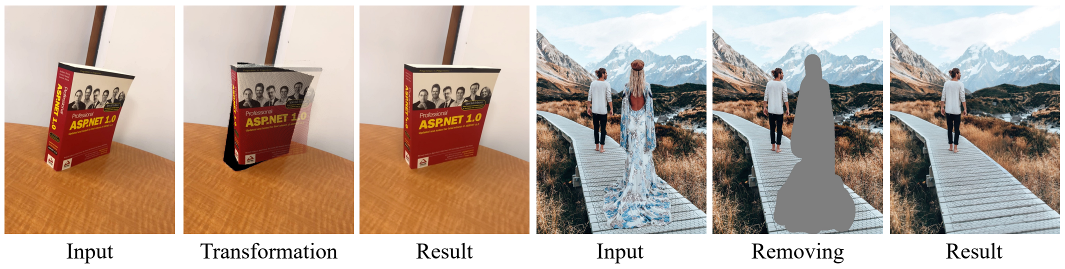

Figure 9: Applications. The left example shows object 3D editing, while the right example shows object removal, highlighting the versatility of our model in different spatial editing tasks.

4.4 Other Applications

After training, our NVS model acquires the ability to reason over visual tokens in 3D space and generate consistent content. Consequently, it can naturally adapt to other tasks with similar spatial logic, even in the **absence of task-specific training**. As illustrated in Figure 9, in the left example we perform object-level 3D editing by isolating the point cloud of the book, rotating it to a new viewpoint, and recomposing it with the original background. In the right example, we achieve object removal by discarding the tokens corresponding to the masked human region and replenishing them with noise, resulting in a realistic removal effect. More results can be found in the **Appendix**.

5 Conclusions

In this work, we revisited the internal mechanisms of Diffusion Transformers and revealed that spatial coherence is largely governed by positional encodings rather than explicit token interactions. Building on this observation, we introduced the Positional Encoding Field (PE-Field), which extends standard 2D encodings into a 3D, depth-aware and hierarchical framework. This design equips DiTs with geometry-aware generative capabilities, achieving state-of-the-art results on single-image novel view synthesis while also enabling flexible and controllable spatial image editing. We hope our study sheds light on the overlooked role of positional encodings and inspires future research into more principled and spatially grounded generative architectures.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

Chenjie Cao, Chaohui Yu, Shang Liu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Mvgenmaster: Scaling multi-view generation from any image via 3d priors enhanced diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6045–6056, 2025.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *ICCV*, 2023.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024.

Luxi Chen, Zihan Zhou, Min Zhao, Yikai Wang, Ge Zhang, Wenhao Huang, Hao Sun, Ji-Rong Wen, and Chongxuan Li. Flexworld: Progressively expanding 3d scenes for flexiable-view synthesis. *arXiv preprint arXiv:2503.13265*, 2025.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.

Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *SIGGRAPH Conference*, 2022.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson WH Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *arXiv preprint arXiv:2506.04225*, 2025.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=QQBPWtvtcn`.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Xin Kong, Daniel Watson, Yannick Strümpler, Michael Niemeyer, and Federico Tombari. Causnvs: Autoregressive multi-view diffusion for flexible 3d novel view synthesis. *arXiv preprint arXiv:2509.06579*, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`.

Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Mannequinchallenge: Learning the depths of moving people by watching frozen people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4229–4241, 2020.

Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 798–810, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2016–2029, 2025.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Byeongjun Park, Hyojun Go, and Changick Kim. Bridging implicit and explicit geometric transformation for single-image view synthesis. *IEEE TPAMI*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Pexels. Pexels — Free Stock Videos and Photos. `https://www.pexels.com/`.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6121–6132, 2025.

Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *ICCV*, 2021.

Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *ICCV*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. In *CVPR*, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. *Advances in Neural Information Processing Systems*, 37:80220–80243, 2024.

Junyoung Seo, Jisang Han, Jaewoo Jung, Siyoon Jin, Joungbin Lee, Takuya Narihira, Kazumi Fukuda, Takashi Shibuya, Donghoon Ahn, Shoukang Hu, et al. Vid-camedit: Video camera trajectory editing with generative rendering from estimated geometry. *arXiv preprint arXiv:2506.13697*, 2025.

Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199*, 2024.

Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. Worldforge: Unlocking emergent 3d/4d generation in video diffusion model via training-free guidance. *arXiv preprint arXiv:2509.15130*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv preprint arXiv:2411.04928*, 2024.

Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26057–26068, 2025b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. *arXiv preprint arXiv:2405.15364*, 2024.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE TVCG*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Songchun Zhang, Huiyao Xu, Sitong Guo, Zhongwei Xie, Hujun Bao, Weiwei Xu, and Changqing Zou. Spatialcrafter: Unleashing the imagination of video diffusion models for scene reconstruction from limited observations. *arXiv preprint arXiv:2505.11992*, 2025.

Jensen Jinghao Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM TOG*, 2018.

A Appendix

A.1 Implementation Details

To better adapt the Flux transformer to our PE-Field design, we fine-tune all Flux transformer parameters with a learning rate of 2×10^{-5}. All training images are resized to approximately 1024×1024, the default resolution supported by Flux. Experiments are conducted on 8 H100 GPUs with a batch size of 1 for about 15,000 training steps. Regarding the allocation of embedding dimensions across the three coordinate axes, Flux assigns a 128-dimensional vector to each head. We allocate 56 dimensions to both the x and y axes, and 16 dimensions to the z axis. This design minimally modifies the original Flux structure, which also allocates 56 dimensions each for x and y, plus an additional 16 binary dimensions to distinguish noise and reference-image tokens. For multi-level positional encodings, we obtain hierarchical features by bilinearly downsampling the original image grid by factors of 4, 8, and 16 along both axes. For the source-view image, the final positional encodings are constructed as follows: a point map is generated from VGGT-predicted depth and camera parameters, transformed into the target viewpoint pointmap, and then directly downsampled to serve as the input positional encodings. Following previous work (Liang et al., 2025; Chen et al., 2025), we sample 100, 300, and 300 videos from T&T, RE10K, and DL3DV, respectively, for evaluation.

To enable object-level 3D editing (Figure 9), we first detect the target object using GroundingDINO (Liu et al., 2024) and obtain its mask with SAM (Kirillov et al., 2023). The mask is then used to extract the object's point cloud. We perform viewpoint transformation within the object's coordinate system and place the transformed object back into the corresponding location of the original point map. Finally, this warped point map is used as the positional encoding input.

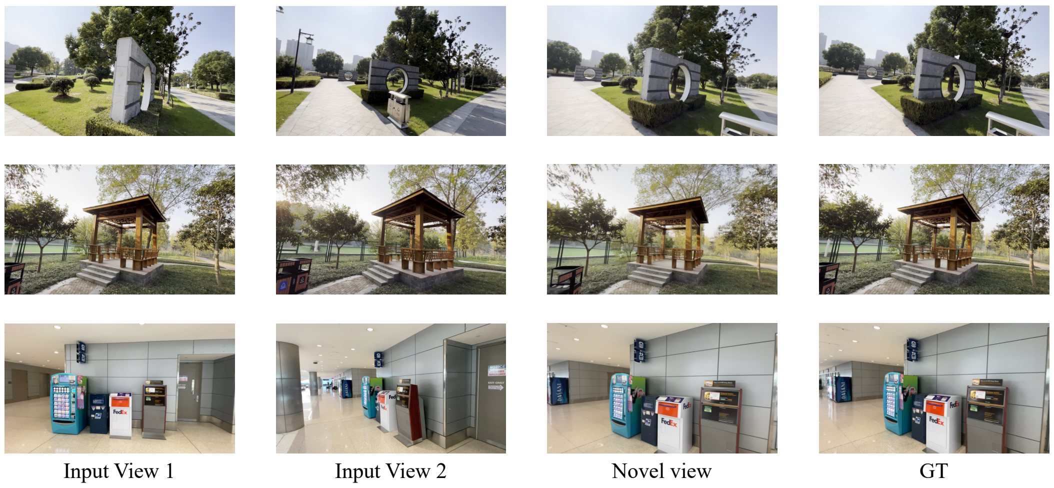

Figure 10: The results show how the depth and camera poses predicted by VGGT from the two left images are used to warp their image tokens into a shared intermediate target view, followed by synthesizing a novel view from these aligned tokens.

A.2 Additional Qualitative Results

In this appendix, we provide additional qualitative comparisons and visualizations. Figures 16 to 41 include comparisons between our approach and the baseline methods listed in Table 1. Figures 42 to 47 present comparisons with recent prompt-based image editing models, Flux.1 Kontext and Qwen-Image-Edit. Figures 48 and 49 showcase our results on 3D-aware object editing, while Figures 50 and 51 illustrate our method applied to object removal. The images in Figures 16 and 17 are sourced from the DL3DV dataset, Figures 48 and 49 are sourced from the Objectron dataset (Ahmadyan et al., 2021), while the remaining examples are collected from in-the-wild images, primarily originating from the (Pexels) website and the LAION (Schuhmann et al., 2022) dataset. Due to space limitations, some images here are compressed; **please refer to the supplementary material for the uncompressed versions.**

A.3 Quantitative Comparisons

As discussed in the main text, gradually transforming to the target viewpoint through multiple steps can improve the quality of the generated results. Here we report the quantitative effects of varying the number of steps, as shown in Table 2. The results indicate that splitting the transformation into up to 5 steps consistently improves the metrics, while further increasing the number of steps yields only marginal gains.

To further demonstrate the effectiveness of our method, we additionally report Fréchet Inception Distance (FID) (Heusel et al., 2017) to assess the quality of the synthesized images, and Rotation error (R_{err}) and Translation error (T_{err}) to evaluate the accuracy of the novel-view poses. The computation of these metrics follows prior works (Liang et al., 2025; Chen et al., 2025; Yu et al., 2024), except that here the poses are directly estimated using VGGT. These results are shown in Table 3.

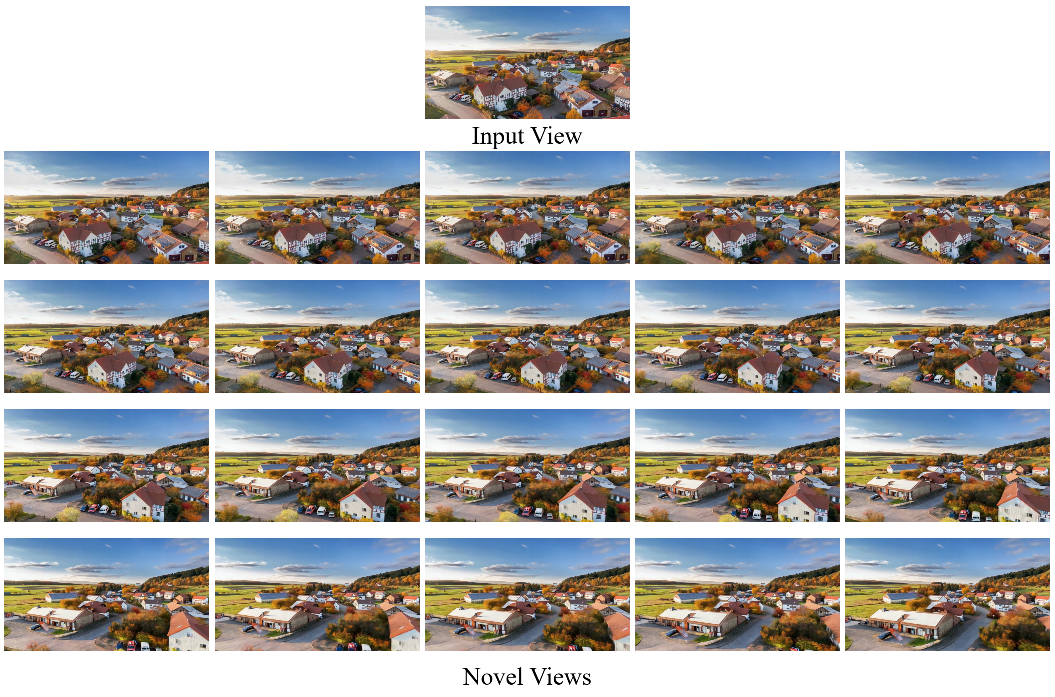

Figure 11: Visualization of a sequence of consecutive frames generated from a single input view.

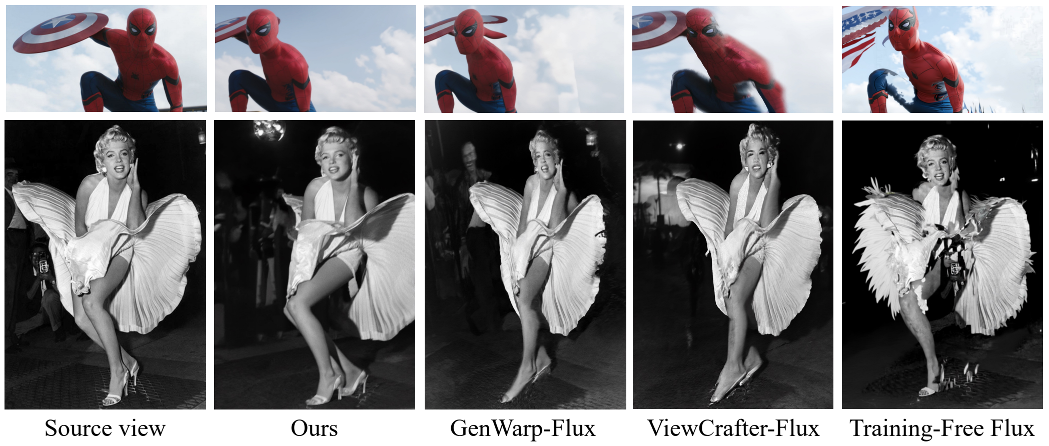

Figure 12: Comparison with several Flux-based baselines.

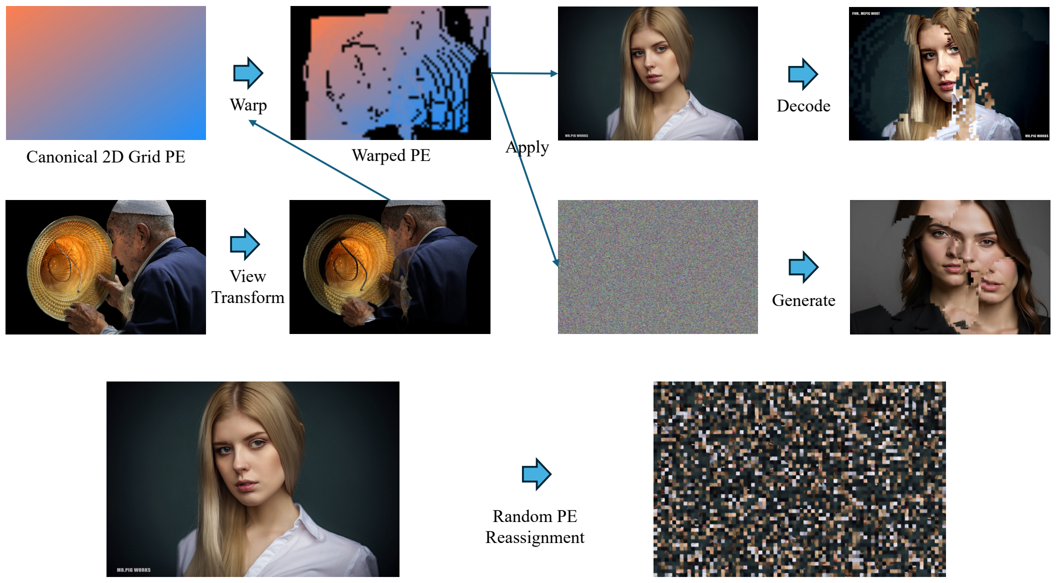

Figure 13: Interpretation of Figure 1 in the main paper.

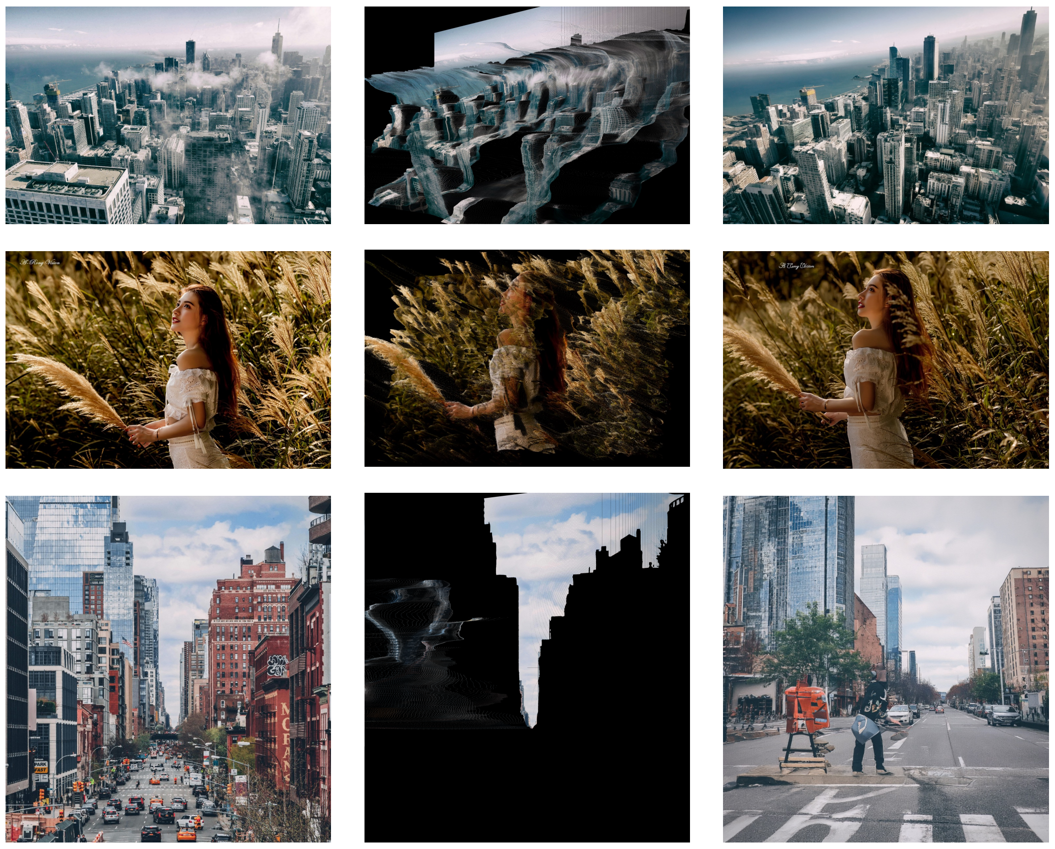

Figure 14: Across various noisy monocular 3D reconstruction scenarios, our model consistently produces robust outputs.

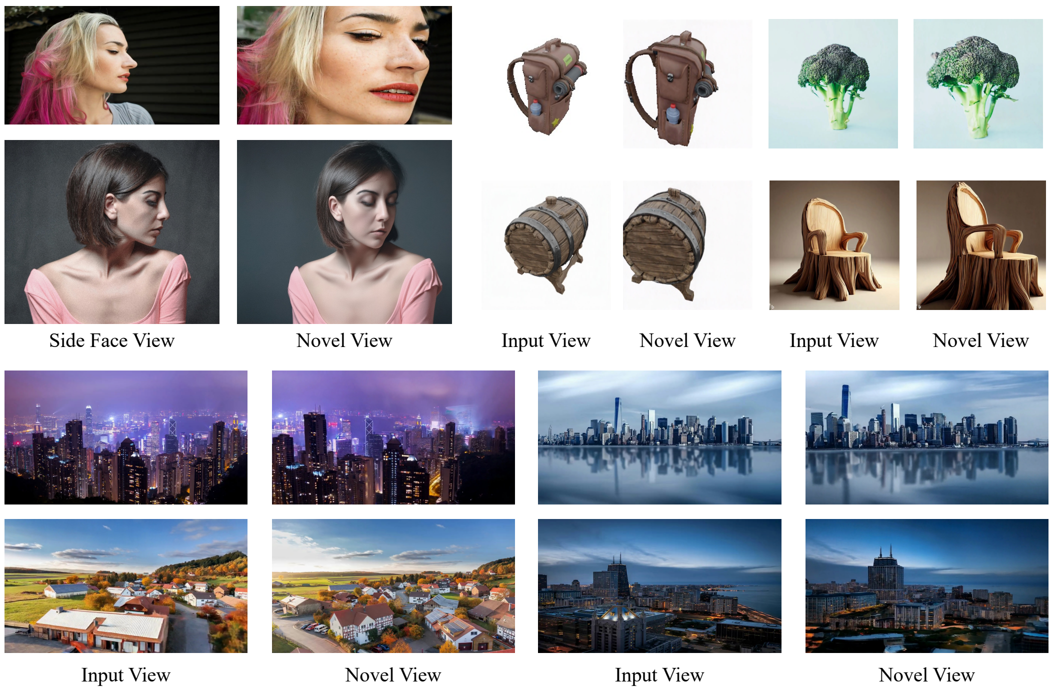

Figure 15: Novel-view synthesis results for three specific test conditions.

