

A Appendix

The content of the supplementary material involves:

- Details of ColorNet in Appendix A.1.1.
- Dual-camera data sources for the ZI dataset in Appendix A.1.2.
- Details of the synthetic ZI dataset and computational cost in Appendix A.1.3.
- Formula reasoning of 3D-TPR framework in Appendix A.2.
- Comparative evaluation of virtual view synthesis method for ZI task in Appendix A.3.
- Ablation experiment of 3D-TPR framework in Appendix A.4.
- Details of the real-world multi-device ZI test sets in Appendix A.5.
- Additional visual results of OmniZoom on the real-world ZI test sets in Appendix A.6.
- Benchmarking ZI dataset across 1D, 2D, and 3D-TPR frameworks in Appendix A.7.
- Upper bound analysis of 3D-TPR framework in Appendix A.8.
- Additional visual results of 3D-TPR framework in Appendix A.9.
- Limitations in Appendix A.10.
- Broader impacts in Appendix A.11.

We provide a project page at <https://omnizoom.github.io/OmniZoom/>, where visual results are available, and the code/dataset will be released soon.

A.1 Implementation details of ZI dataset construction

A.1.1 Details of ColorNet

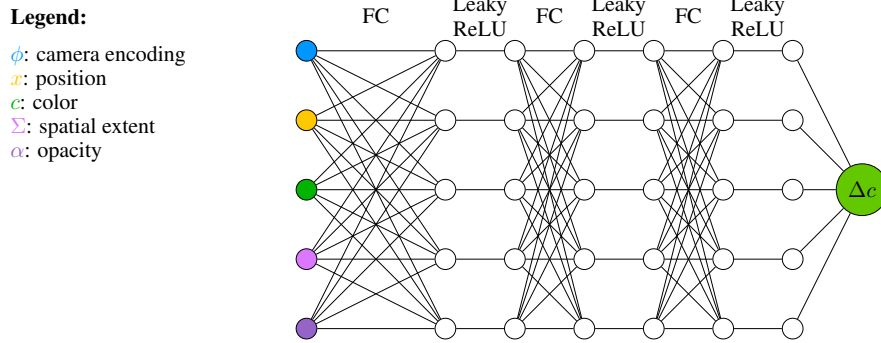


Figure 5: Architecture of ColorNet.

As illustrated in Figure 5, ColorNet is a lightweight multilayer perceptron (MLP) designed to estimate color residuals Δc for neural 3D rendering. The network takes as input a five-dimensional feature tuple: the camera encoding ϕ , the center position x , the base color c , the anisotropic covariance Σ , and the opacity α . These inputs are first projected into a latent feature space through a fully connected (FC) layer, which is then processed by three consecutive FC blocks, each followed by a LeakyReLU activation to ensure non-linearity and mitigate the risk of neuron inactivation. Each FC layer contains 5 hidden units. Finally, the output layer maps the intermediate representation to a 3-dimensional color offset vector Δc via a linear transformation. This residual is then added to the original color c , yielding the refined output color used for rendering.

A.1.2 Dual-camera data sources for the ZI dataset

As mentioned in the main paper, our ZI dataset comprises dual-camera zoom image pairs from HuaweiPura70Ultra and RedmiK50Ultra. Specifically, we collect Huawei data ourselves across a variety of indoor and outdoor scenarios, while the Redmi sequences are adopted from the publicly



Figure 6: Representative dual-camera data sources. The top row displays samples from HuaweiPura70Ultra, while the bottom row presents examples from the publicly available RedmiK50Ultra dataset [47]. Each scene contains paired wide-angle (I_w) and main-camera (I_m) images, covering diverse content and geometric configurations.

available dataset introduced in [47]. The image resolutions vary by device: 2133×1600 for Huawei and 1632×1224 for Redmi, consistent with their native imaging pipelines.

In total, the combined dataset covers 132 real-world scenes, consisting of 79 indoor and 53 outdoor environments. Indoor scenes include structured environments such as classrooms, shopping malls, and cafeterias, while outdoor scenes cover playgrounds, parks, and other open spaces with greater geometric variation. For each scene, we capture 15 pairs of main–wide camera images, along with 6 additional wide-angle views from peripheral viewpoints to enhance spatial diversity. These supplementary images increase the geometric baseline and play a key role in stabilizing 3DGS optimization. Representative examples from both devices are shown in Figure 6.

A.1.3 Details of the synthetic ZI dataset and computational cost

We generate 205 synthetic zoom sequences, each consisting of 16 interpolated frames. These sequences serve as high-quality supervision for training and evaluation in ZI tasks. Representative examples of the rendered dataset are shown in Figure 7.

The training duration for each scene depends on its visual complexity and structural content, with an average of approximately 1.5 hours per scene. Rendering a complete zoom sequence with 16 interpolated frames typically takes less than 20 seconds. All procedures are conducted on a workstation with 40 GB of memory, consistent with our main experimental setup.

A.2 Formula reasoning of 3D-TPR framework

A.2.1 Mathematical modeling and depth constraints for projective ambiguity

Modeling the projection from 3D space to 2D image plane: Let $\mathbf{P} = (X, Y, Z)^T$ denote a 3D point in the scene, which is projected onto 2D image coordinates (x, y) via the pinhole camera model. The projection is formulated as:

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}; \quad \pi(\mathbf{P}) = \left(\frac{fX}{Z}, \frac{fY}{Z} \right)^T, \quad (19)$$

where π denotes the 3D-to-2D projection function, and f represents the camera’s focal length. When an object moves through 3D space, its instantaneous displacement over a time interval Δt is governed by its 3D velocity vector $\mathbf{v} = (v_X, v_Y, v_Z)^T$:

$$\Delta \mathbf{P} = \mathbf{v} \Delta t = (v_X \Delta t, v_Y \Delta t, v_Z \Delta t)^T, \quad (20)$$

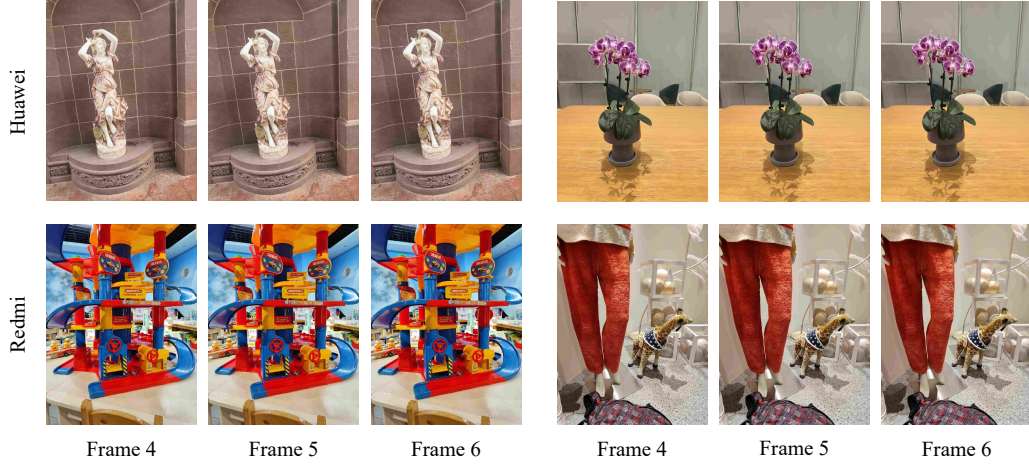


Figure 7: Representative frames (Frames 4, 5, and 6) from our synthetic ZI dataset. Each row illustrates three temporally adjacent frames sampled from a 16-frame virtual zoom sequence, demonstrating smooth transitions in both appearance and geometry. Examples are shown from Huawei and Redmi devices, covering diverse indoor and outdoor scenes with varying structural complexity.