Method	Tanks-and-Temples			RE10K			DL3DV		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Ours 1 step	22.12	0.732	0.174	21.65	0.816	0.162	22.23	0.742	0.154
Ours 3 steps	22.23	0.734	0.171	22.04	0.818	0.160	22.65	0.743	0.153
Ours 5 steps	**22.51**	0.737	0.170	22.31	0.821	0.157	22.84	0.745	**0.151**
Ours 10 steps	22.46	**0.738**	**0.169**	**22.35**	**0.822**	**0.156**	**22.90**	**0.748**	0.152

Table 2: Quantitative evaluation of different generation steps on Tanks-and-Temples, RE10K, and DL3DV datasets. We report the average PSNR, SSIM, and LPIPS scores for novel view synthesis from a single input image.

Method	Tanks-and-Temples			RE10K			DL3DV		
	FID↓	R_{err} ↓	T_{err} ↓	FID↓	R_{err} ↓	T_{err} ↓	FID↓	R_{err} ↓	T_{err} ↓
ZeroNVS (Sargent et al., 2024)	39.532	0.907	1.751	25.888	0.162	0.263	32.558	0.588	1.405
CameraCtrl (He et al., 2024)	34.128	0.950	1.726	25.539	0.149	0.224	29.463	0.532	1.285
GenWarp (Seo et al., 2024)	31.663	0.832	1.498	23.633	0.161	0.245	23.588	0.419	0.992
NVS-Solver (You et al., 2024)	35.699	1.022	1.395	29.250	0.190	0.363	27.271	0.483	1.187
ViewCrafter (Yu et al., 2024)	26.067	0.144	0.356	24.088	0.063	0.172	23.409	0.105	0.275
DimensionX (Sun et al., 2024)	24.131	0.126	0.315	23.185	0.071	0.149	22.233	0.099	0.253
SEVA (Zhou et al., 2025)	26.263	0.114	0.280	27.322	0.067	0.176	26.526	0.142	0.301
MVGenMaster (Cao et al., 2025)	28.159	0.136	0.339	24.378	0.060	0.144	25.894	0.107	0.277
See3D (Ma et al., 2025)	27.134	0.086	0.240	27.775	0.054	0.144	24.215	0.166	0.318
Voyager (Huang et al., 2025)	22.903	0.074	0.194	17.456	0.050	0.156	20.964	0.083	0.155
FlexWorld (Chen et al., 2025)	22.194	0.108	0.201	18.519	0.062	0.129	20.408	0.069	0.147
GEN3C (Ren et al., 2025)	21.978	0.054	0.168	16.063	0.030	0.111	21.426	0.095	0.142
Ours	**18.418**	**0.040**	**0.131**	**14.778**	**0.023**	**0.076**	**17.775**	**0.063**	**0.110**

Table 3: Quantitative comparison of different methods on Tanks-and-Temples, RE10K, and DL3DV datasets. We report the average FID, R_{err}, and T_{err} for novel view synthesis.

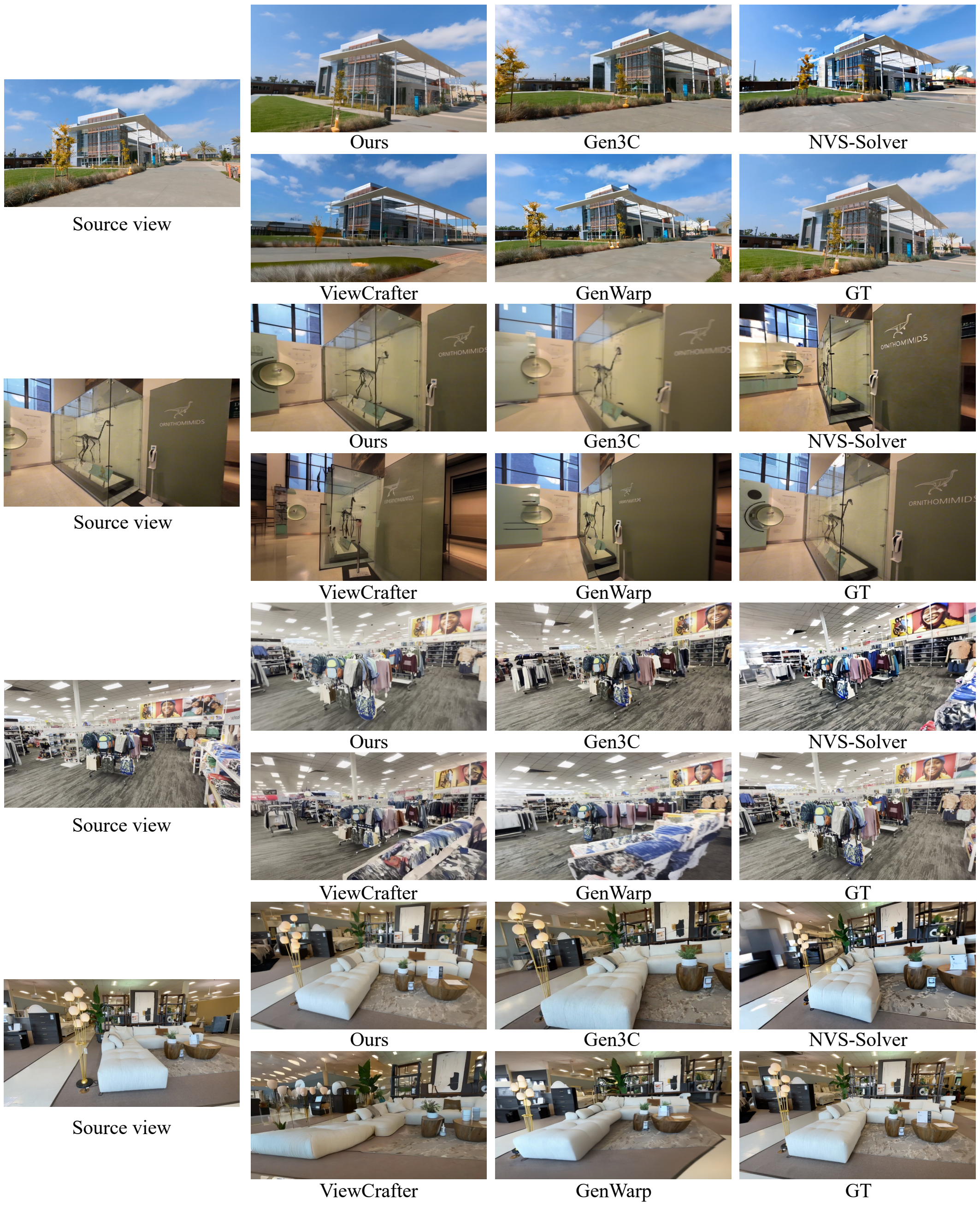

Figure 16: Supplementary novel view synthesis (NVS) examples on DL3DV dataset.

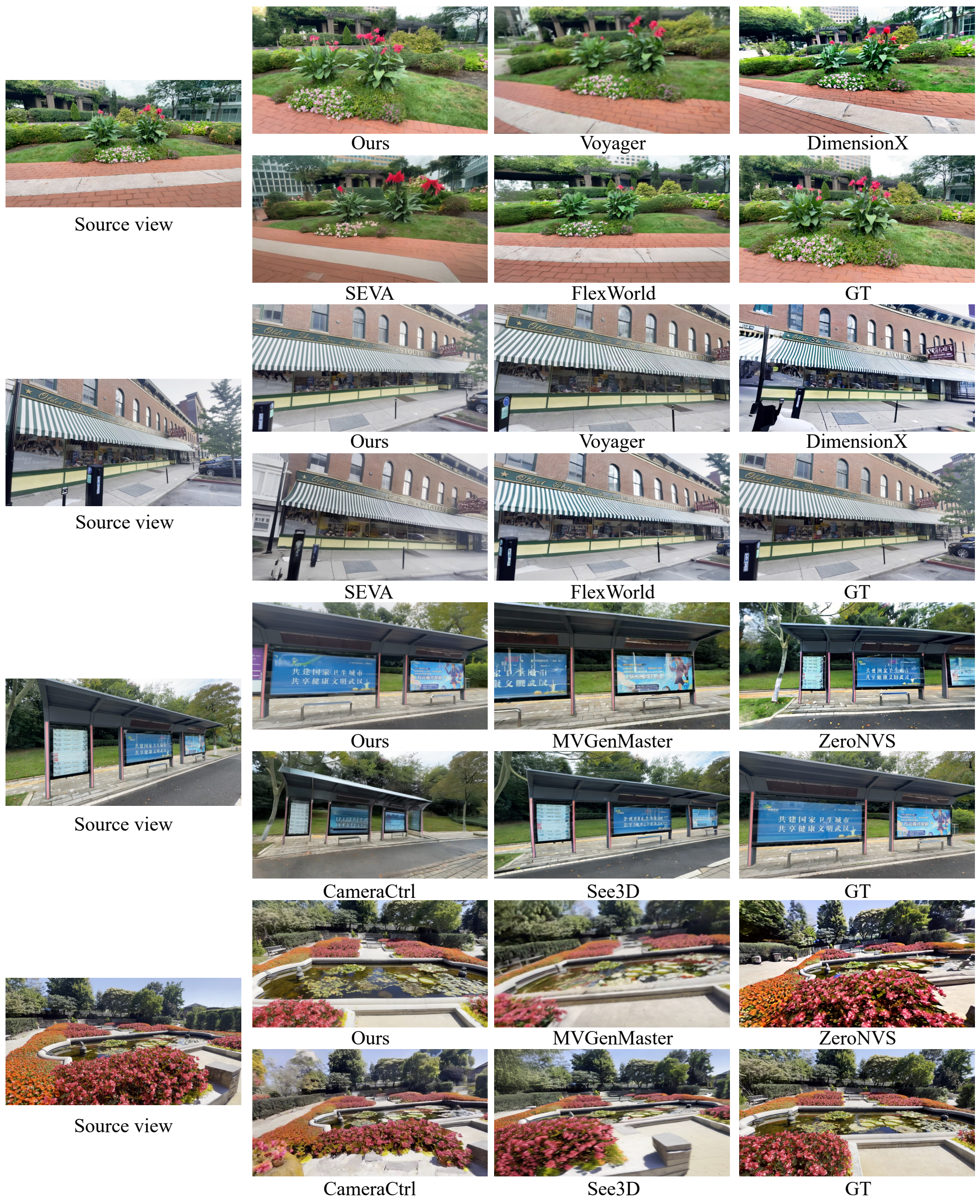

Figure 17: Supplementary novel view synthesis (NVS) examples on DL3DV dataset.

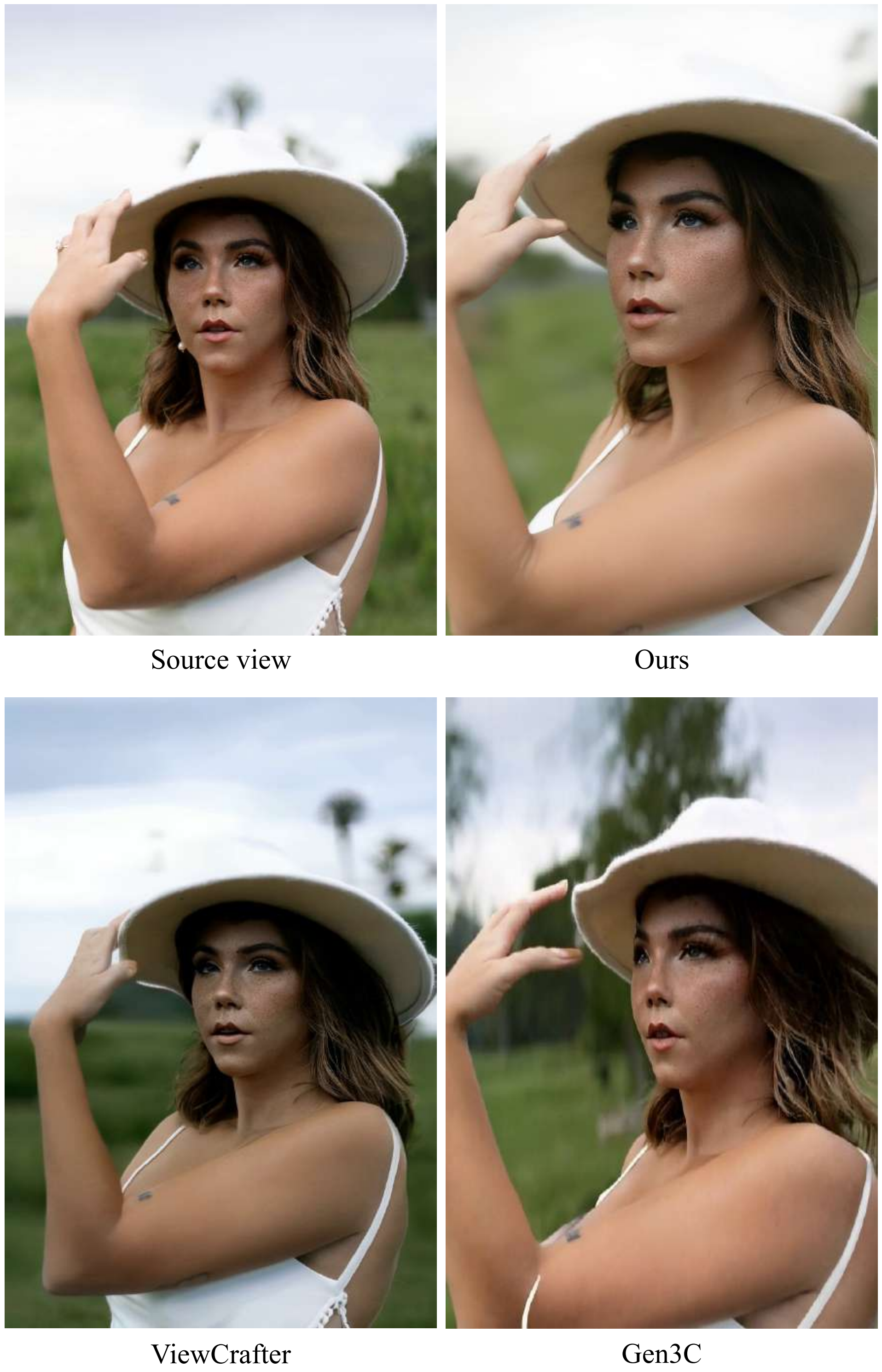

Figure 18: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, ViewCrafter produces only a small rotation with hand distortions and skin color changes, GEN3C introduces facial distortions.

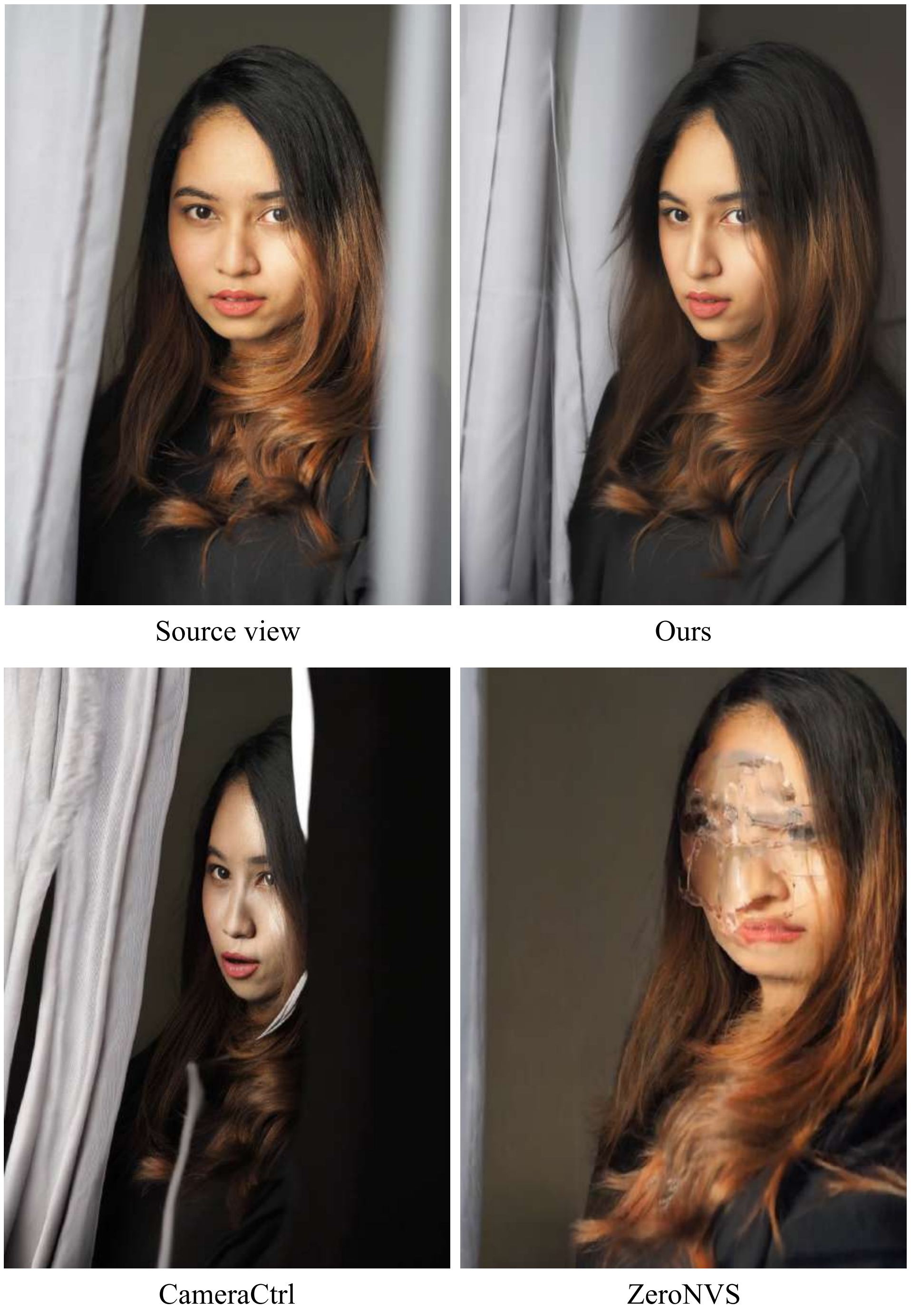

Figure 19: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, CameraCtrl and ZeroNVS both cause facial distortions.