where $\Delta \mathbf{P}$ denotes the change in 3D position during Δt . Let $I(\mathbf{P}, t)$ denote the projection of a 3D point $\mathbf{P} = (X, Y, Z)^T$ onto the image plane at time t . When the object is in motion, the position of \mathbf{P} evolves over time as:

$$\mathbf{P}(t + \Delta t) = \mathbf{P}(t) + \Delta \mathbf{P} = \mathbf{P}(t) + \mathbf{v} \Delta t. \quad (21)$$

The corresponding image plane displacement $\delta = (\Delta x, \Delta y)$ induced by 3D motion can be approximated via a first-order Taylor expansion:

$$\delta = \pi(\mathbf{P} + \mathbf{v} \Delta t) - \pi(\mathbf{P}) \approx \frac{\partial \pi}{\partial \mathbf{P}} \mathbf{v} \Delta t, \quad (22)$$

where $\frac{\partial \pi}{\partial \mathbf{P}}$ is the projection Jacobian matrix J_π , which characterizes how variations in 3D positions affect the projected 2D coordinates. The term $\mathbf{v} \Delta t$ represents the instantaneous 3D displacement, and its transformation through J_π yields the corresponding image plane motion. By substituting the partial derivatives of the projection function, the Jacobian matrix J_π can be explicitly written as:

$$J_\pi = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{fX}{Z^2} \\ 0 & \frac{f}{Z} & -\frac{fY}{Z^2} \end{bmatrix}. \quad (23)$$

Equation 23 constitutes an underdetermined system of equations, whose solution space forms a straight line. Mathematically, this can be expressed as:

$$\mathbf{v} = \mathbf{v}_0 + k \mathbf{n}, \quad \mathbf{n} = \left(\frac{X}{Z}, \frac{Y}{Z}, 1 \right)^T, \quad (24)$$

where \mathbf{v}_0 represents a particular solution, \mathbf{n} denotes the direction corresponding to the viewing ray, and $k \in \mathbb{R}$ denotes an arbitrary real number. This formulation implies that infinitely many 3D velocity vectors, which differ only in their components along the line of sight, can result in the same 2D image plane displacement. As a concrete example, consider a 3D scene point located at $\mathbf{P} = (f, 0, Z)^T$:

Scene 1. Suppose the object moves purely along the horizontal axis with the velocity vector $\mathbf{v}_1 = (1, 0, 0)^T$. The resulting image plane displacement δ_1 can be approximated via the projection Jacobian J_π as:

$$\delta_1 = J_\pi \cdot \mathbf{v}_1 \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \cdot 1 + 0 \cdot 0 + \left(-\frac{fX}{Z^2} \right) \cdot 0 \\ 0 \cdot 1 + \frac{f}{Z} \cdot 0 + \left(-\frac{fY}{Z^2} \right) \cdot 0 \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \\ 0 \end{bmatrix} \cdot \Delta t. \quad (25)$$

504 **Scene 2.** Now consider the case where the object moves along the optical axis (i.e., in depth), with
 505 the velocity vector $\mathbf{v}_2 = (0, 0, -Z/f)^T$. For the 3D point $\mathbf{P} = (f, 0, Z)^T$, the resulting image plane
 506 displacement δ_2 is computed as:

$$\delta_2 = J_\pi \cdot \mathbf{v}_2 \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \cdot 0 + 0 \cdot 0 + (-\frac{fX}{Z^2}) \cdot (-\frac{Z}{f}) \\ 0 \cdot 0 + \frac{f}{Z} \cdot 0 + (-\frac{fY}{Z^2}) \cdot (-\frac{Z}{f}) \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \\ 0 \end{bmatrix} \cdot \Delta t. \quad (26)$$

507 The distinct 3D motions illustrated in **Scene 1** and **Scene 2** become indistinguishable when projected
 508 onto the 2D image plane, thereby *leading to the non-uniqueness of motion estimation and highlighting*
 509 *the necessity of depth-direction constraints*. By enforcing a constraint along the depth direction, the
 510 solution space is reduced from a continuous line to a single, physically plausible point, effectively
 511 eliminating the ambiguity and suppressing implausible variations along the depth axis.

512 A.2.2 Gradient reallocation with texture-focus strategy

513 Let $\mathcal{S}_T = (x, y) \mid \mathbf{M}(x, y) = 1$ denote the set of texture pixels, where \mathbf{M} is the binary mask defined
 514 in Equation 14, and let \mathcal{S}_C denote its complement. According to Equation 16, the expected value of
 515 gradient norm over texture regions satisfies the following for $\lambda > 0$:

$$\mathbb{E} [\|\nabla_\theta \mathcal{L}_{\text{texture-focus}}\|_{\mathcal{S}_T}] = (1 + \lambda) \cdot \mathbb{E} [\|\nabla_\theta \mathcal{L}_{\text{base}}\|_{\mathcal{S}_T}]. \quad (27)$$

516 This formulation indicates that the texture-focus loss reweights the gradient flow to amplify super-
 517 vision in structurally informative regions. By scaling the gradient magnitude in \mathcal{S}_T by a factor of
 518 $(1 + \lambda)$, the network is encouraged to prioritize learning from texture-rich areas, which are typically
 519 more perceptually salient and semantically meaningful.

520 While expected value of the gradient norm within the texture region \mathcal{S}_T is magnified by a factor of
 521 $(1 + \lambda)$, the gradient norm in the non-texture region \mathcal{S}_C remains unchanged:

$$\mathbb{E} [\|\nabla_\theta \mathcal{L}_{\text{texture-focus}}\|_{\mathcal{S}_C}] = \mathbb{E} [\|\nabla_\theta \mathcal{L}_{\text{base}}\|_{\mathcal{S}_C}]. \quad (28)$$

522 Consequently, the texture-focus strategy effectively redistributes the gradient energy in the parameter
 523 space by selectively amplifying directions associated with texture regions, thereby enhancing supervi-
 524 sion in structurally informative areas. Under momentum-based stochastic gradient descent (SGD),
 525 the network parameters are updated as:

$$\theta_{k+1} = \theta_k - \eta \mathbf{v}_k, \quad \mathbf{v}_k = \beta \mathbf{v}_{k-1} + (1 - \beta) \nabla_\theta \mathcal{L}_{\text{texture-focus}}(\theta_k), \quad (29)$$

526 where θ_k denotes the network parameters at the k -th iteration, and θ_{k+1} represents the updated
 527 parameters after applying a weighted update in the direction of the momentum term \mathbf{v}_k . Here, η is
 528 the learning rate, and \mathbf{v}_k is the accumulated momentum, which integrates the current gradient with
 529 past updates. We set the momentum coefficient to $\beta = 0.9$, which determines the proportion of the
 530 previous momentum \mathbf{v}_{k-1} retained in the current step. The gradient term $\nabla_\theta \mathcal{L}_{\text{texture-focus}}(\theta_k)$ is
 531 computed with respect to the texture-focus loss at iteration k .