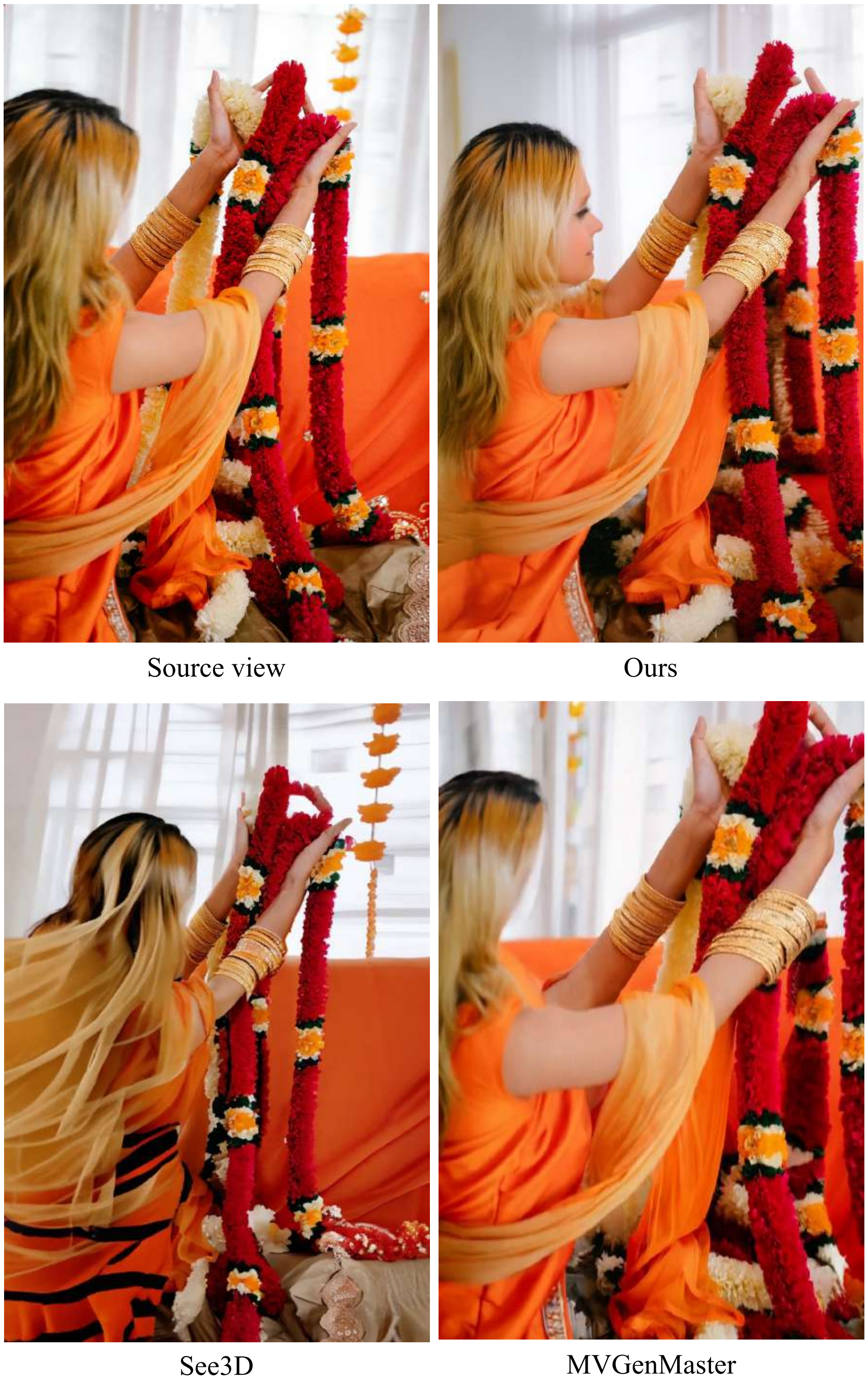

Figure 20: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, See3D yields inaccurate viewpoint changes and significantly alters the original appearance, MVGenMaster fails to complete reasonable content for the human face.

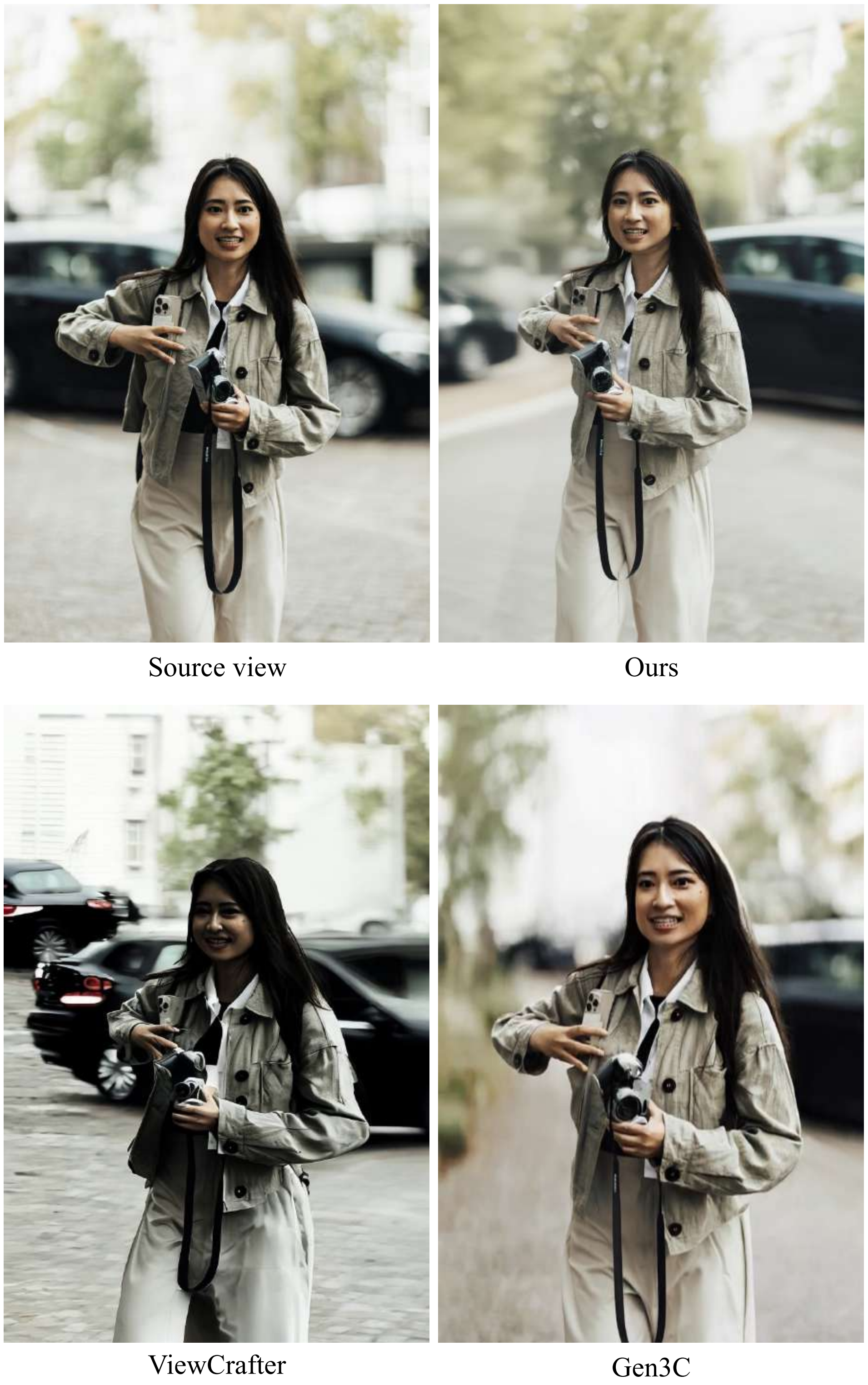

Figure 21: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, ViewCrafter produces only a small rotation with hand distortions and skin color changes, GEN3C introduces facial distortions.

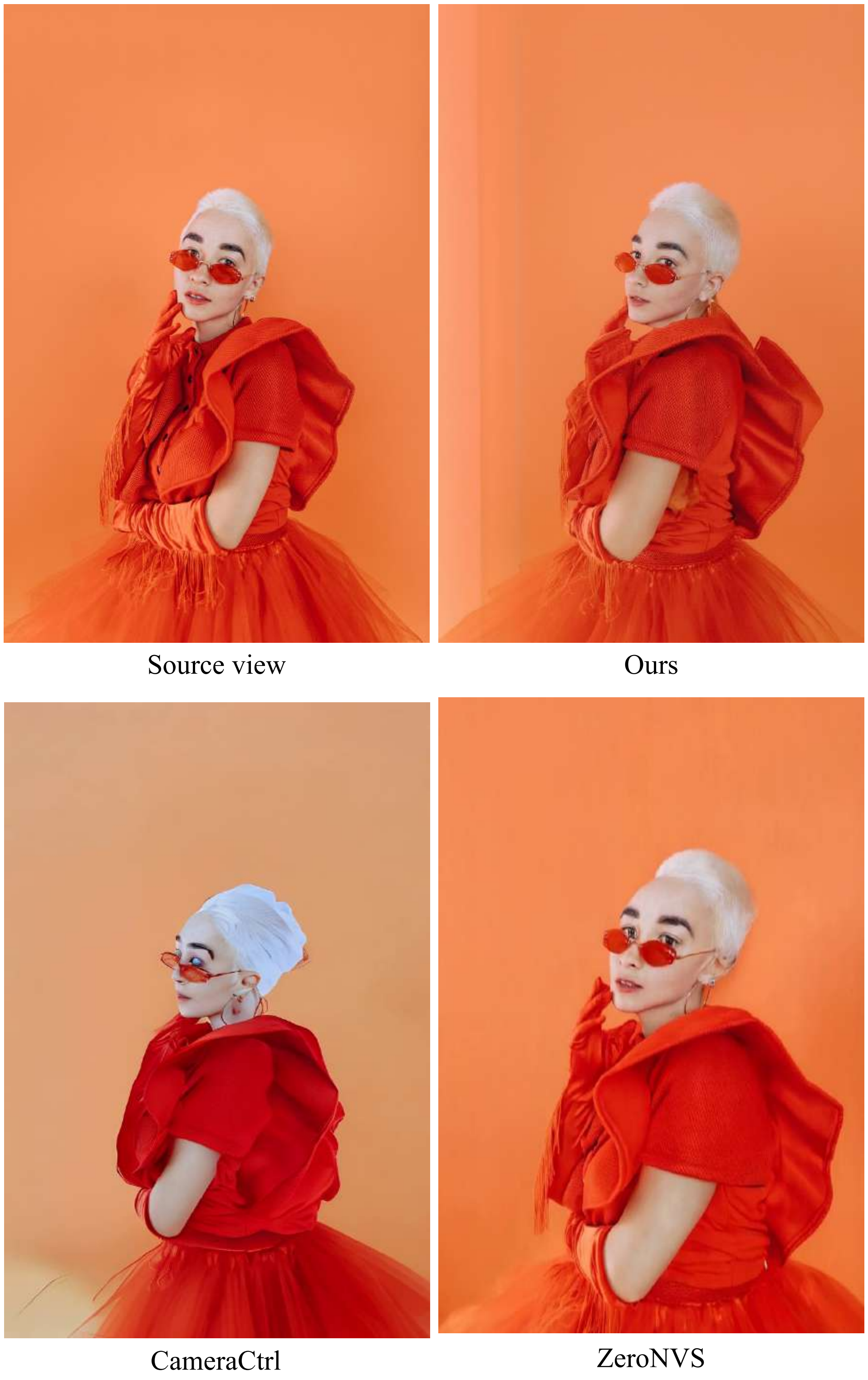

Figure 22: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, CameraCtrl and ZeroNVS both cause facial distortions.

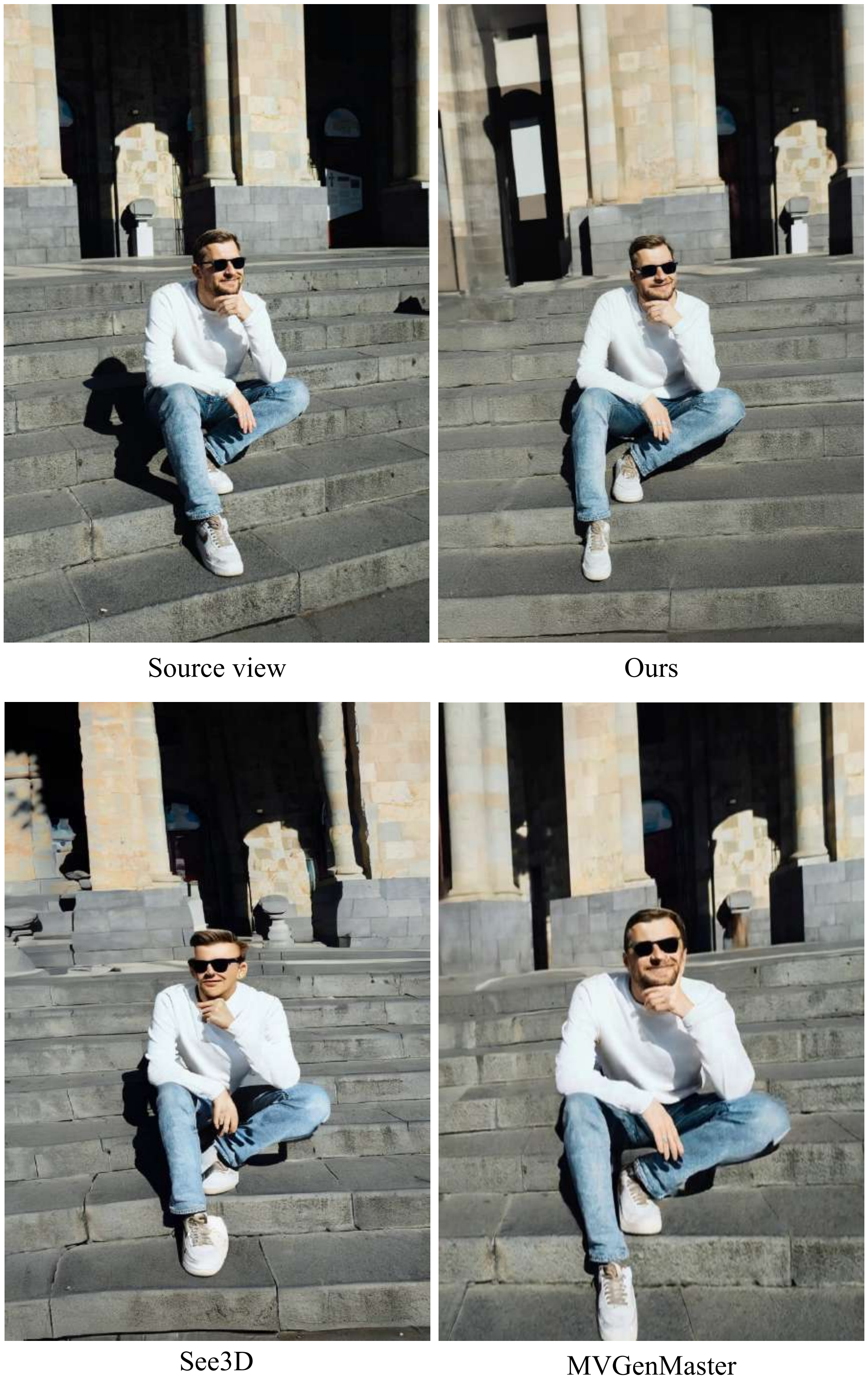

Figure 23: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, See3D and MVGenMaster both introduce distortions in the human face and shoes.

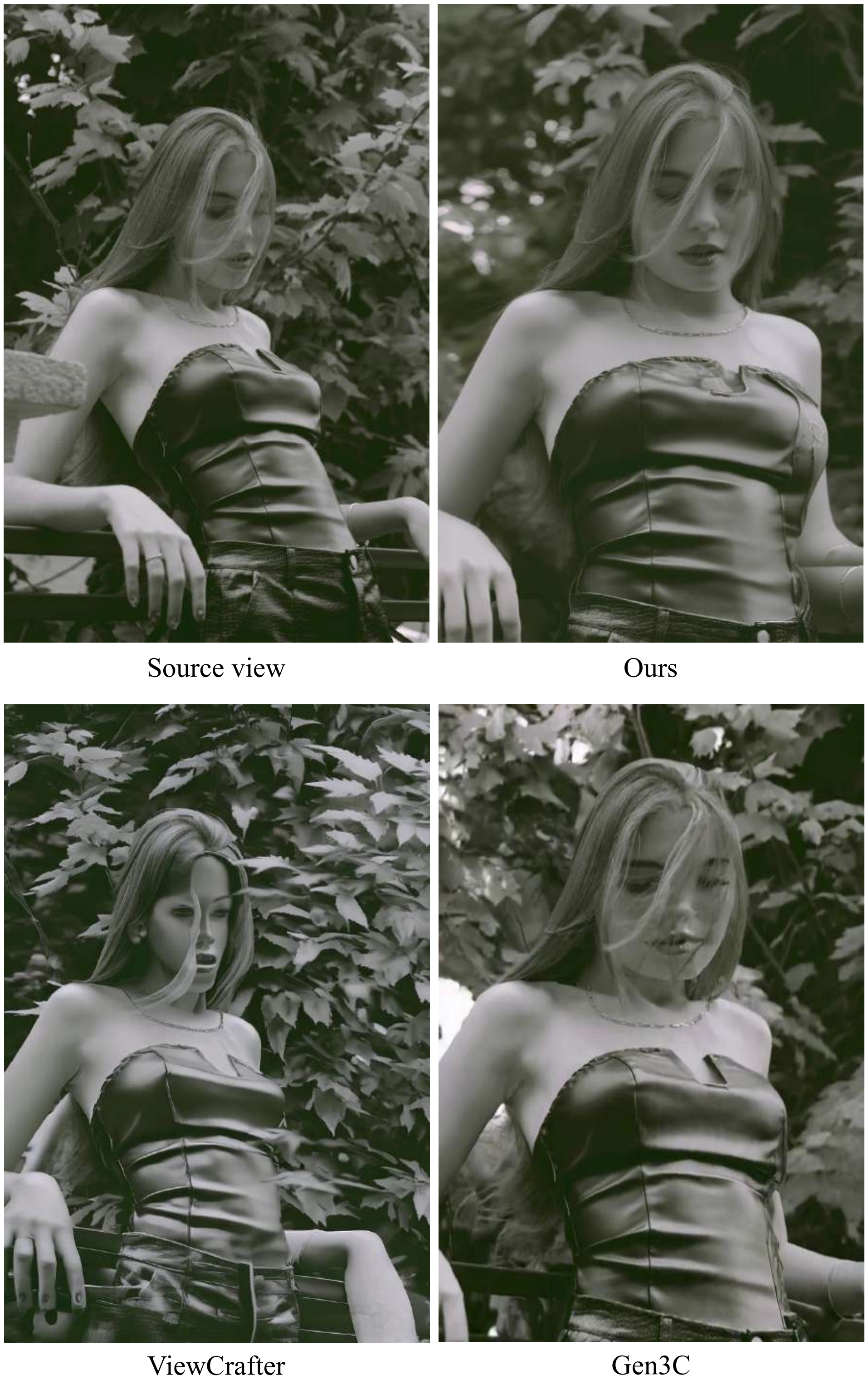

Figure 24: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, ViewCrafter produces only a small rotation but completely distorts the face, GEN3C introduces facial distortions.

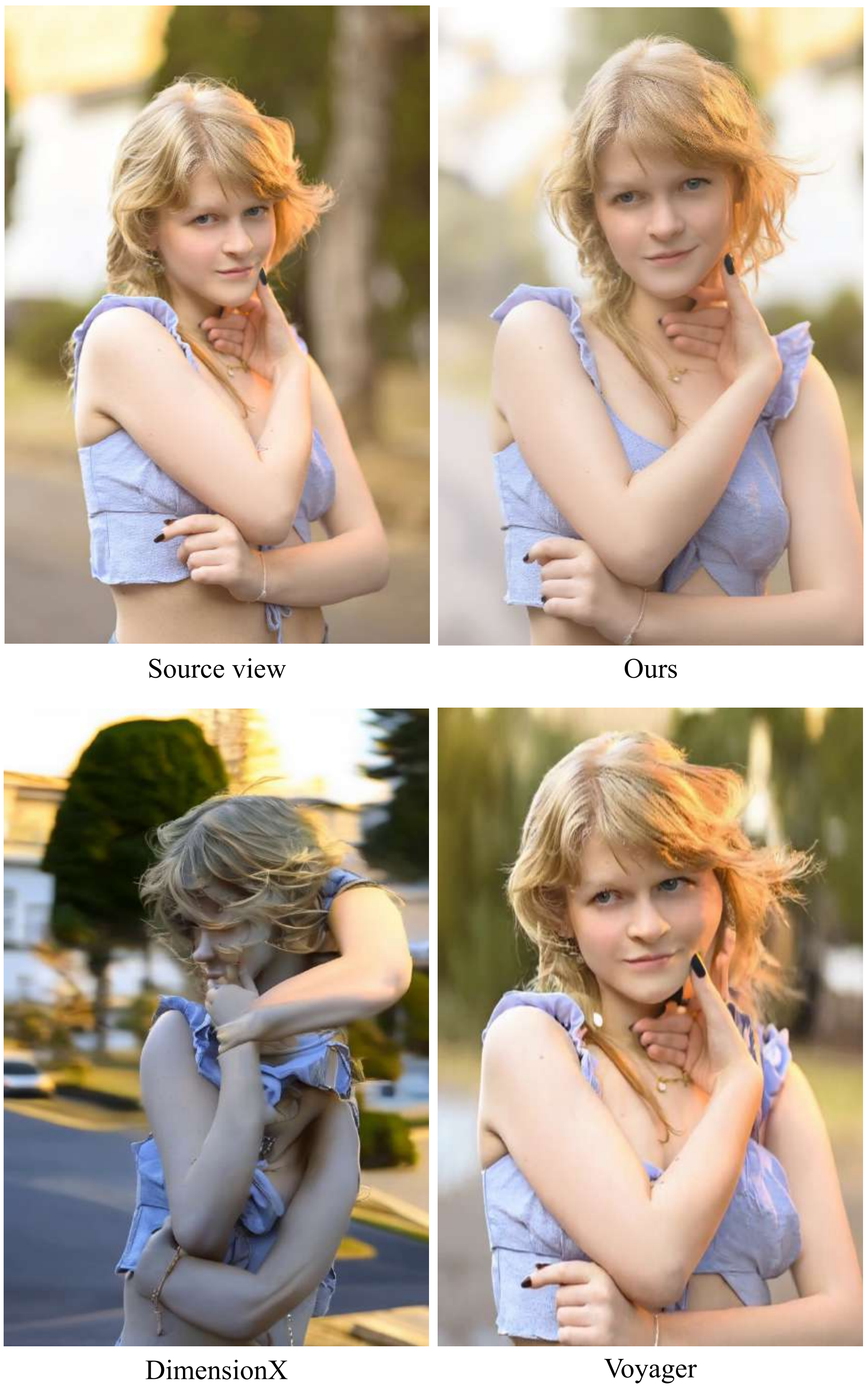

Figure 25: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, DimensionX completely distorts the human figure, Voyager introduces facial distortions.

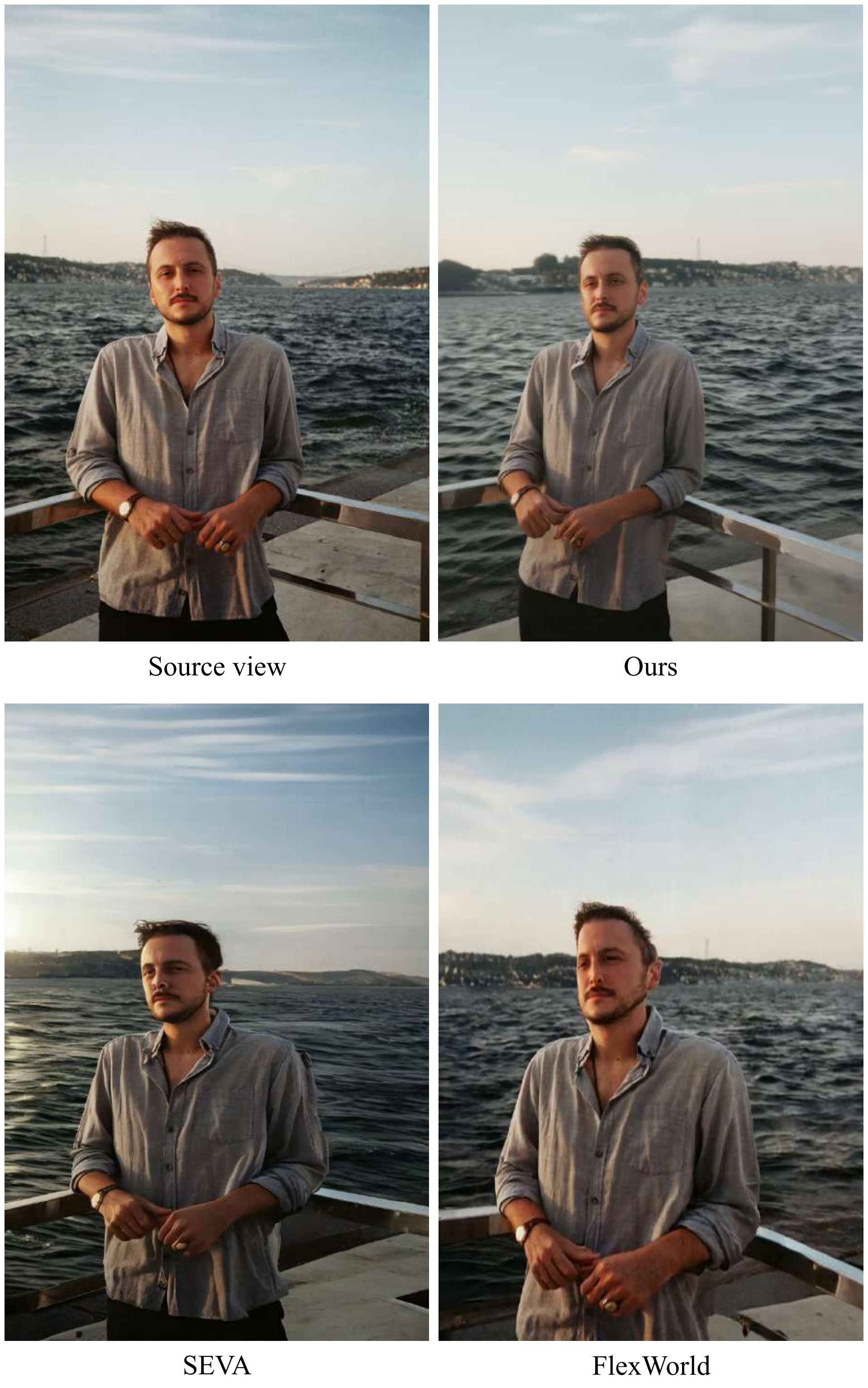

Figure 26: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, SEVA completely distorts the human face, FlexWorld introduces facial distortions.

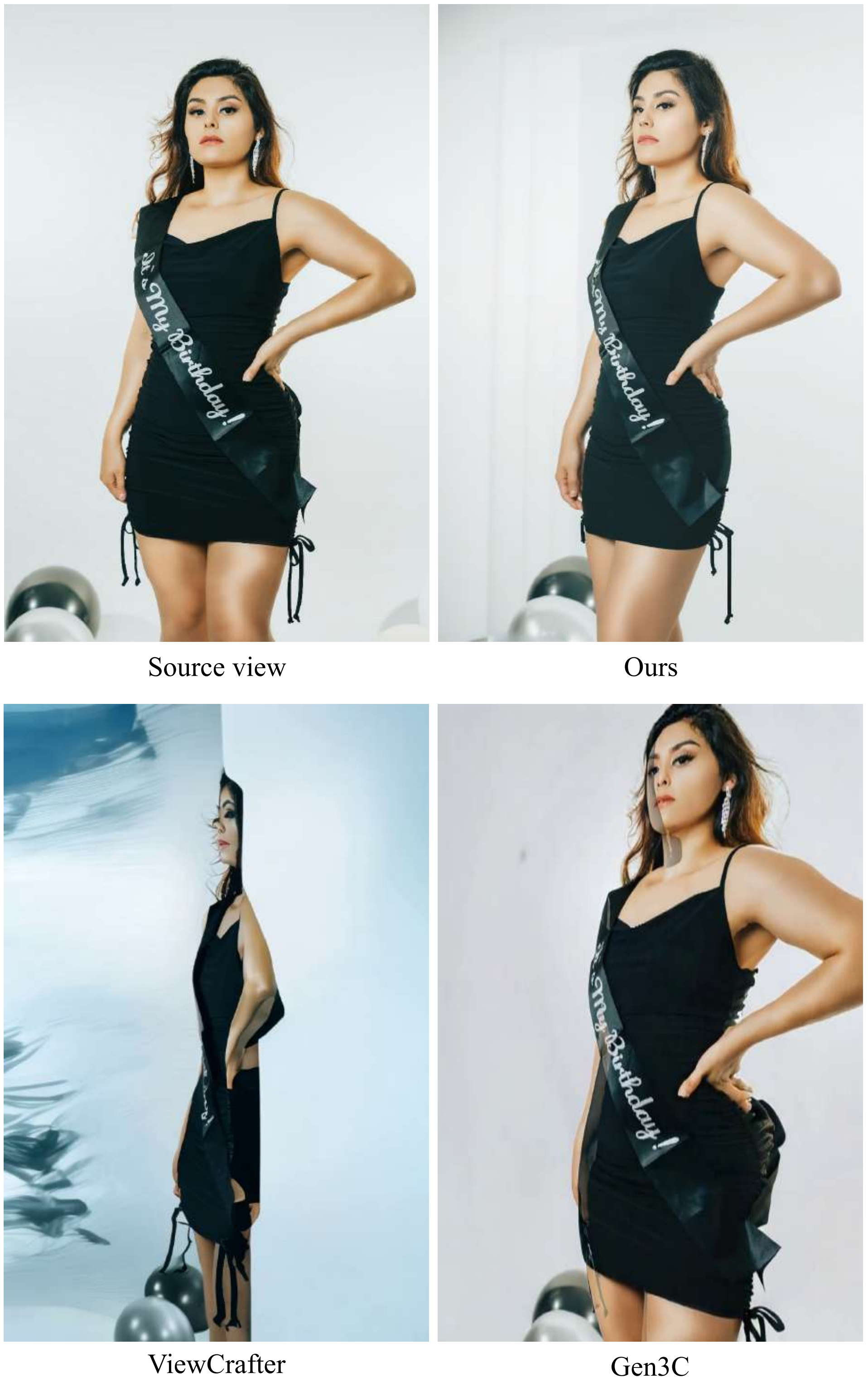

Figure 27: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, ViewCrafter completely distorts the human figure, GEN3C introduces facial artifacts.

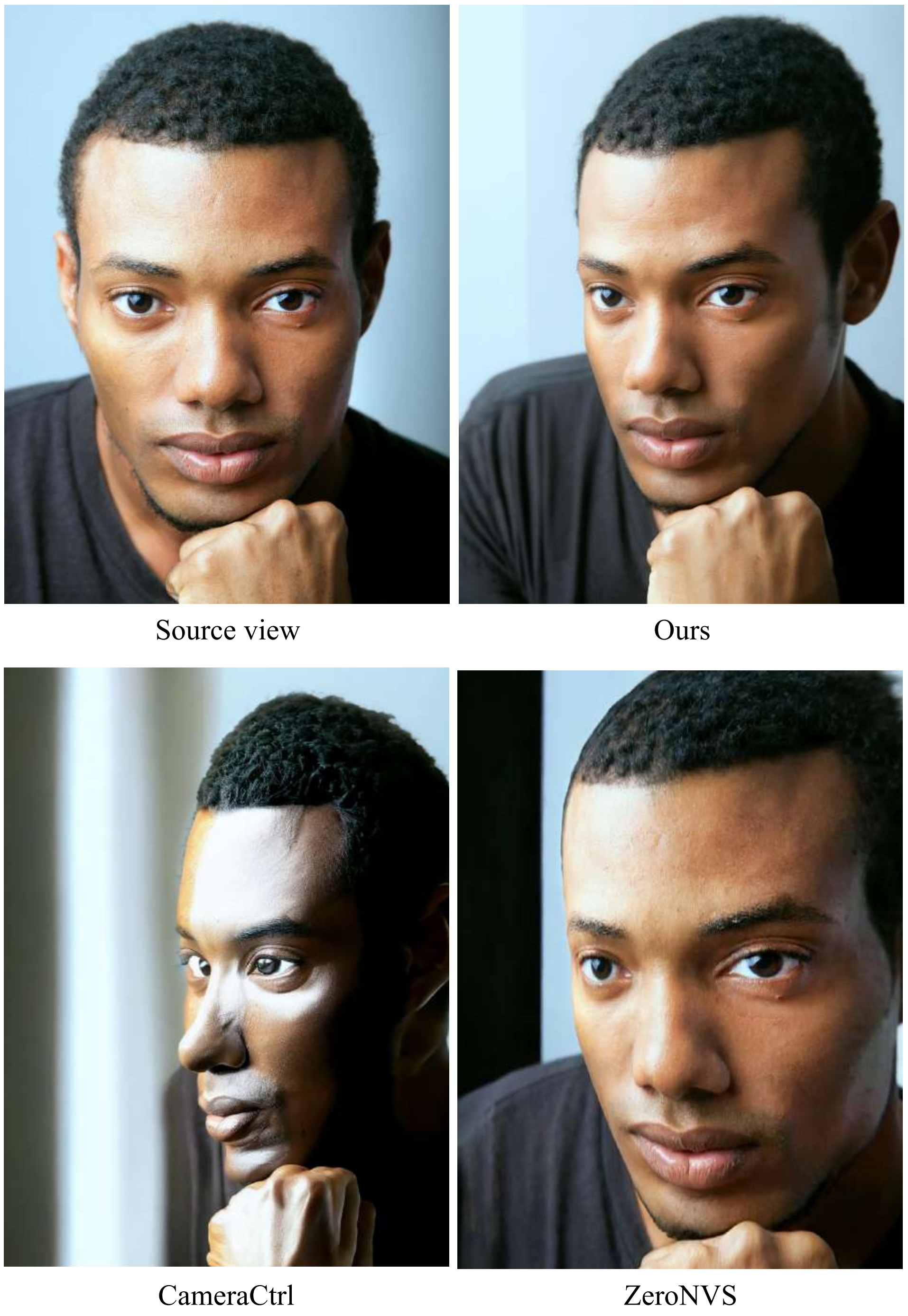

Figure 28: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, CameraCtrl and ZeroNVS both cause facial distortions.

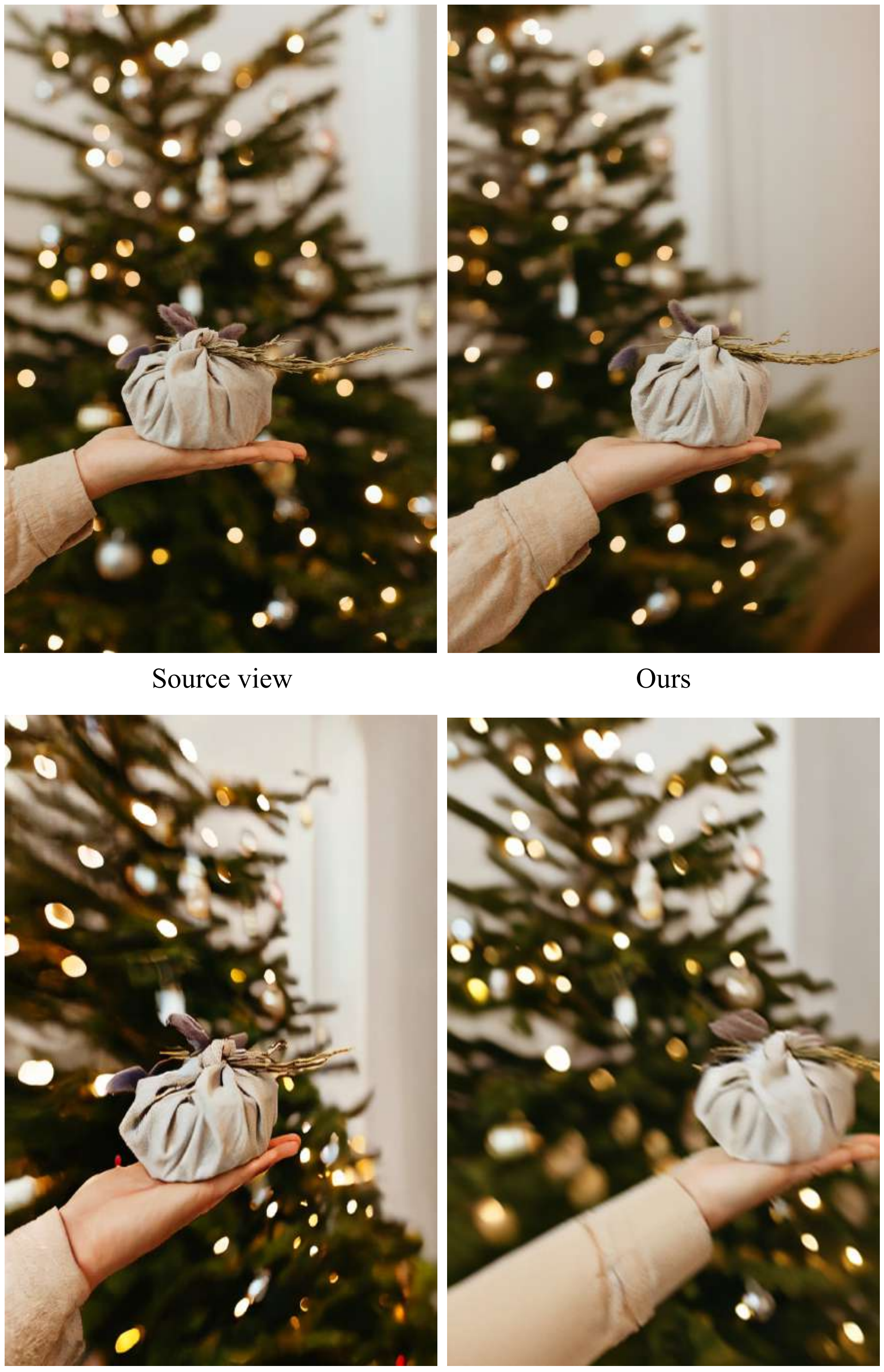

Figure 29: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, NVS-Solver shows little change but introduces some distortions, GenWarp produces blurred results.

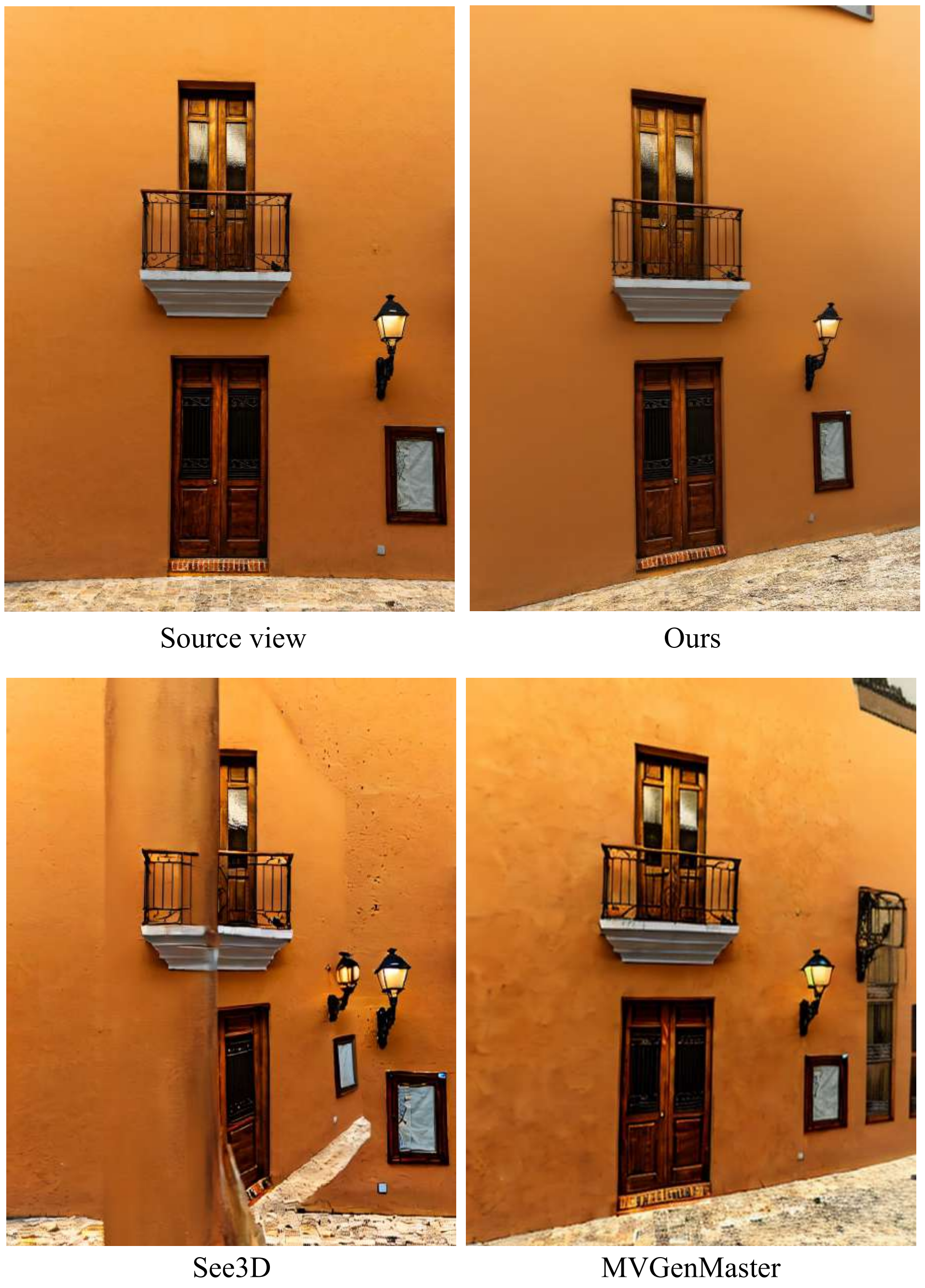

Figure 30: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, See3D distorts the scene, and MVGenMaster adds noise to the results.