532 Since the gradient of $\mathcal{L}_{\text{texture-focus}}$ within the texture region \mathcal{S}_T is scaled by a factor of $(1 + \lambda)$,
 533 the corresponding entries in the gradient tensor exhibit proportionally larger magnitudes. This
 534 induces an optimization bias toward texture-dense regions, thereby encouraging edge sharpening
 535 and the recovery of high-frequency details. In contrast, gradients within the non-texture region \mathcal{S}_C
 536 remain unchanged. This targeted reallocation strategy prioritizes learning in structurally salient areas,
 537 ultimately enhancing the perceptual fidelity and fine-detail quality of the interpolated frames.

538 Our proposition formally demonstrates that explicitly amplifying momentum components along
 539 high-frequency texture directions offers two key benefits: (a) it reduces the number of iterations
 540 required to reach local optima by modulating the effective learning rate, and (b) it suppresses
 541 oscillatory behaviors in the loss surface caused by the competing objectives of texture underfitting
 542 and background overfitting.

543 Under the flat-minima hypothesis for over-parameterized networks, this gradient reallocation strategy
 544 further flattens the loss landscape during finetuning, acting as an implicit regularizer that improves
 545 generalization to unseen texture distributions through curvature-driven optimization dynamics. Unlike
 546 perceptual losses that entangle high- and low-frequency signals within the shared feature space of
 547 pretrained encoders, our texture-focus strategy achieves frequency disentanglement via pixel-level
 548 gradient modulation in an encoder-agnostic manner, allowing more precise control over structural
 549 detail reconstruction only with an ancillary supervision.

550 A.2.3 Additional proof of mask penalty strategy

551 Normalization coefficient $4 \cdot \mathbf{M}(1 - \mathbf{M}) \in [0, 1]$, reaching the maximum peak value at $\mathbf{M} = 0.5$. As
 552 $\mathbf{M} \rightarrow 0$ or $\rightarrow 1$, the normalization coefficient value tends to zero. From a Bayesian perspective, the
 553 mask value $\mathbf{M}_p^{(i)}$ is modelled as the parameter $\theta_p^{(i)}$ of the posterior Bernoulli distribution \mathcal{Q} :

$$(\mathbf{M}_p^{(i)} = \theta_p^{(i)}) \sim (\mathcal{Q} = \text{Bernoulli}(\theta_p^{(i)})), \quad (30)$$

554 where $\mathbf{M}_p^{(i)}$ represents the model-inferred confidence at pixel position p of the i -th scale, which is
 555 calculated by the middle layer of the network and equivalent to $\theta_p^{(i)}$. $\theta_p^{(i)}$ is the parameter of the
 556 Bernoulli distribution, representing the probability that the mask value is 1 at position p , indicating
 557 the network’s confidence that the current pixel is dominated by the features from I_0 .

558 The case $\theta_p^{(i)} = 1$ indicates that the mask value at position p is fully dominated by I_0 , whereas
 559 $\theta_p^{(i)} = 0$ implies complete dominance by I_1 . $\theta_p^{(i)} = 0.5$ shares the equal contributions from both two
 560 reference frames.

561 The interpolation network dynamically adjusts the weights by learning the distribution of $\theta_p^{(i)}$. Since
 562 $\theta_p^{(i)}$ is directly adopted as the parameter of the Bernoulli distribution, its entropy $\mathcal{H}(\theta_p^{(i)})$ quantifies
 563 the uncertainty in the fusion decision:

$$\mathcal{H}(\mathcal{Q}) = \mathcal{H}(\theta_p^{(i)}) = -\theta_p^{(i)} \log \theta_p^{(i)} - (1 - \theta_p^{(i)}) \log(1 - \theta_p^{(i)}). \quad (31)$$

564 A higher entropy value indicates greater uncertainty in the network’s fusion decision at the current
 565 pixel, typically when $\theta_p^{(i)}$ approaches 0.5. In contrast, lower entropy represents higher confidence,
 566 corresponding to $\theta_p^{(i)}$ values close to 0 or 1. Assume that an implicit Beta prior distribution \mathcal{P} is
 567 applied to $\theta_p^{(i)}$:

$$\mathcal{P}(\theta_p^{(i)}) \propto \theta_p^{\alpha-1} (1 - \theta_p)^{\beta-1}, \quad (32)$$

568 where we adopt a symmetric, U-shaped configuration by setting $\alpha = \beta < 1$. The probability density
 569 is higher at $\theta_p = 0$ or $\theta_p = 1$ and lower at median $\theta_p = 0.5$, thereby encoding prior knowledge that
 570 induces a binarization tendency. Minimizing the KL divergence between the posterior distribution \mathcal{Q}
 571 and the prior distribution \mathcal{P} yields:

$$D_{KL}(\mathcal{Q}||\mathcal{P}) = \mathcal{H}(\mathcal{Q}, \mathcal{P}) - \mathcal{H}(\mathcal{Q}), \quad (33)$$

572 where $\mathcal{H}(\mathcal{Q}, \mathcal{P}) = -\mathbb{E}_{\mathcal{Q}}[\log \mathcal{P}(\theta_p^{(i)})]$ measures the cross-entropy between the posterior distribution
 573 \mathcal{Q} and the prior distribution \mathcal{P} , reflecting how well the posterior aligns with the prior. A smaller
 574 value indicates that \mathcal{Q} aligns more closely with the distributional preferences encoded in \mathcal{P} . $\mathcal{H}(\mathcal{Q})$
 575 quantifies the intrinsic uncertainty of \mathcal{Q} , with a higher value indicating increased ambiguity in
 576 decision-making.

577 By designing $\mathcal{L}_{mask}^{(i)} \propto \theta_p^{(i)}(1 - \theta_p^{(i)})$, as derived in Equation 18, we implicitly impose a gradient
 578 direction consistent with the low-entropy Beta prior distribution \mathcal{P} . When $\theta_p^{(i)}$ approaches to 0.5,
 579 the gradient of $\mathcal{H}(\mathcal{Q}, \mathcal{P})$ dominates the optimization direction since $\log \mathcal{P}(\theta_p^{(i)})$ tends to be negative
 580 infinity. In that case, $\theta_p^{(i)}$ will be forced to flee quickly from high-uncertainty areas, in alignment
 581 with the gradient of $\mathcal{L}_{mask}^{(i)}$. Once $\theta_p^{(i)}$ enters the vicinity of 0 or 1, the gradient of $\mathcal{H}(\mathcal{Q}, \mathcal{P})$
 582 decays, and the gradient of $\mathcal{H}(\mathcal{Q})$ begins to dominate. At this point, the gradient of $\mathcal{L}_{mask}^{(i)}$ remains
 583 persistently operative throughout the optimization process, thus ensuring stable convergence of the
 584 model. Consequently, minimizing $\mathcal{L}_{mask}^{(i)}$ effectively approximates the minimizing KL divergence
 585 $D_{KL}(\mathcal{Q}||\mathcal{P})$. This design is theoretically anchored in the Bayesian framework and promotes training
 586 stability through a principled simplification of the target loss.

587 A.2.4 Overall training loss

588 To sum up, the overall training target loss is:

$$\mathcal{L}_{overall} = \alpha_T \cdot \mathcal{L}_{texture-focus} + \alpha_m \cdot \sum_{i=1}^S \mathcal{L}_{mask}^{(i)} + \alpha_o \cdot \mathcal{L}_{original}, \quad (34)$$



Figure 8: Visual comparison of virtual frame generation methods for ZI supervision. Compared to ZoomGS [47] and 3DGS [21], our method produces intermediate frames with fewer artifacts and better detail preservation, demonstrating superior supervision quality for training ZI models.

where αT , α_m , and α_o denote the weighting coefficients for the texture-focus loss, mask regularization loss, and baseline supervision loss, respectively. These weights can be manually specified or adaptively optimized during training, depending on the architectural design and optimization dynamics of the model.

A.3 Comparative evaluation of virtual view synthesis method for ZI task

A.3.1 Comparison of virtual frame generation quality

To evaluate the effectiveness of synthetic supervision for the ZI task, we compare three virtual frame generation methods: ZoomGS [47], 3DGS [21], and our proposed pipeline. For a fair comparison, all baselines are reproduced using their official implementations and default hyperparameters. As shown in Figure 8, our method yields noticeably higher-quality intermediate views, exhibiting sharper textures, reduced artifacts, and more accurate geometric consistency. In contrast, ZoomGS and 3DGS often exhibit geometric distortion and texture degradation, particularly in scenes with complex spatial layouts or high-frequency surface details. These findings highlight the superiority of our pipeline in producing visually reliable and semantically aligned supervision tailored for ZI training.

A.3.2 Evaluating virtual supervision via ZI finetuning

To assess the effectiveness of our synthetic supervision in real-world ZI scenarios, we conduct a downstream finetuning evaluation against two baselines: ZoomGS [47] and 3DGS [21]. For each method, the corresponding virtual dataset is used to finetune four representative FI networks, all integrated into our unified 3D-TPR framework. The resulting models are then evaluated on real-world test sets captured from four distinct smartphone platforms, enabling a comprehensive assessment of cross-device generalization performance.

Quantitative results. Table 3 reports results on Huawei and Redmi devices, both of which are included in all three synthetic ZI datasets. Our method consistently outperforms ZoomGS and 3DGS across all networks and metrics, demonstrating strong supervision quality in seen-device scenarios. While ZoomGS performs reasonably due to its device-specific training coverage, it lacks explicit spatial and color modeling. In contrast, our spatial transition and dynamic color adaptation modules effectively handle cross-camera misalignment and photometric variation, resulting in sharper textures and more coherent frame synthesis, reflected by gains in CLIP-IQA and MUSIQ. 3DGS also employs Huawei and Redmi data, but builds on monocular reconstruction assumptions and ignores

Table 3: Quantitative comparison on real-world dual-camera test sets captured by Huawei and Redmi devices, which are included in the synthetic dataset construction. Each FI model is evaluated under four supervision variants: original 3D-TPR pretrained (Base), and finetuned on 3DGS, ZoomGS, and our proposed supervision f . **Bold** indicates the best result, and underline denotes the second best.

Networks	Methods	HuaweiPura70Pro				RedmiK50Ultra			
		NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑	NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑
RIFE	Base	3.8651	4.1537	0.4939	58.8362	4.8764	5.0247	0.4664	<u>61.9235</u>
	3DGS $_f$	3.8309	4.2570	<u>0.5183</u>	<u>60.2724</u>	4.8457	5.1787	0.3459	52.4562
	ZoomGS $_f$	<u>3.7902</u>	<u>4.0084</u>	0.5013	59.0780	<u>4.5598</u>	<u>4.7340</u>	<u>0.4716</u>	61.2694
	Ours $_f$	3.7422	3.9360	0.5233	60.8585	4.4720	4.5195	0.4930	63.6990
IFRNet	Base	3.4852	3.3150	0.5784	73.0233	4.2837	3.5993	0.4913	73.7573
	3DGS $_f$	3.5310	<u>3.2570</u>	0.5183	70.2725	4.2232	3.5776	0.4259	70.0659
	ZoomGS $_f$	3.5132	3.2769	<u>0.5867</u>	<u>74.0218</u>	<u>4.0879</u>	<u>3.3928</u>	<u>0.5127</u>	<u>74.2312</u>
	Ours $_f$	<u>3.5035</u>	3.2520	0.5909	74.1220	4.0695	3.3880	0.5152	74.3871
EMA-VFI	Base	3.8470	3.8566	<u>0.5621</u>	<u>62.7403</u>	4.3424	4.5974	0.4807	59.6610
	3DGS $_f$	3.8888	3.4184	0.5540	62.7141	4.2464	4.4593	<u>0.4908</u>	59.2312
	ZoomGS $_f$	3.6360	3.8499	0.5229	61.2508	<u>4.2106</u>	<u>4.4610</u>	0.4718	<u>60.1914</u>
	Ours $_f$	<u>3.8341</u>	<u>3.8467</u>	0.5684	63.5048	3.8852	4.5519	0.4947	60.7579
AMT	Base	3.6459	3.6132	0.5498	71.7016	4.7426	4.0235	0.4754	71.9982
	3DGS $_f$	3.4894	3.3929	0.5184	70.5856	4.3934	3.7834	0.4646	71.4254
	ZoomGS $_f$	<u>3.4593</u>	<u>3.3395</u>	<u>0.5847</u>	<u>73.7058</u>	<u>4.0653</u>	<u>3.4472</u>	<u>0.5209</u>	74.4339
	Ours $_f$	3.4547	3.3121	0.6018	73.7092	3.9835	3.3548	0.5383	<u>74.3325</u>

Table 4: Quantitative evaluation of cross-device generalization on iPhone and OPPO, two unseen devices excluded from the synthetic dataset generation. Despite the domain shift, models finetuned with our supervision consistently outperform those trained with 3DGS and ZoomGS. **Bold** indicates the best result, and underline denotes the second best.