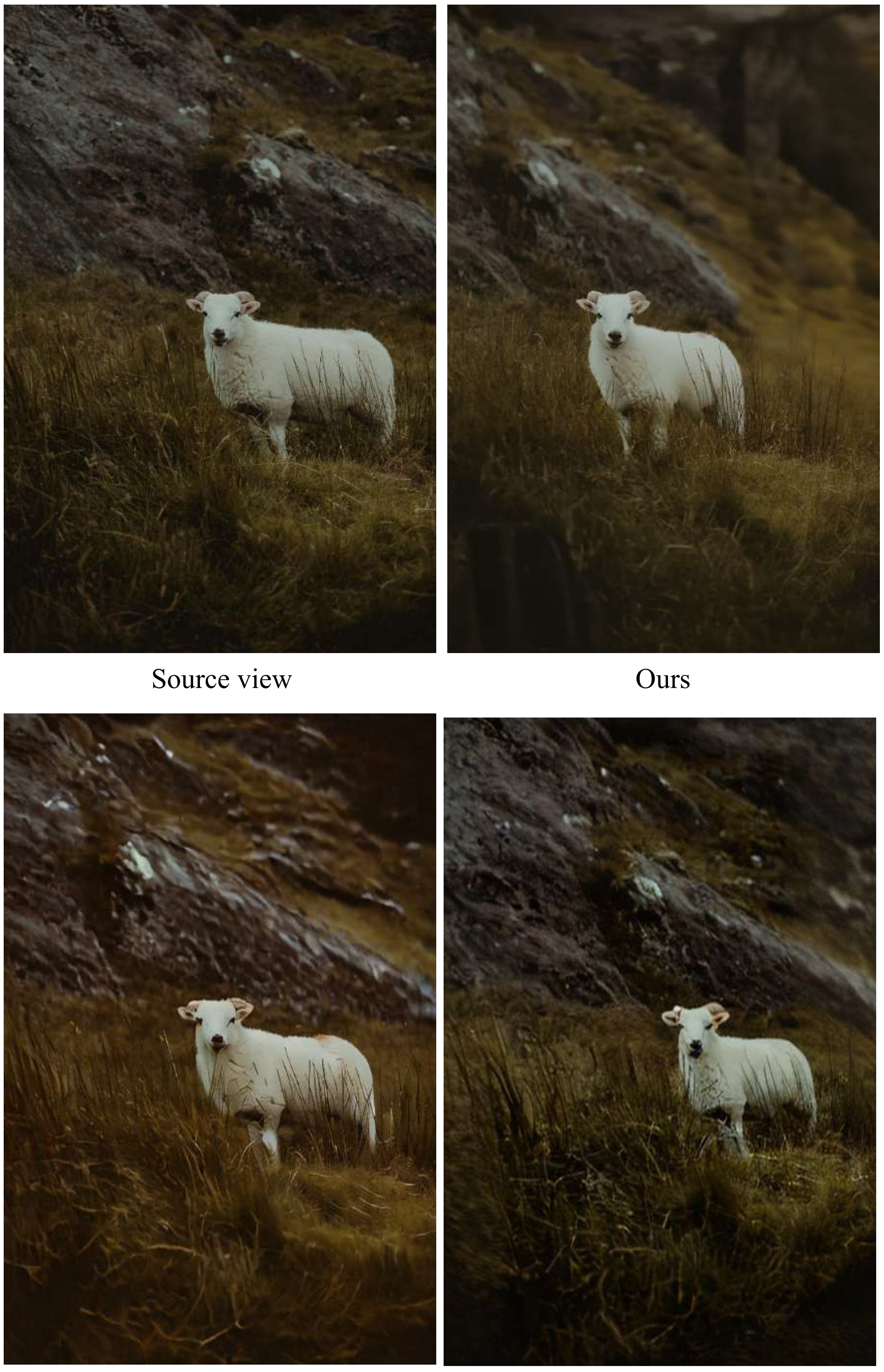

Figure 31: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, NVS-Solver introduces distortions on the sheep and alters the overall color tone, GenWarp produces smaller changes but still distorts the sheep.

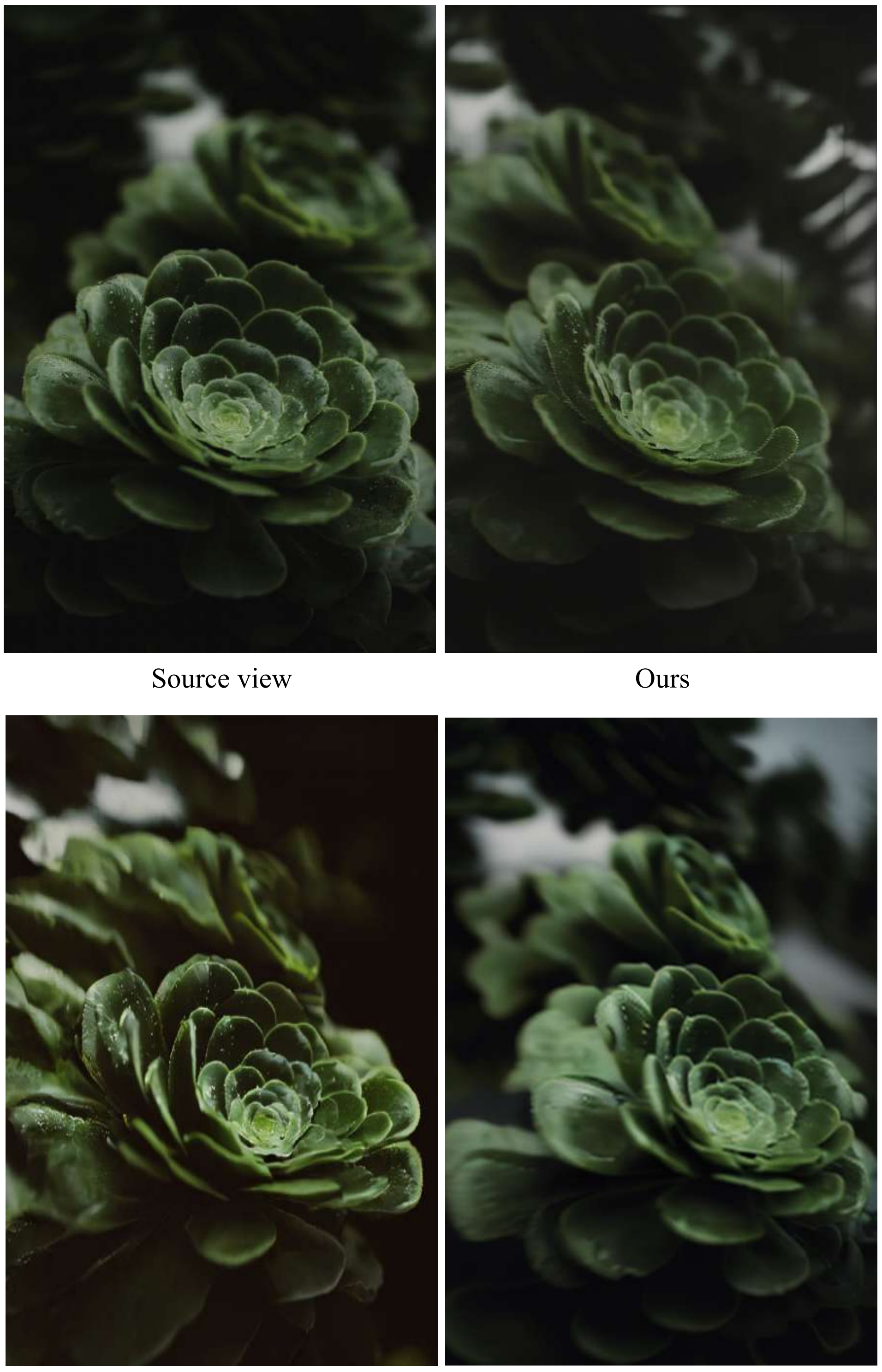

Figure 32: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, DimensionX and Voyager both alter the shape of the flowers.

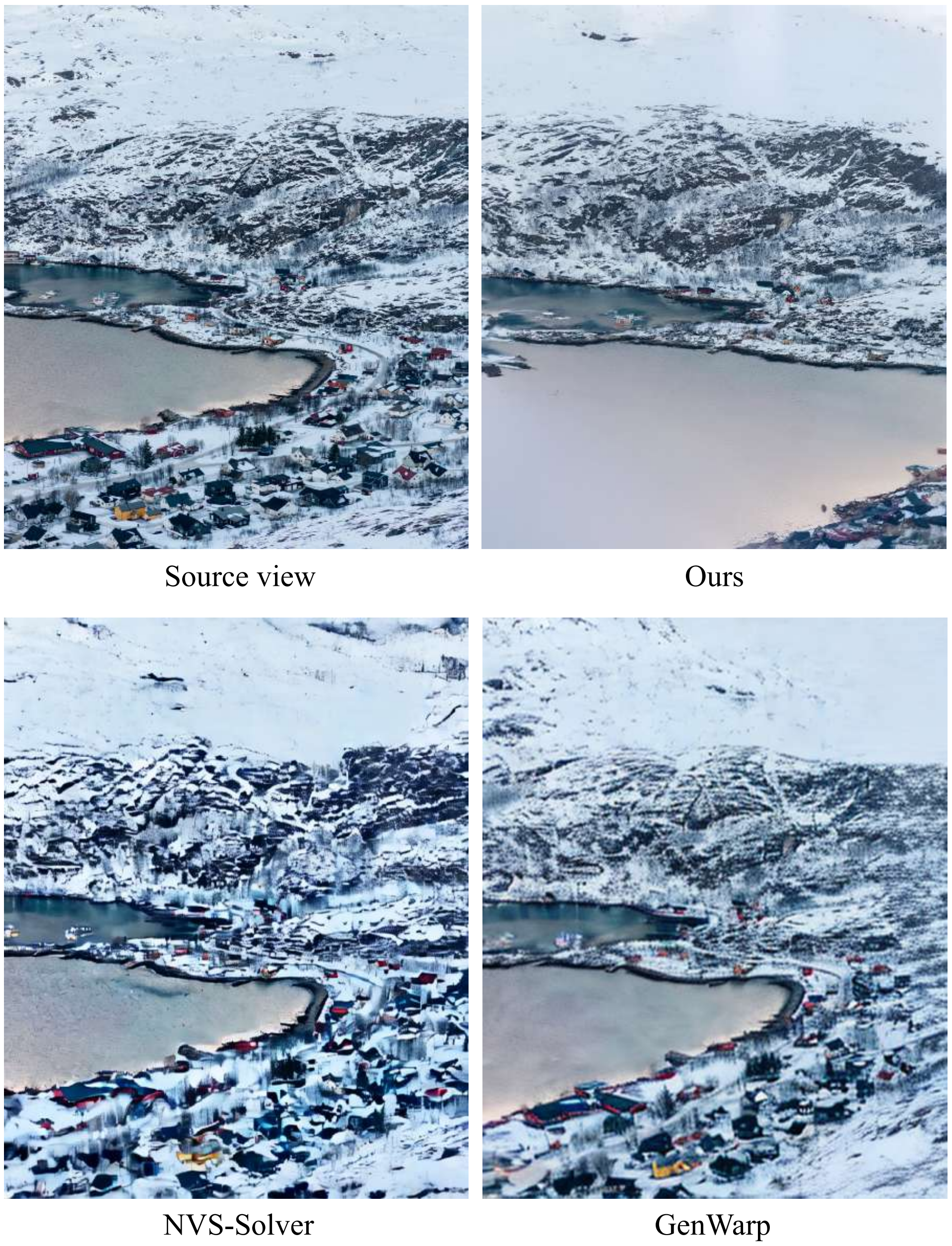

Figure 33: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, NVS-Solver and GenWarp show little change but generate blurred images.

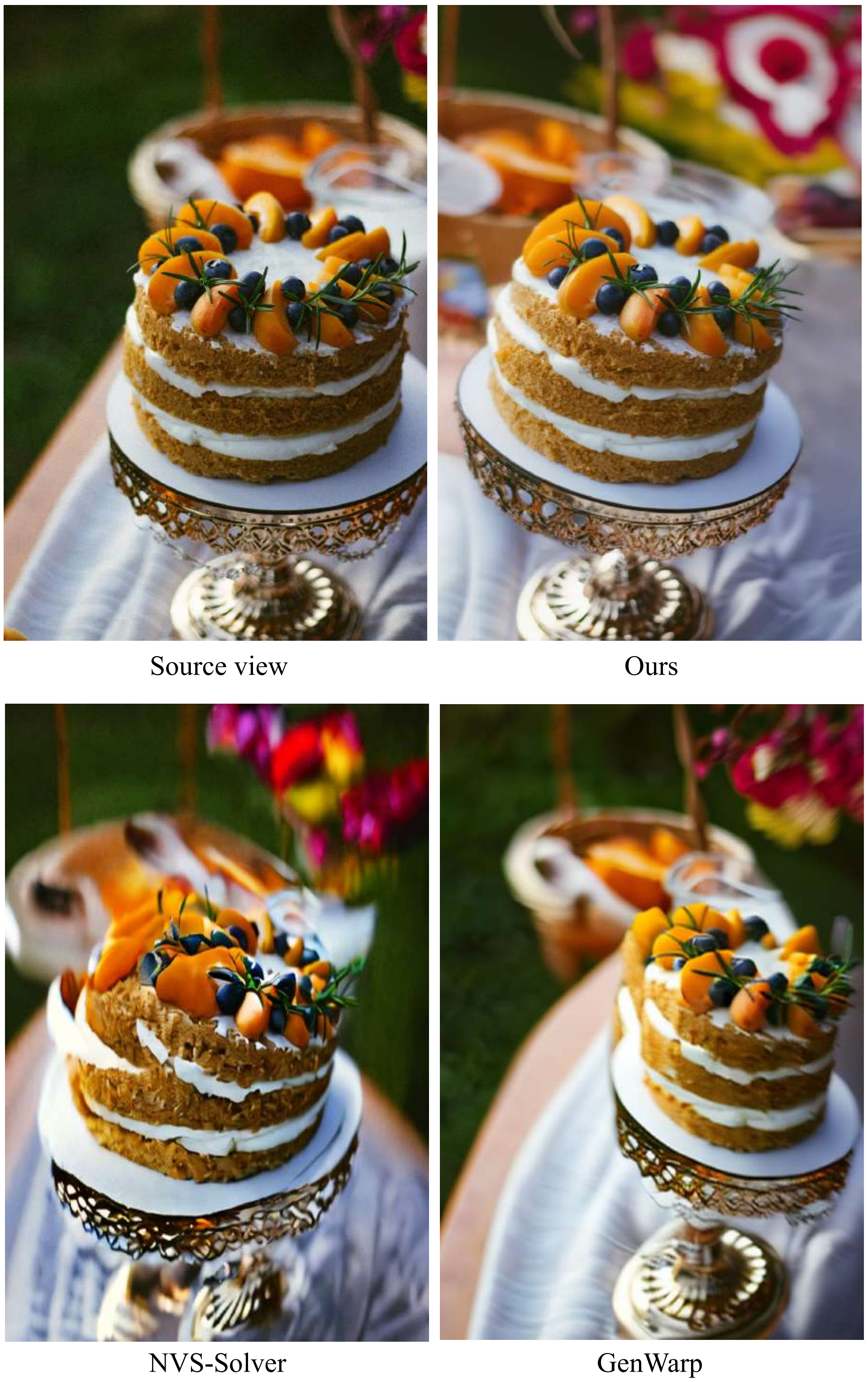

Figure 34: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, NVS-Solver and GenWarp both introduce object distortions and produce blurred results.

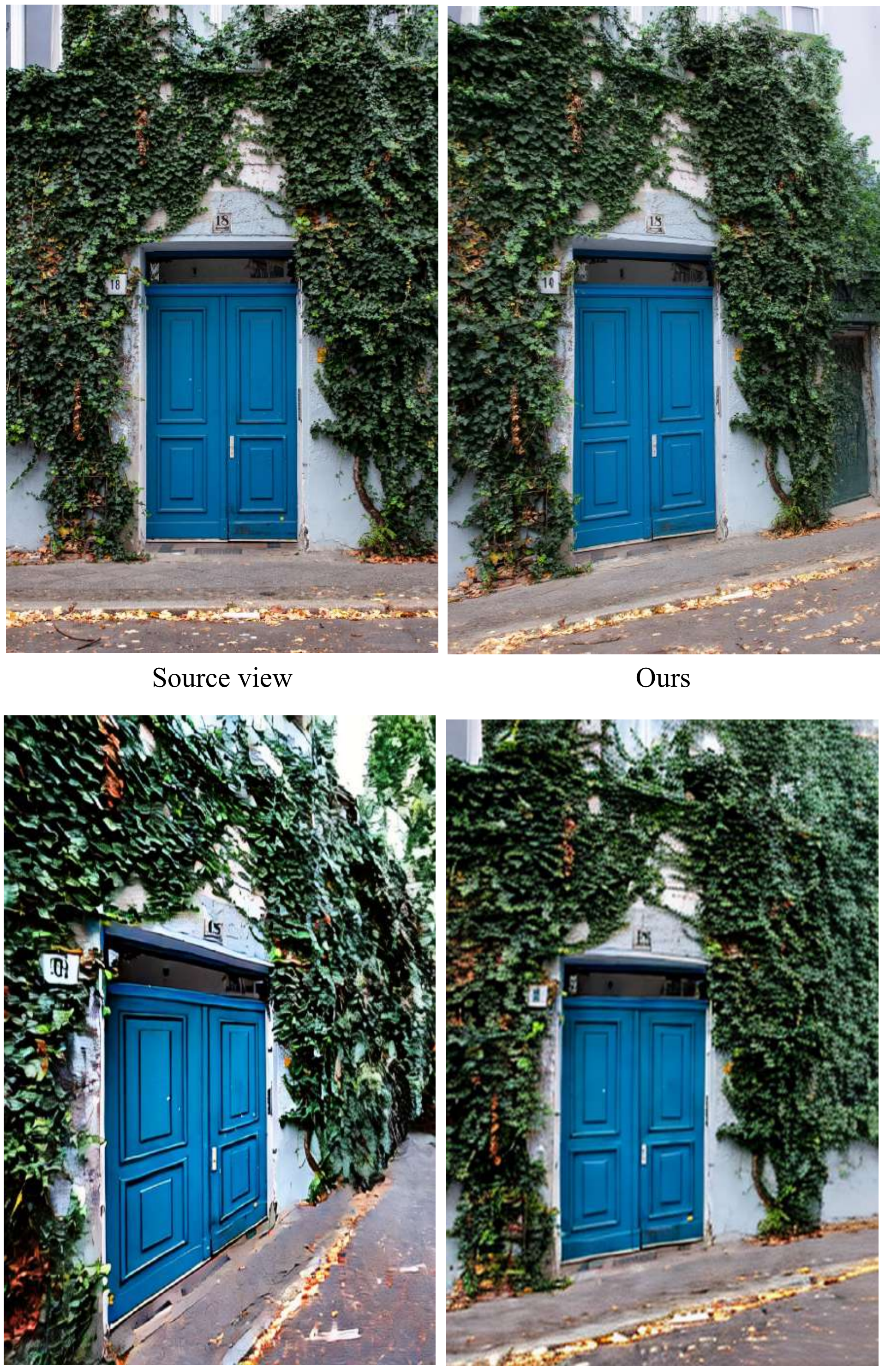

Figure 35: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, See3D distorts the scene, and MVGenMaster adds noise to the results.