Networks	Methods	iPhone16ProMax				OPPOFindX8Ultra			
		NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑	NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑
RIFE	Base	4.1027	4.3503	0.5366	<u>59.6800</u>	4.6100	5.0104	0.5366	65.7912
	3DGS $_f$	3.9618	4.2914	0.4687	54.5013	4.4927	4.9957	0.4408	60.6637
	ZoomGS $_f$	<u>3.8950</u>	<u>4.1679</u>	<u>0.5411</u>	59.6339	<u>4.4543</u>	<u>4.8111</u>	<u>0.5779</u>	<u>66.6404</u>
	Ours $_f$	3.8601	4.0657	0.5492	60.7988	4.3968	4.6815	0.5830	67.1494
IFRNet	Base	<u>3.6923</u>	3.3942	0.5829	73.3569	5.2039	4.9603	0.5212	75.2718
	3DGS $_f$	3.6233	<u>3.3400</u>	0.5583	71.7621	<u>5.1306</u>	4.8809	0.4755	74.3894
	ZoomGS $_f$	3.7501	3.3498	<u>0.5941</u>	<u>73.7598</u>	5.1630	<u>4.7927</u>	<u>0.5364</u>	<u>75.3461</u>
	Ours $_f$	3.6953	3.3165	0.5949	73.8292	5.1133	4.7144	0.5461	75.4545
EMA-VFI	Base	4.0555	4.1304	0.5387	58.6204	<u>4.5166</u>	4.9022	0.5590	67.5921
	3DGS $_f$	3.6258	3.8183	<u>0.5418</u>	59.2020	4.8225	4.9720	0.5868	66.8282
	ZoomGS $_f$	<u>3.8068</u>	<u>3.9542</u>	0.5377	<u>59.5872</u>	4.5455	<u>4.8818</u>	0.5551	68.2806
	Ours $_f$	4.0934	4.1123	0.5503	60.1002	4.5071	4.8752	<u>0.5591</u>	<u>67.6444</u>
AMT	Base	3.7053	3.5367	0.5930	<u>72.1243</u>	5.3017	5.3808	0.4518	72.1563
	3DGS $_f$	3.6991	3.4205	0.5644	71.6435	4.9154	5.0089	0.4639	72.5701
	ZoomGS $_f$	<u>3.5192</u>	<u>3.4044</u>	<u>0.5985</u>	71.8655	<u>4.6889</u>	<u>4.7544</u>	<u>0.5581</u>	<u>72.6175</u>
	Ours $_f$	3.5087	3.2661	0.6211	73.6150	4.6699	4.5702	0.5595	75.2187

618 dual-camera geometric or ISP discrepancies. Consequently, the composite supervision shows less
619 consistency, which universally degrades finetuning efficacy on all tasks.

620 Table 4 evaluates generalization to iPhone and OPPO: two unseen devices that are not involved in
621 training or synthetic data generation. Despite the domain gap, our method consistently achieves the
622 best performance across all networks and evaluation metrics. In contrast, ZoomGS and 3DGS exhibit
623 clear performance degradation due to their limited capacity to model cross-device geometric and

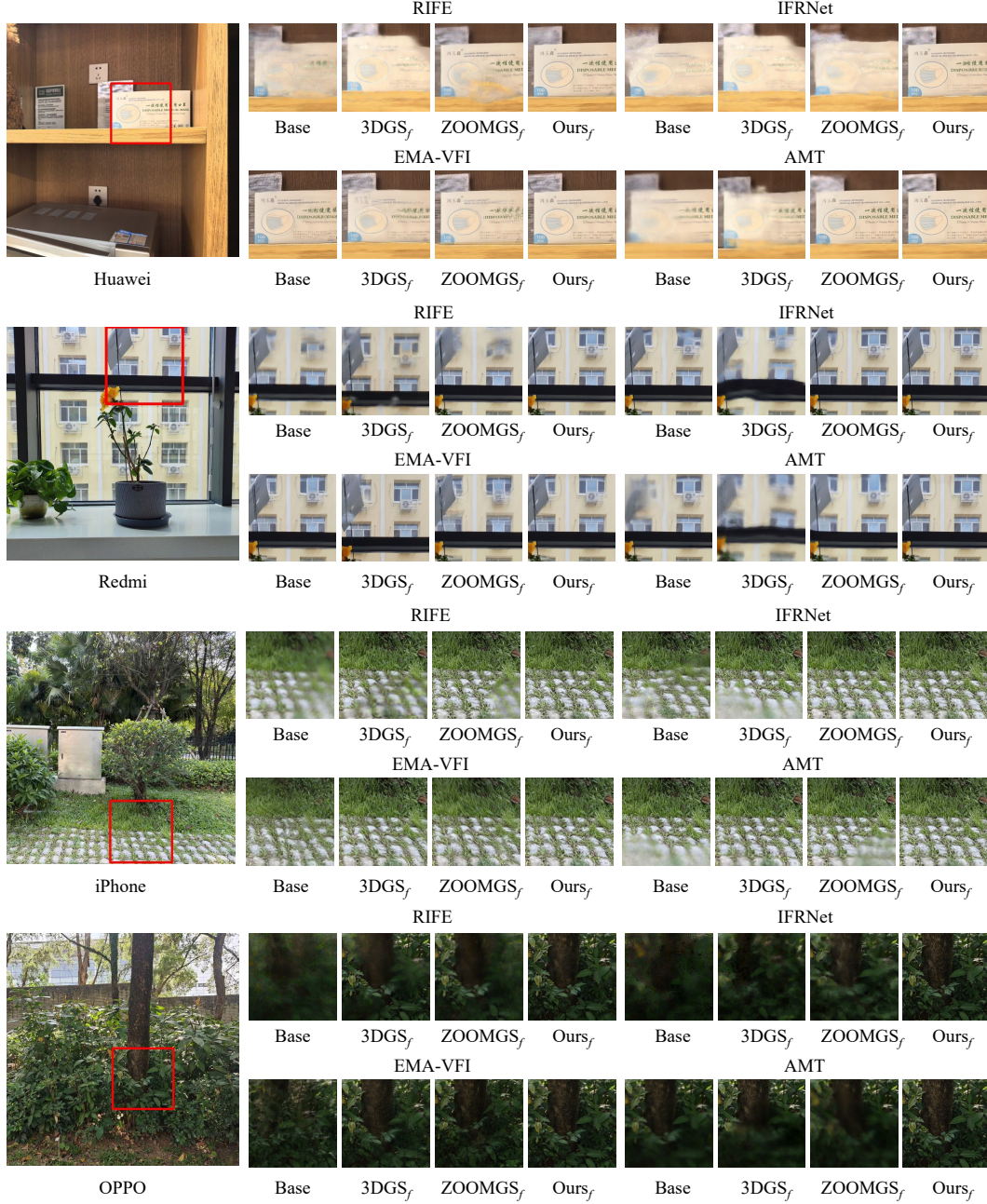


Figure 9: Qualitative comparison of finetuned interpolation results on real-world ZI tasks across four smartphone platforms (Huawei, Redmi, iPhone, and OPPO) and four FI networks. We compare models finetuned with virtual supervision from 3DGS, ZoomGS, and our proposed method, alongside unfinetuned baselines ("Base") implemented within the unified 3D-TPR framework. Our method consistently produces sharper structures and fewer artifacts. Notably, text contours are better preserved on Huawei, boundary lines are more clearly delineated on Redmi, and perceptual blurring is substantially reduced on iPhone and OPPO.

624 photometric variations. These results highlight the strong generalization ability of our pipeline, which
 625 provides robust, device-agnostic supervision across heterogeneous smartphone platforms.

626 Our method consistently outperforms existing virtual frame generation pipelines across both seen and
 627 unseen device scenarios. These gains are attributed to the explicit modeling of cross-device geometric

transitions and photometric inconsistencies, two key factors for ensuring structurally accurate and perceptually consistent zoom interpolation in real-world.

Qualitative results. We further perform a qualitative comparison of interpolation outcomes across four FI networks and four smartphone platforms in real-world ZI tasks. As shown in Figure 9, we visualize the results of models finetuned with three types of synthetic supervision: ZoomGS, 3DGS, and our proposed approach, alongside the original 3D-TPR pretrained baselines without finetuning. ZoomGS_f performs reasonably on Huawei and Redmi devices, which are included in its training dataset. However, it still suffers from noticeable blurring and diminished structural fidelity, particularly around object boundaries and fine textures. 3DGS_f, which lacks explicit device-specific modeling, exhibits limited generalization and introduces geometric distortions and color artifacts across all devices. In contrast, our method consistently generates sharper structures, more coherent textures, and significantly fewer artifacts. For instance, text contours appear more defined on Huawei, linear boundaries are better preserved on Redmi, and perceptual blurring is substantially reduced on OPPO and iPhone. These qualitative observations further underscore the importance of modeling device geometric and photometric characteristics, and highlight the superior visual fidelity enabled by our synthetic supervision pipeline.