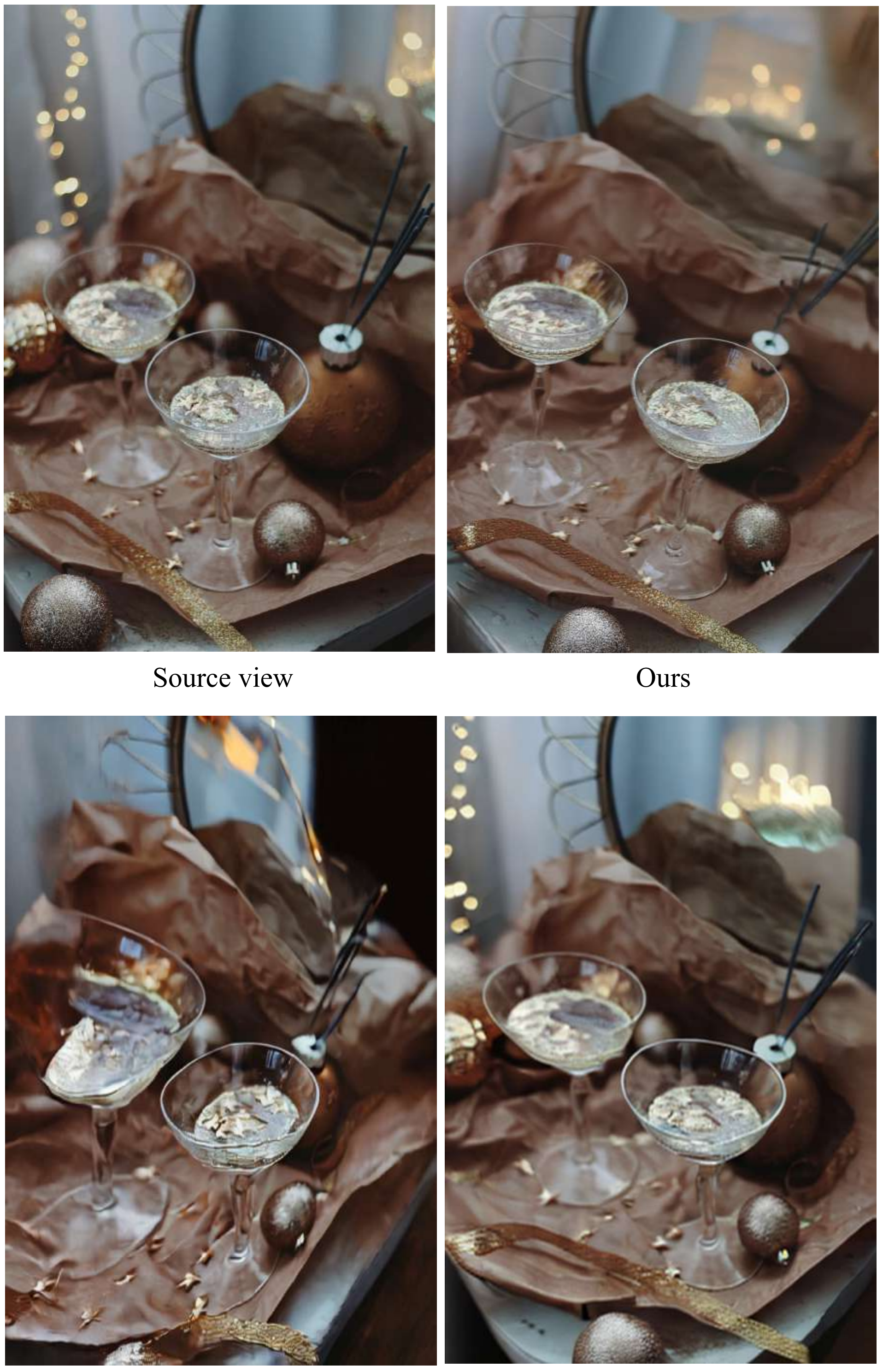

Figure 36: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, DimensionX and Voyager both alter the shape of the goblet.

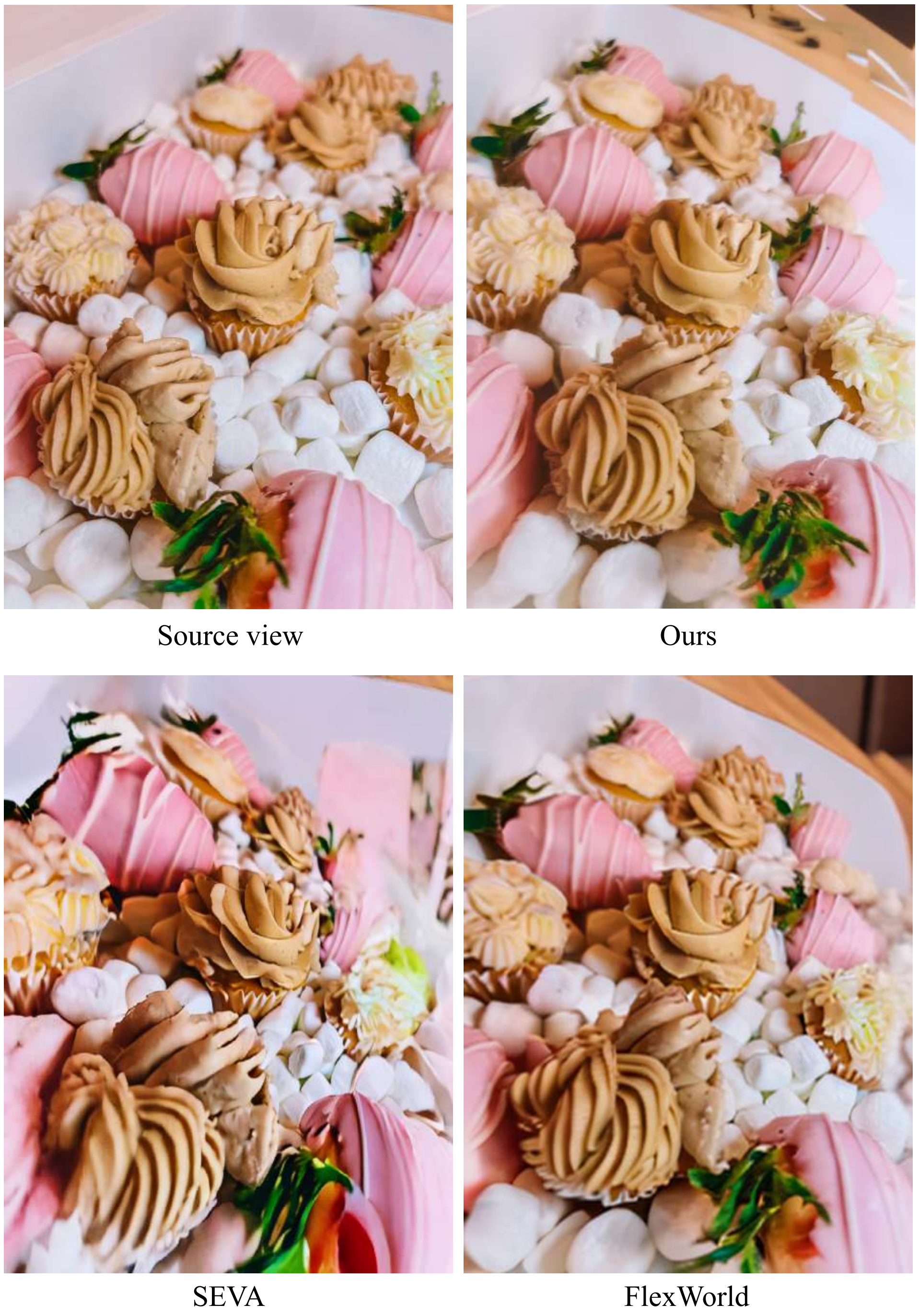

Figure 37: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, SEVA produces completely distorted results, FlexWorld introduces object distortions and alters the overall image style.

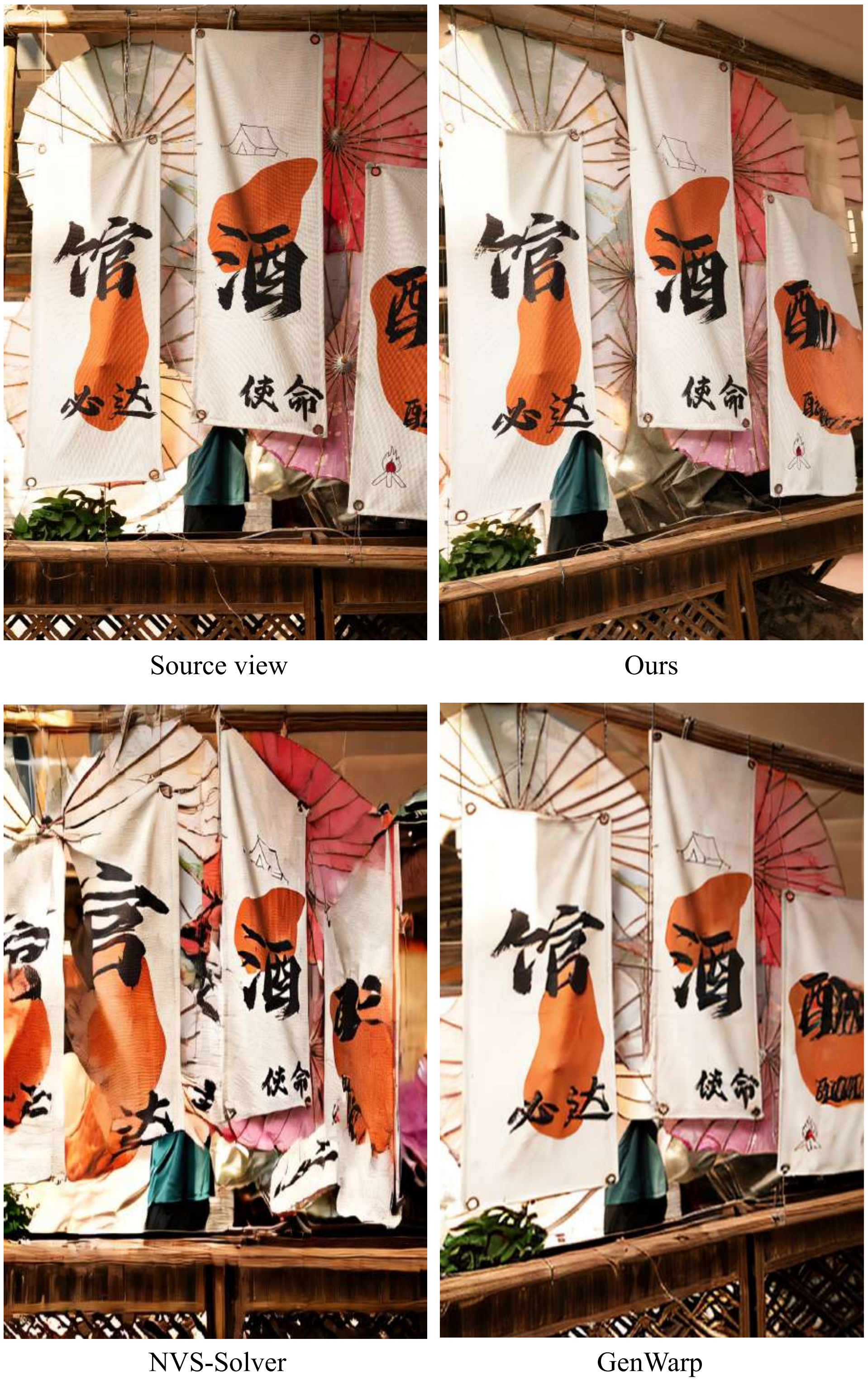

Figure 38: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, NVS-Solver shows little change but introduces some distortions, GenWarp produces blurred results.

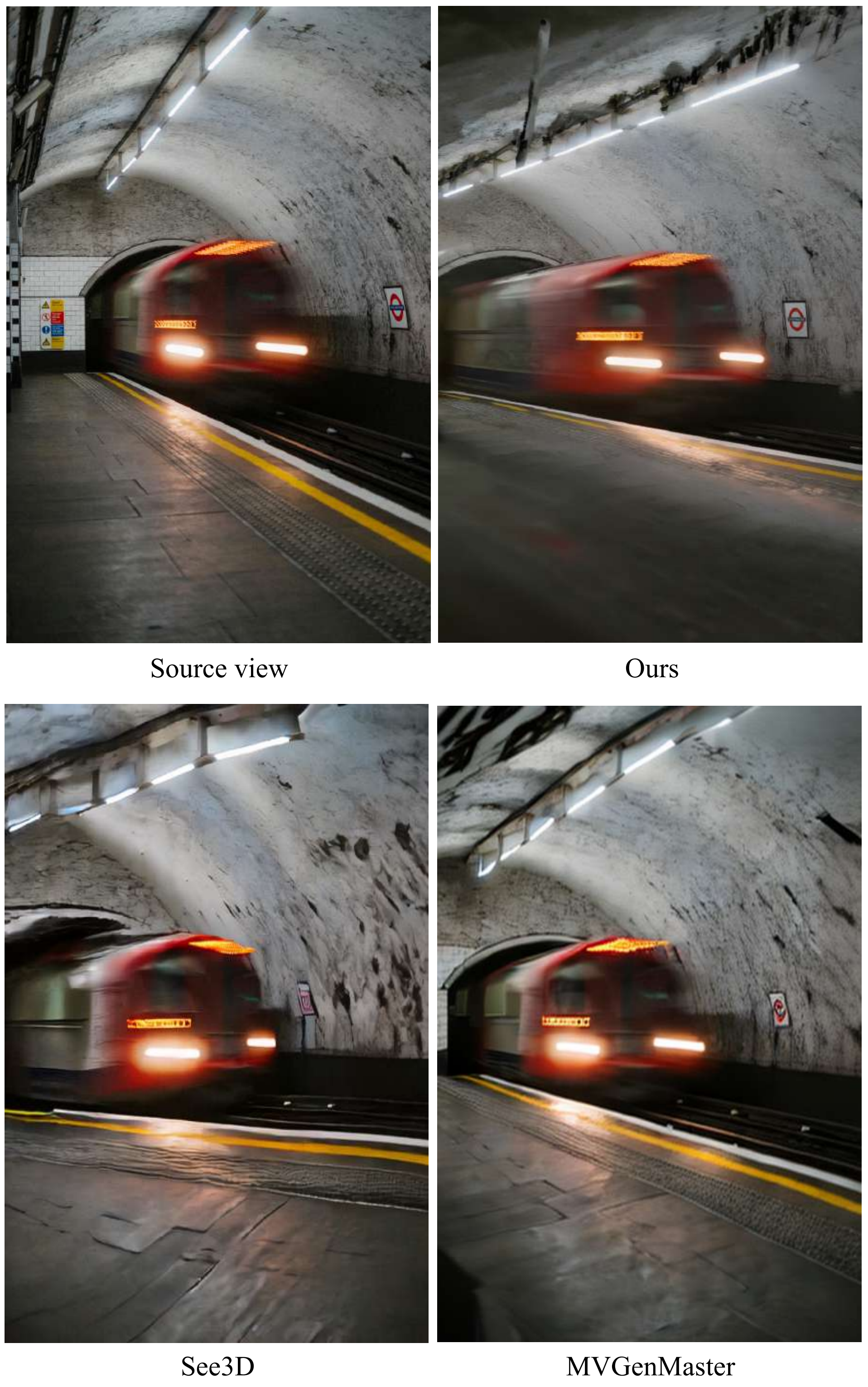

Figure 39: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, See3D distorts the scene, and MVGenMaster adds noise to the results.

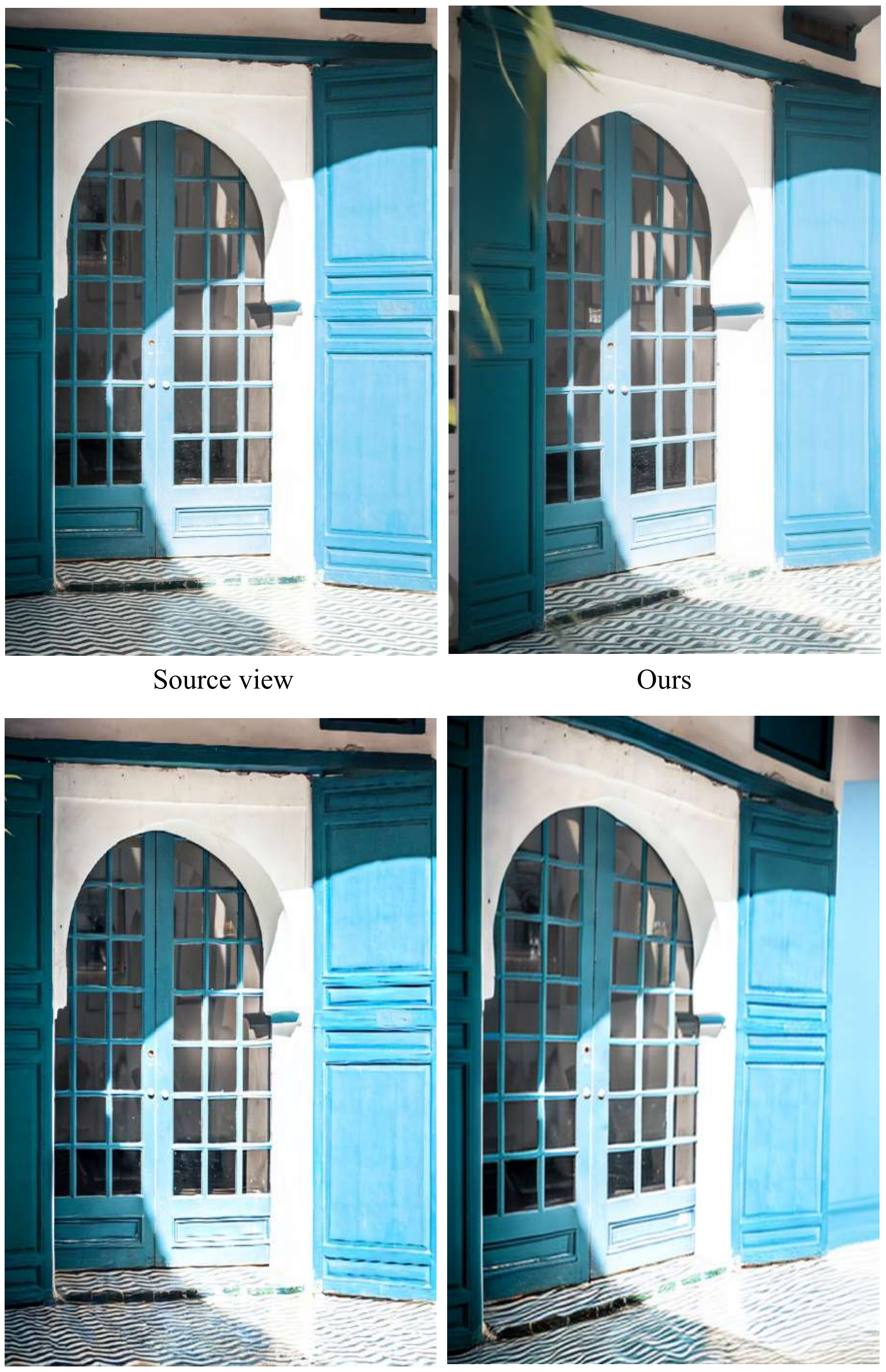

Figure 40: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, SEVA shows little change, FlexWorld affects the floor patterns.