Overall. Both quantitative and qualitative results consistently demonstrate the superiority of our synthetic supervision across a wide range of FI networks and smartphone platforms. By explicitly modeling cross-device geometric misalignment and photometric variation, our method enables more effective finetuning, yielding improvements in perceptual quality, structural fidelity, and generalization to unseen devices.

A.4 Ablation experiment of 3D-TPR framework

To evaluate the effectiveness and compatibility of each component in our 3D-TPR framework, we conduct a controlled ablation study across four representative FI networks using a unified training protocol. Specifically, we compare four configurations: (1) **3D-t-m**, which applies only the 3D-TPR encoding; (2) **3D-m**, which combines 3D-TPR with the texture-focus strategy; (3) **3D-t**, which incorporates the mask penalty constraint; and (4) **3D**, the full model with all strategies enabled. As shown in Table 5, this setup enables a clear analysis of the individual and combined effects of each component on interpolation performance.

Effect of texture-focus strategy. Introducing the texture-focus strategy consistently improves both pixel-level fidelity and perceptual quality across most networks. Compared to using 3D-TPR encoding alone, the 3D-m configuration yields small but stable gains in PSNR and SSIM, and reduced LPIPS in several cases. These improvements suggest that focusing optimization on high-frequency, salient textures can better preserve structural details. Notably, this component requires no architectural changes and uses a fixed weight of 5.0, making it efficient and easily adoptable.

Effect of mask penalty constraint. The mask penalty constraint (3D-t) introduces a mild regularization that encourages confident, low-entropy fusion decisions. While the numerical improvements are modest, we observe a consistent reduction in perceptual artifacts, especially in networks such as EMA-VFI. This suggests that the constraint is particularly beneficial in ambiguous regions, guiding the network to make more reliable fusion predictions. The loss is scaled to $0.1 \times \mathcal{L}_{base}$, ensuring minimal disruption to structural learning while enhancing perceptual robustness.

Effect of full configuration. While each component individually contributes modest improvements, their integration in the full configuration (3D) leads to consistent and cumulative gains across all four networks. Compared to the 2D baseline and partial variants, the full model achieves higher PSNR and SSIM in nearly all cases, along with lower LPIPS in three out of four networks. These results highlight the complementary nature of the proposed modules, which jointly enhance structural fidelity, perceptual clarity, and temporal consistency. Importantly, these improvements are achieved without any architectural modifications or additional inference cost, supporting the practicality and plug-and-play compatibility of the 3D-TPR framework. Although the absolute gains may appear small, their consistency across diverse architectures demonstrates the robustness and generalization ability of our design. Furthermore, when provided with high-quality disparity supervision (see Appendix A.8), the framework achieves substantially larger gains, indicating that 3D-TPR has strong potential when supported by more accurate geometric priors.

Table 5: Ablation study of the proposed 3D-TPR framework across four FI networks. The 2D framework [55] serves as the baseline. **3D-t-m** denotes the use of 3D-TPR encoding only; **3D-m** adds the texture-focus strategy; **3D-t** incorporates the mask penalty constraint; and **3D** integrates all components. **Bold** values indicate improvements over the 2D baseline. Each component contributes incremental gains, and the full model yields consistent performance boosts across metrics and networks.

Network	RIFE			IFRNet			EMA-VFI			AMT		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2D	27.4050	0.9010	0.0863	27.1267	0.8994	0.0780	24.7313	0.8514	0.081	27.1726	0.9017	0.0807
3D-t-m	27.5050	0.9020	0.0814	27.1750	0.9002	0.0756	24.7804	0.8519	0.0811	27.2197	0.9021	0.0842
3D-m	27.5102	0.9023	0.0811	27.2102	0.9008	0.0744	24.7444	0.8520	0.0838	27.1345	0.9011	0.0814
3D-t	27.4130	0.9001	0.0812	27.0775	0.8976	0.0746	24.8030	0.8527	0.0809	27.1061	0.9004	0.0844
3D	27.4653	0.9016	0.0812	27.1561	0.8988	0.0715	24.8573	0.8532	0.0799	27.1909	0.9019	0.0836

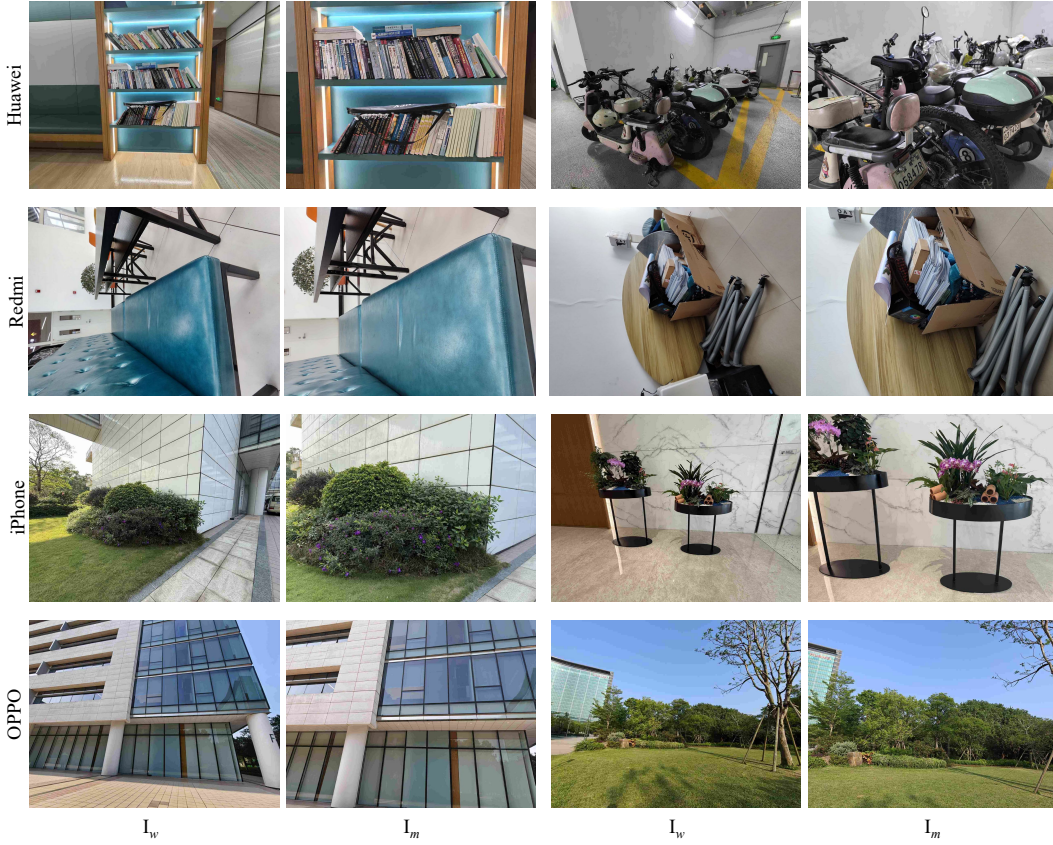


Figure 10: Representative dual-camera samples from our real-world multi-device ZI test sets. Each row corresponds to a different smartphone platform (Huawei, Redmi, iPhone and OPPO), showing paired zoom images captured under diverse scenes and lighting conditions. For layout consistency, RedmiK50Ultra samples are rotated 90 degrees counterclockwise.

681 A.5 Details of the real-world multi-device ZI test sets

682 To comprehensively evaluate the generalization ability of different ZI methods, we construct real-
683 world multi-device test sets spanning four mainstream smartphone platforms: HuaweiPura70Ultra,
684 RedmiK50Ultra, OPPOFindX7Ultra, and iPhone16ProMax. For each device, we collect a substantial
685 number of dual-camera zoom sequences under diverse environmental conditions and usage scenarios.

The complete test sets consist of 499 sequences, including 160 from Huawei, 69 from OPPO, 170 from iPhone, and 100 from Redmi. These sequences span a broad range of real-world scenes, such as indoor environments (e.g., offices, shopping malls, cafés) and outdoor locations (e.g., parks, sidewalks, playgrounds), with variations in illumination, motion intensity, and zoom transitions to reflect practical deployment conditions.

Image resolution varies by device: Huawei (2580×1560), Redmi (1216×1632), iPhone (2016×1512), and OPPO (2048×1536). Each sequence captures paired wide and main camera images along the zoom trajectory, providing a challenging and diverse benchmark for evaluating ZI methods. Representative samples from the test sets are shown in Figure 10.

A.6 Additional visual results of OmniZoom on the real-world ZI test sets

To further evaluate the generalization capability of our method, we present additional visual comparisons on the real-world ZI test sets. Specifically, we compare 3D-TPR models before and after finetuning on the ZI dataset across four smartphone platforms. As shown in Figure 11, the finetuned model produces more temporally consistent and perceptually sharper interpolations, particularly in challenging regions involving parallax, fine textures, or low-light conditions. These results further demonstrate the effectiveness of our ZI dataset in enhancing cross-device generalization for real-world zoom interpolation.

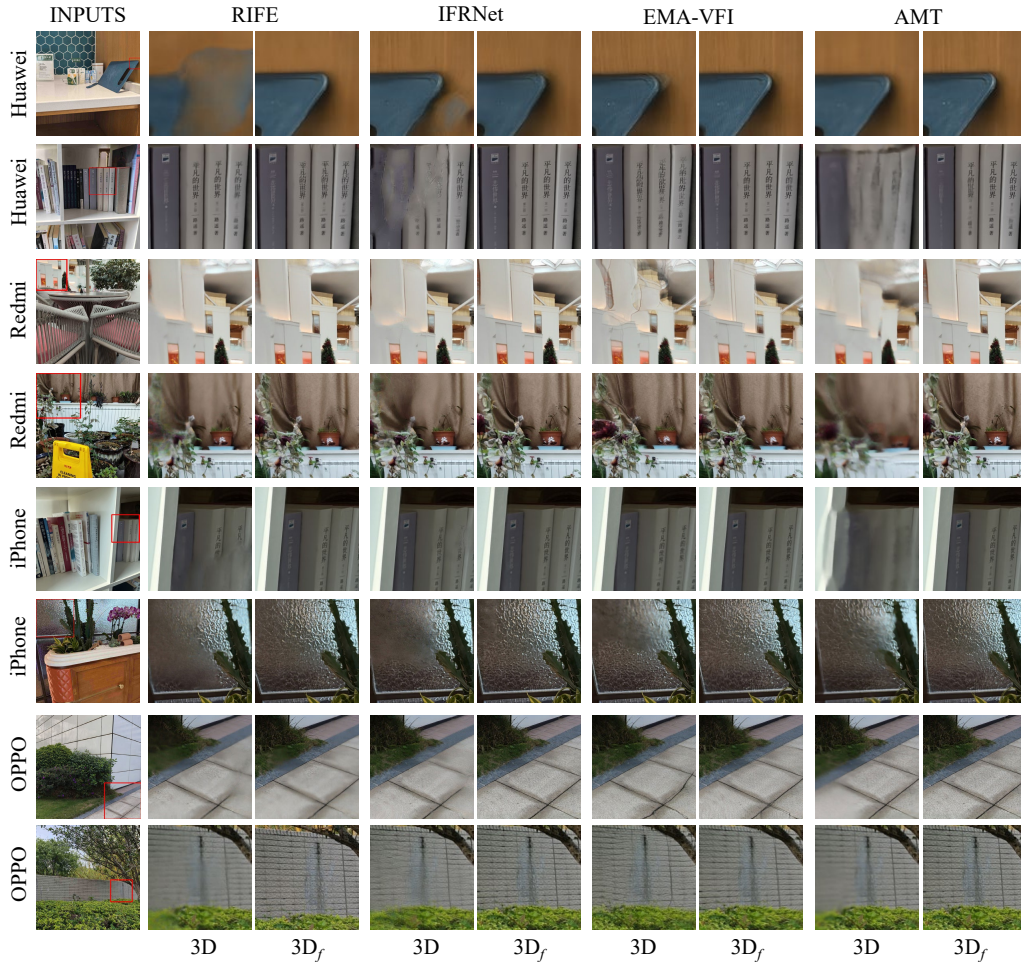


Figure 11: Additional visual comparisons on real-world ZI test data from four smartphone platforms. Subscript _f indicates models finetuned on our ZI dataset. Finetuned models produce sharper structures and improved geometric consistency across diverse scenes.

Table 6: Quantitative result of 1D, 2D, and 3D-TPR frameworks across four FI models and four devices. All models are evaluated before and after finetuning on our ZI dataset. **Bold** indicates the best performance after finetuning, and underline marks the best result before finetuning. 3D-TPR framework reliably achieves superior perceptual quality, and its finetuned variant (3D_f) outperforms others in most cases, demonstrating the effectiveness of our ZI dataset and the 3D-TPR design.