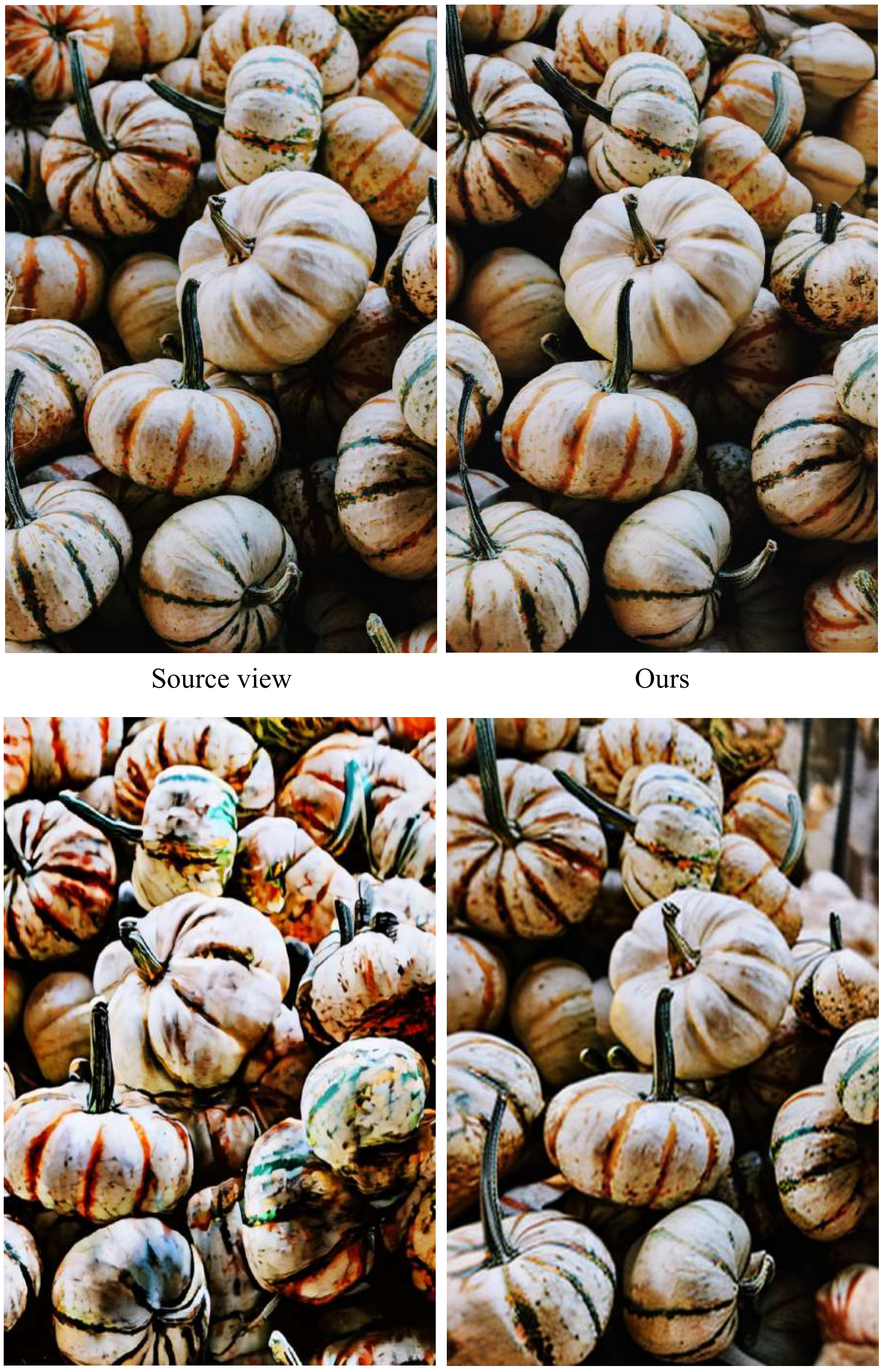

Figure 41: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° leftward rotation, DimensionX and Voyager alter both the object shapes and the overall image style.

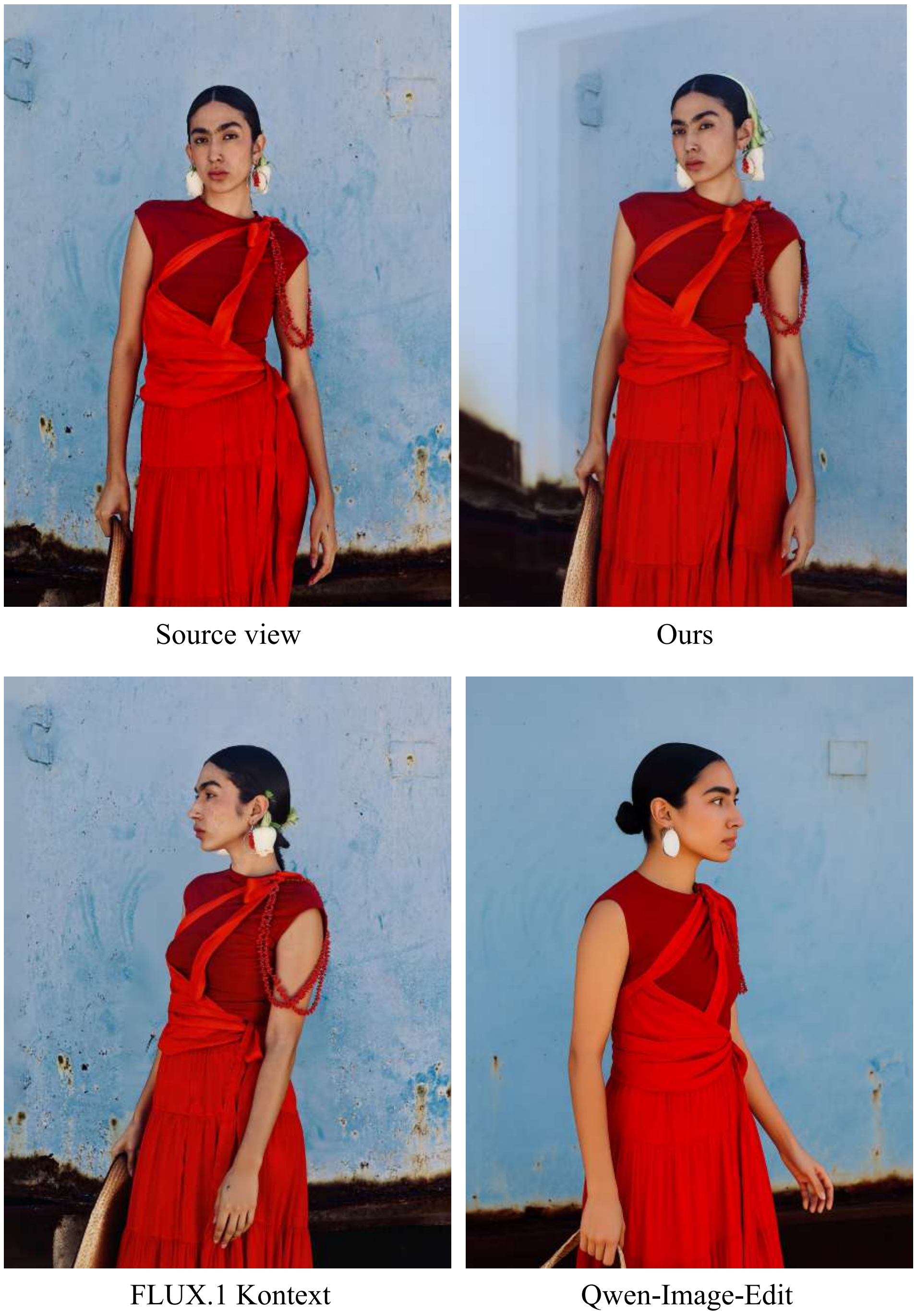

Figure 42: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX only rotates the person without rotating the background, Qwen rotates in the opposite direction and also changes the person's pose.

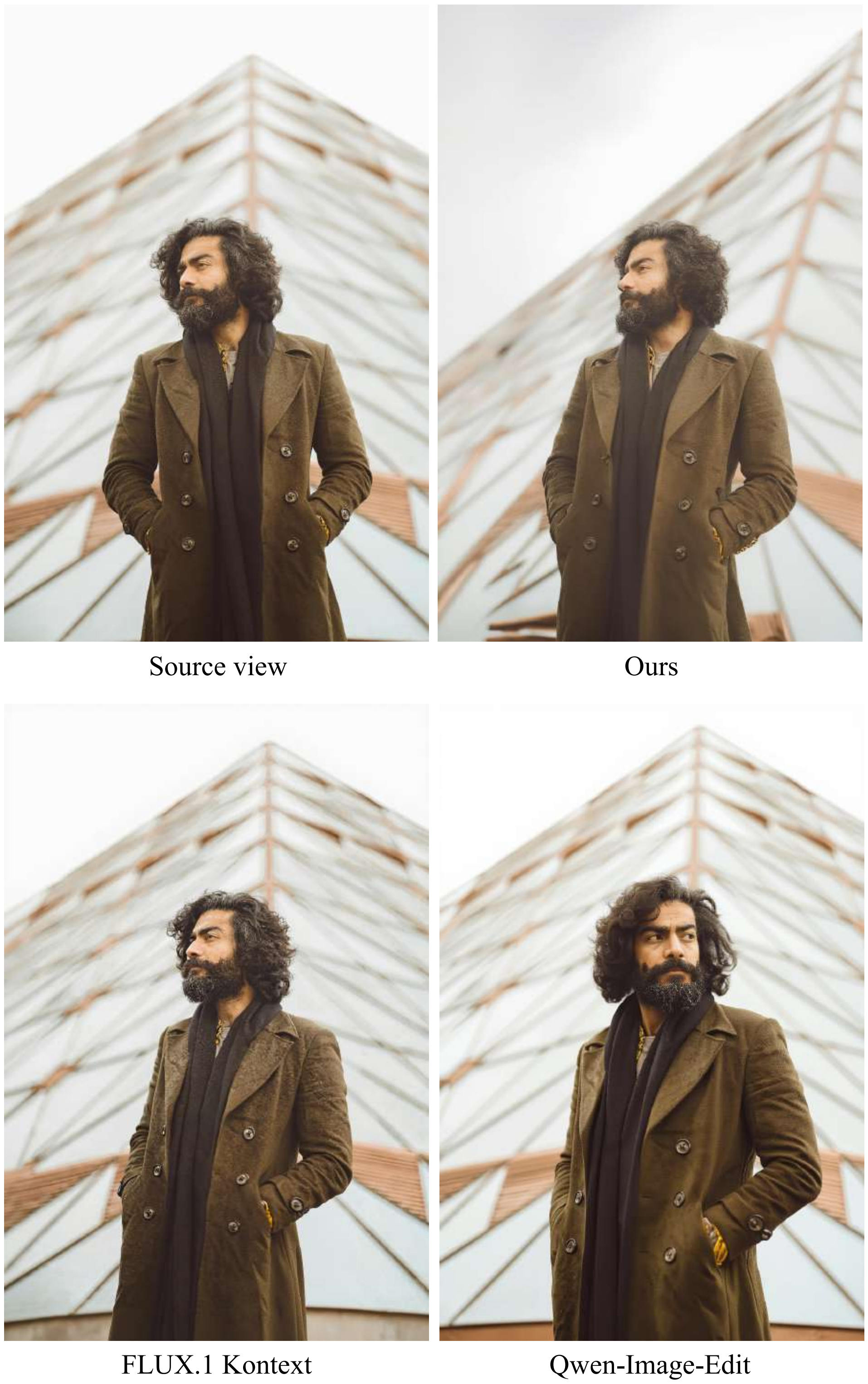

Figure 43: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX shows only minor changes without rotating the background, Qwen alters the person's pose and expression without rotating the background.

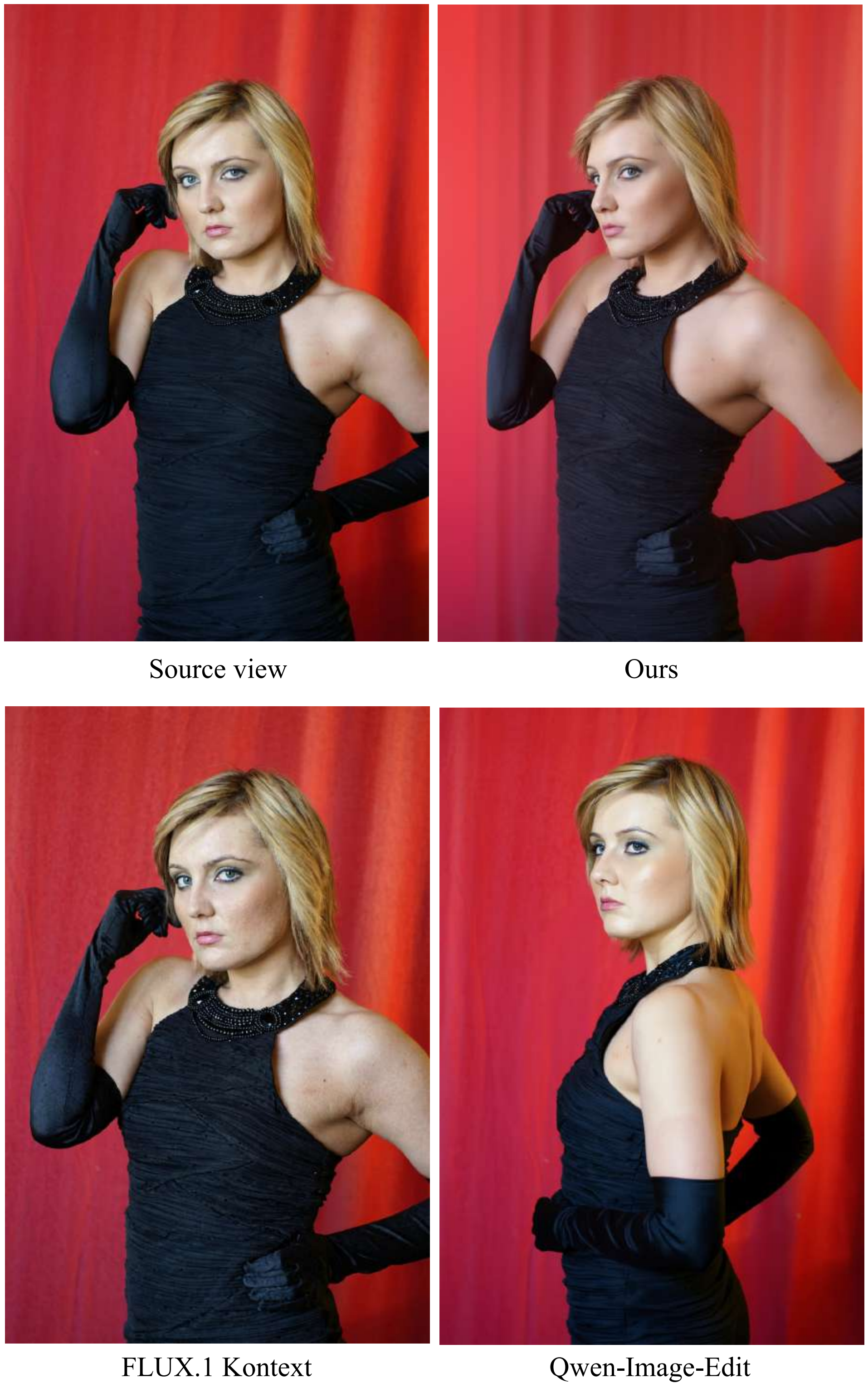

Figure 44: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX shows only minor changes, Qwen changes the person's pose without rotating the background.

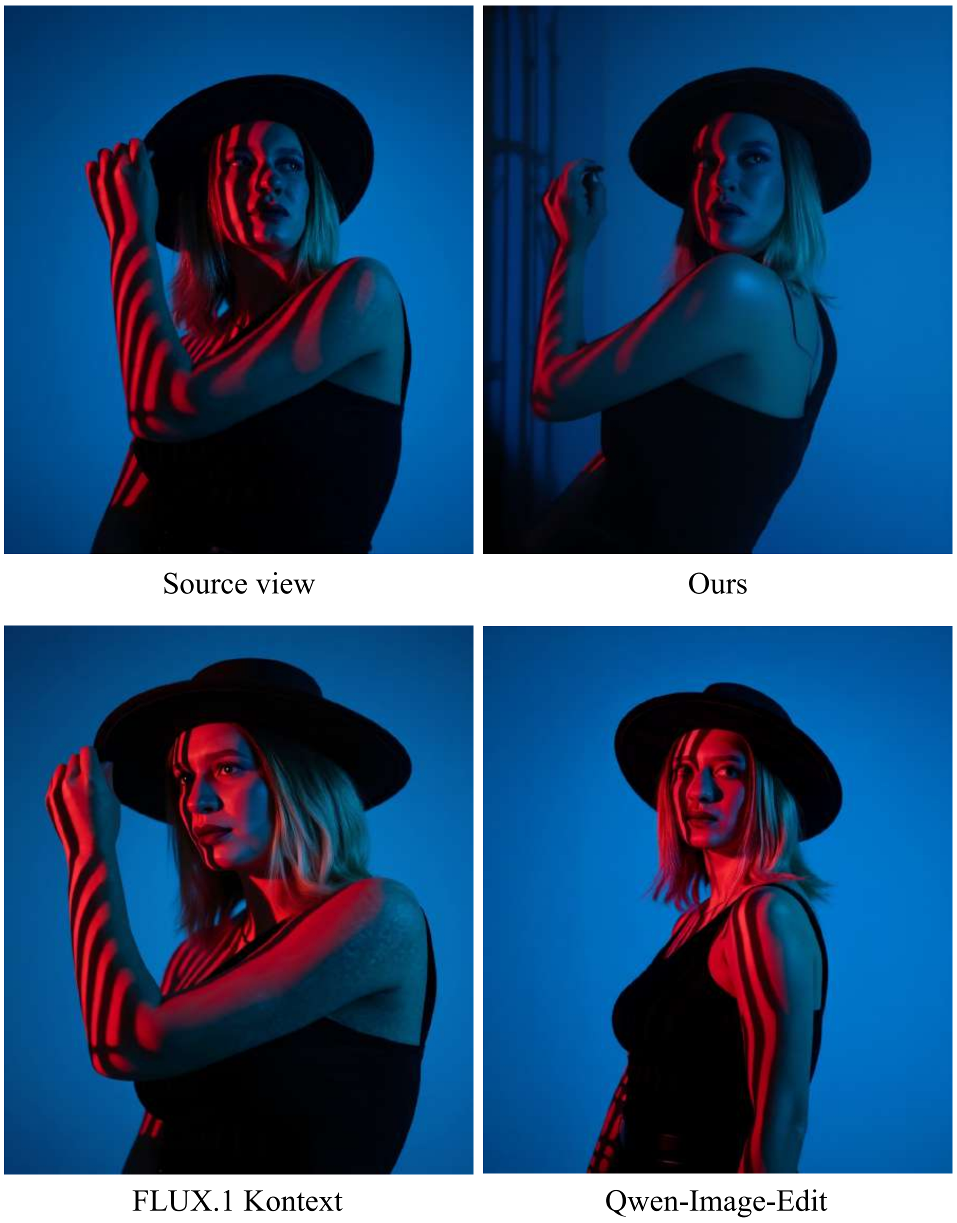

Figure 45: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX only rotates the person's head, Qwen changes the person's pose.

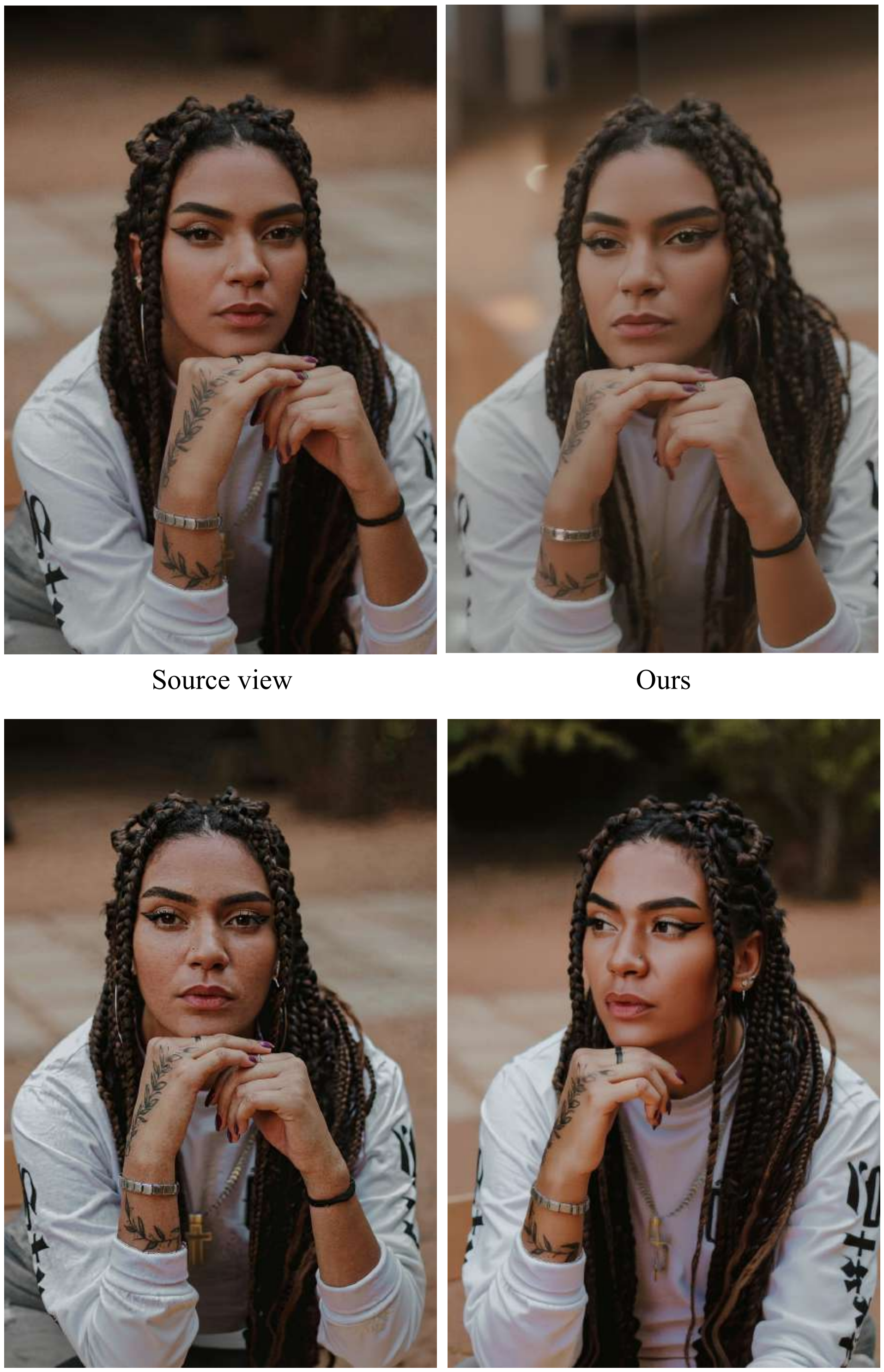

Figure 46: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX shows only minor changes, Qwen changes the person's pose.

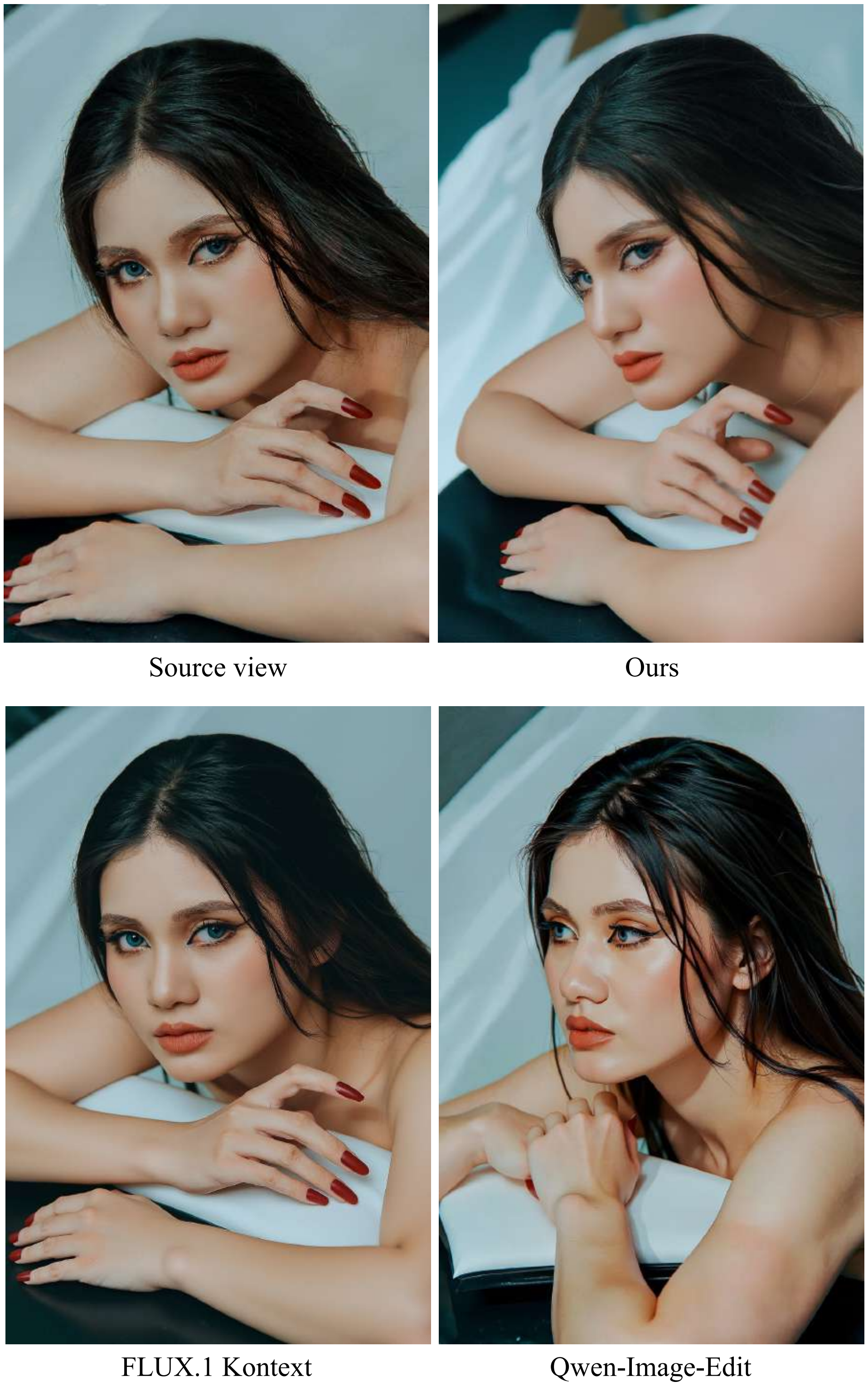

Figure 47: Supplementary novel view synthesis (NVS) examples on in-the-wild images. For the 30° rightward rotation, FLUX shows only minor changes, Qwen changes the person's pose.

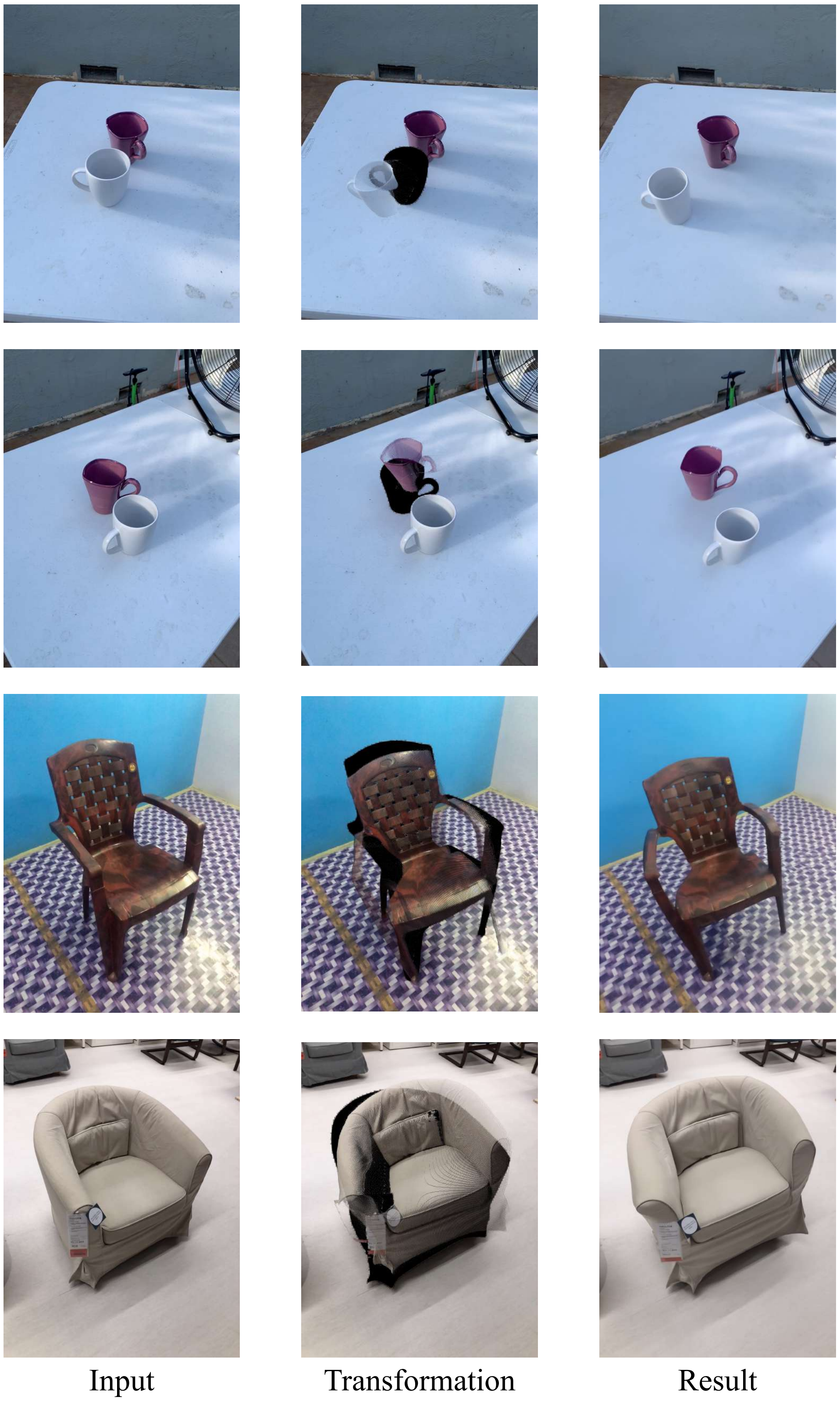

Figure 48: Supplementary 3D-aware object editing examples on Objectron dataset.

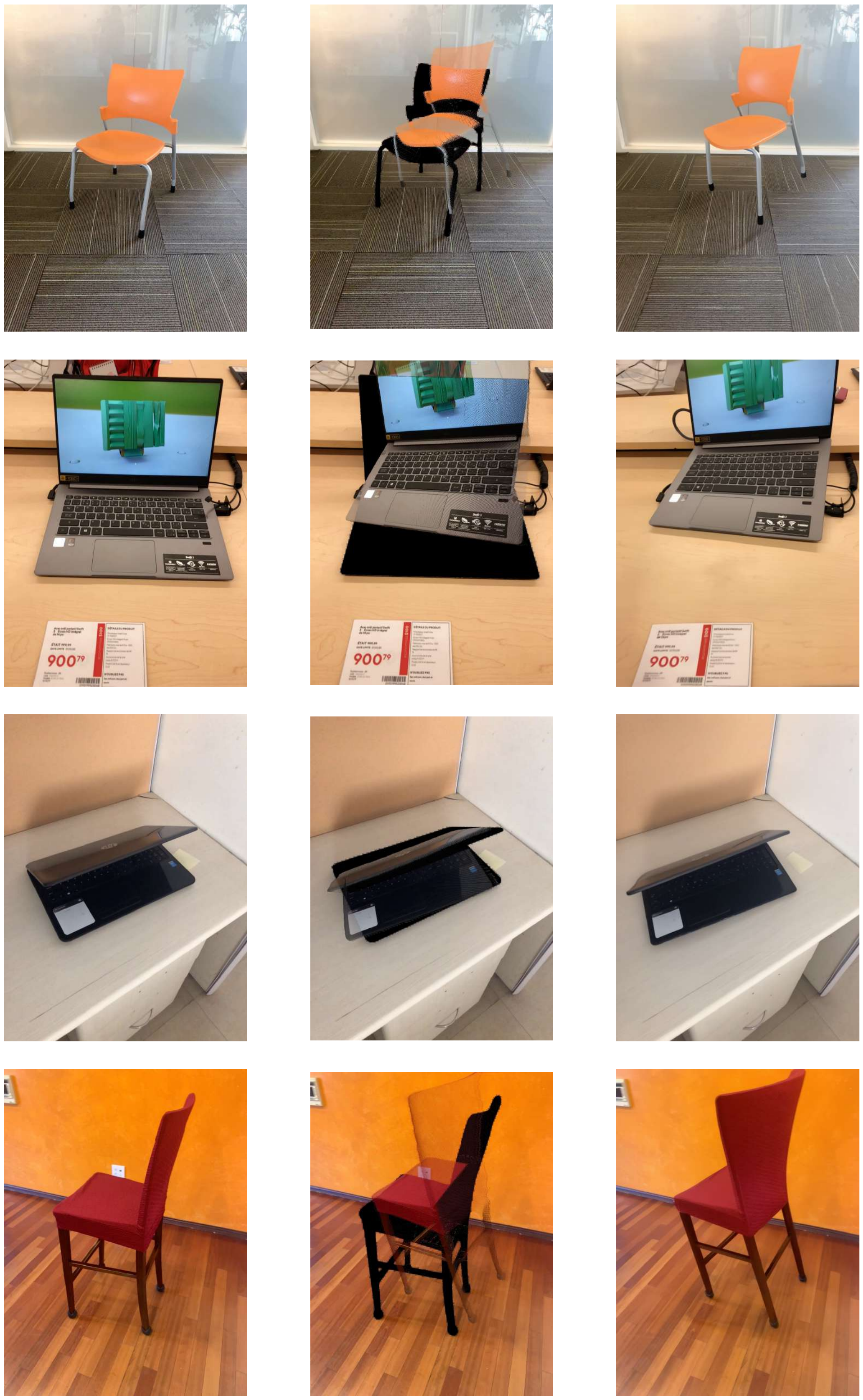

Figure 49: Supplementary 3D-aware object editing examples on Objectron dataset.

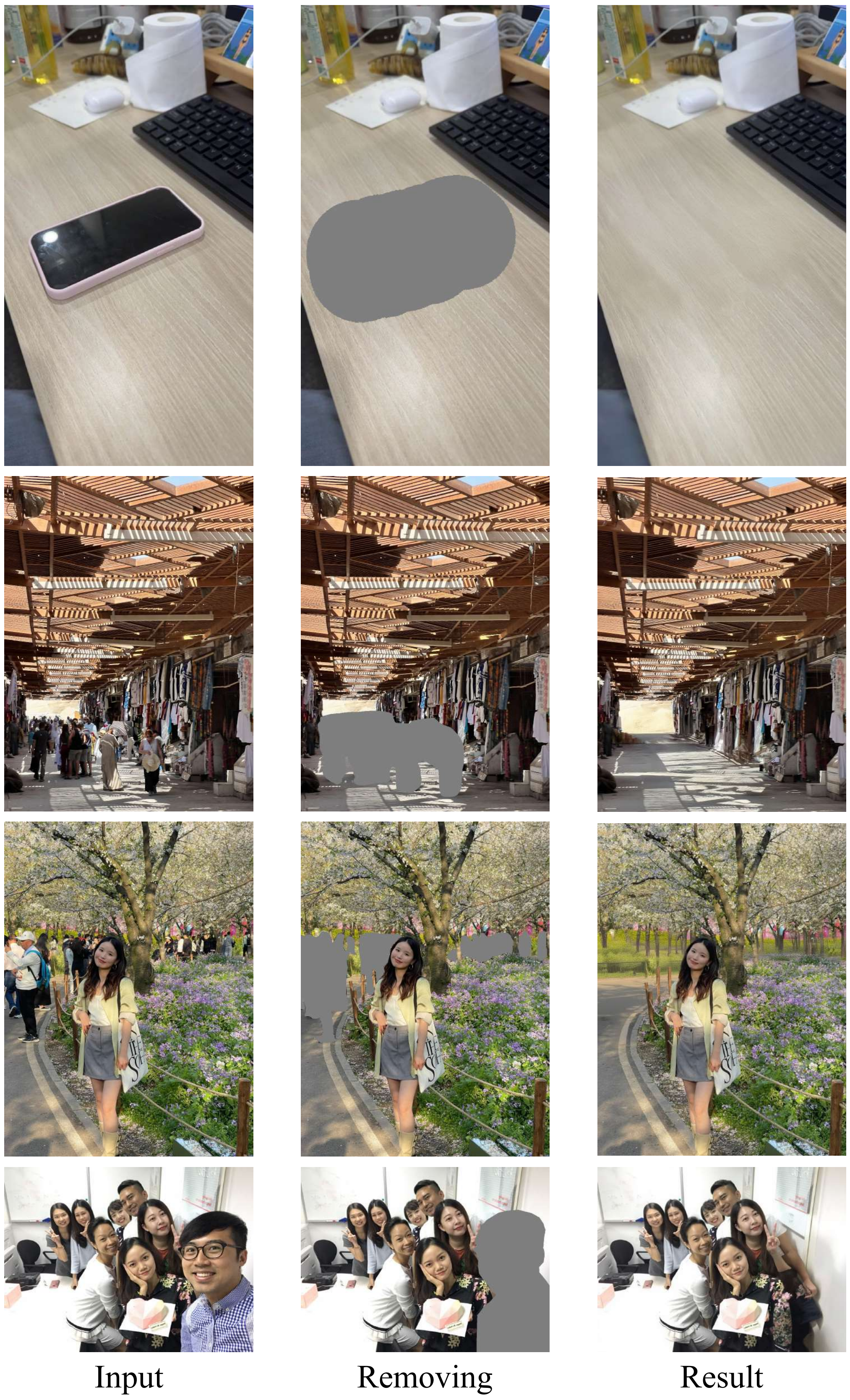

Figure 50: Supplementary object removing examples on in-the-wild images.

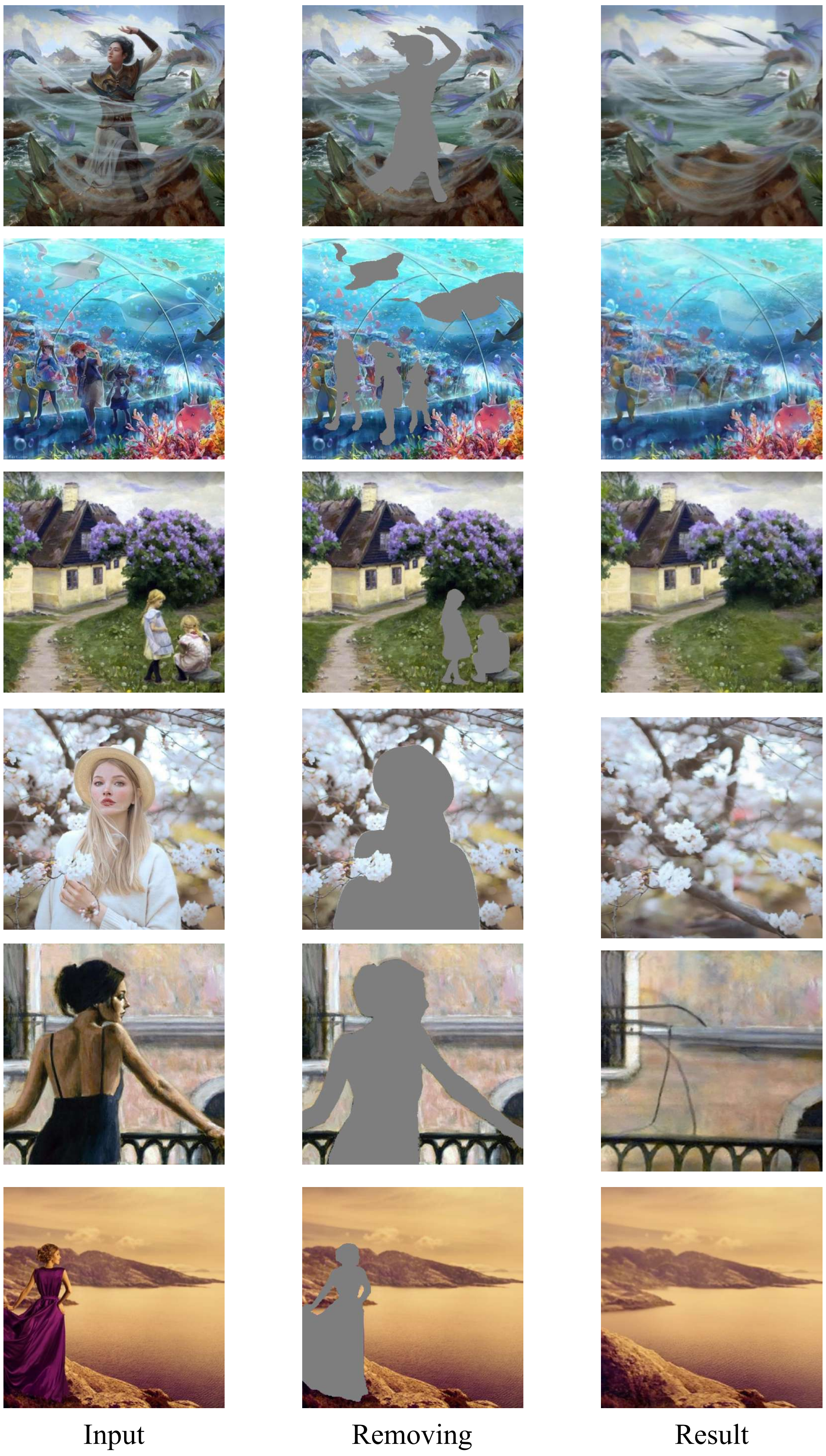

Figure 51: Supplementary object removing examples on in-the-wild images.