Device	Metrics	RIFE						IFRNet					
		1D	2D	3D	1D _f	2D _f	3D _f	1D	2D	3D	1D _f	2D _f	3D _f
Huawei	NIQE↓	3.9181	<u>3.8464</u>	3.8651	3.8678	3.8170	3.7422	3.6995	3.6801	<u>3.4852</u>	3.6044	3.6455	3.5035
	PI↓	4.2567	4.2505	<u>4.1537</u>	4.0698	4.0599	3.9360	4.0141	3.9570	<u>3.3150</u>	3.6448	3.6136	3.2520
	CLIP-IQA↑	0.4078	0.3691	<u>0.4939</u>	0.4573	0.4858	0.5233	0.4910	0.5422	<u>0.5784</u>	0.5649	0.5770	0.5909
	MUSIQ↑	49.4314	44.8268	<u>58.8362</u>	58.4623	58.7098	60.8585	51.2033	57.4632	<u>73.0233</u>	61.2380	63.1001	74.1220
Redmi	NIQE↓	5.2253	5.0138	<u>4.8764</u>	4.6925	4.5118	4.4720	5.3165	5.0098	<u>4.2837</u>	4.6223	4.5292	4.0695
	PI↓	5.6387	5.3615	<u>5.0247</u>	4.8289	4.6331	4.5195	5.5147	5.1108	<u>3.5993</u>	4.6454	4.5053	3.3880
	CLIP-IQA↑	0.3661	0.4077	<u>0.4664</u>	0.4559	0.4851	0.4930	0.3691	0.4219	<u>0.4913</u>	0.4765	0.4947	0.5152
	MUSIQ↑	51.7491	56.3870	<u>61.9235</u>	60.6873	62.0812	63.6990	51.3791	57.5453	<u>73.7573</u>	61.7805	62.4252	74.3871
iPhone	NIQE↓	4.3433	4.2031	<u>4.1027</u>	3.9339	3.9532	3.8601	4.0136	3.8786	<u>3.6923</u>	3.7891	3.8169	3.6953
	PI↓	4.6718	4.5340	<u>4.3503</u>	4.1674	4.2135	4.0657	4.4137	4.2090	<u>3.3942</u>	4.0014	3.9636	3.3165
	CLIP-IQA↑	0.4476	0.4821	<u>0.5363</u>	0.5371	0.5345	0.5492	0.4814	0.5240	<u>0.5829</u>	0.5666	0.5778	0.5949
	MUSIQ↑	52.2715	55.0577	<u>59.6800</u>	59.2367	58.8978	60.7988	52.7781	57.4088	<u>73.3569</u>	60.8860	61.9734	73.8292
OPPO	NIQE↓	5.2103	4.6364	<u>4.6100</u>	5.0142	4.5575	4.3968	5.9164	4.5820	5.2039	5.6720	5.7078	5.1133
	PI↓	5.4442	5.0848	<u>5.0104</u>	5.0965	5.0650	4.6815	5.1909	4.9749	<u>4.9603</u>	5.0287	4.8435	4.7144
	CLIP-IQA↑	0.4270	0.4989	<u>0.5366</u>	0.5450	0.5699	0.5830	0.5181	<u>0.5525</u>	0.5212	0.5207	0.5667	0.5461
	MUSIQ↑	54.8990	63.2645	<u>65.7912</u>	63.3583	65.0204	67.1494	54.8757	67.2635	<u>75.2718</u>	64.9371	67.2739	75.4545
Device	Metrics	EMA-VFI						AMT					
		1D	2D	3D	1D _f	2D _f	3D _f	1D	2D	3D	1D _f	2D _f	3D _f
Huawei	NIQE↓	4.6877	<u>3.6363</u>	3.8470	3.7737	3.4852	3.8341	4.7231	<u>3.5665</u>	3.6459	3.9866	3.6246	3.4547
	PI↓	3.7060	<u>3.7240</u>	3.8566	3.6638	3.7186	3.8467	4.9499	<u>3.5147</u>	3.6132	3.9974	3.4639	3.3121
	CLIP-IQA↑	0.5619	0.5612	<u>0.5621</u>	0.5588	0.5535	0.5684	0.3127	0.5428	<u>0.5498</u>	0.5898	0.5944	0.6018
	MUSIQ↑	60.4395	61.7263	<u>62.7403</u>	62.8008	63.0230	63.5048	53.6305	71.4369	<u>71.7016</u>	73.4803	72.7456	73.7092
Redmi	NIQE↓	4.5060	4.4088	<u>4.3424</u>	4.4864	4.3537	3.8852	5.9112	5.1925	<u>4.7426</u>	4.9700	5.1082	3.9835
	PI↓	4.6410	4.6708	<u>4.5974</u>	4.5829	4.5665	4.5519	5.0623	5.4335	<u>4.0235</u>	4.8863	5.0036	3.3548
	CLIP-IQA↑	0.4651	0.4599	<u>0.4807</u>	0.4809	0.4812	0.4947	0.4191	0.4336	<u>0.4754</u>	0.5082	0.4944	0.5383
	MUSIQ↑	55.3260	57.0371	<u>59.6610</u>	60.2753	59.9715	60.7579	49.2505	57.6671	<u>71.9982</u>	63.6003	61.2496	74.3325
iPhone	NIQE↓	4.9410	4.8155	<u>4.0555</u>	4.7661	4.2139	4.0934	5.0436	<u>3.5932</u>	3.7053	3.5792	3.5883	3.5087
	PI↓	5.0171	<u>3.9994</u>	4.1304	4.5345	3.8462	4.1123	5.0410	<u>3.4293</u>	3.5367	3.3379	3.3233	3.2661
	CLIP-IQA↑	0.4603	0.5352	<u>0.5387</u>	0.5367	0.5462	0.5503	0.3723	<u>0.5931</u>	0.5930	0.6064	0.6120	0.6211
	MUSIQ↑	55.5380	58.1422	<u>58.6204</u>	57.6496	59.1077	60.1002	57.8120	71.9805	<u>72.1243</u>	73.5609	72.9853	73.6150
OPPO	NIQE↓	5.0104	4.5568	<u>4.5166</u>	4.9704	4.5293	4.5071	5.7811	<u>4.9042</u>	5.3017	4.7389	4.7885	4.6699
	PI↓	5.1828	4.9569	<u>4.9022</u>	5.0729	4.9006	4.8752	5.8036	<u>4.9676</u>	5.3808	5.0513	4.9911	4.5702
	CLIP-IQA↑	0.5820	<u>0.5622</u>	0.5590	0.5071	0.5502	0.5591	0.3499	<u>0.4798</u>	0.4518	0.4673	0.5461	0.5595
	MUSIQ↑	67.0395	67.4327	<u>67.5921</u>	67.1736	67.5282	67.6444	54.9076	<u>73.0842</u>	72.1563	72.8543	74.7868	75.2187

A.7 Benchmarking ZI dataset across 1D, 2D, and 3D-TPR frameworks

The 1D-indexed models are directly adopted from publicly released versions of each corresponding FI network. The 2D-indexed counterparts follow the interpolation strategy introduced in [55]. For each framework, we finetune the models using our proposed ZI dataset and evaluate their performance on the real-world multi-device test sets. Both qualitative and quantitative comparisons are conducted to assess the improvements enabled by ZI supervision under different indexing paradigms.

Quantitative comparisons. We conduct a comprehensive evaluation of the 1D, 2D, and 3D-TPR indexing frameworks across four FI models and four real-world smartphone platforms, considering both the base models and those finetuned on our ZI dataset. As shown in Table 6, the 3D-TPR framework consistently outperforms its 1D and 2D counterparts after finetuning, achieving the best performance (highlighted in bold) across nearly all perceptual quality metrics. Notably, even without finetuning, 3D-TPR frequently ranks second (underlined), demonstrating strong generalization capability and architectural robustness. In contrast, 1D and 2D variants exhibit only limited performance gains,

716 particularly under challenging cross-device settings. These results support two conclusions: (1) while
 717 our ZI dataset provides performance gains across all indexing paradigms, the improvements are most
 718 pronounced under the trajectory-aware 3D-TPR framework; and (2) OmniZoom (i.e., 3D_f-TPR)
 719 offers a universal, plug-and-play solution for real-world cross-device zoom interpolation, effectively
 720 bridging domain gaps and enhancing perceptual quality across diverse hardware platforms.

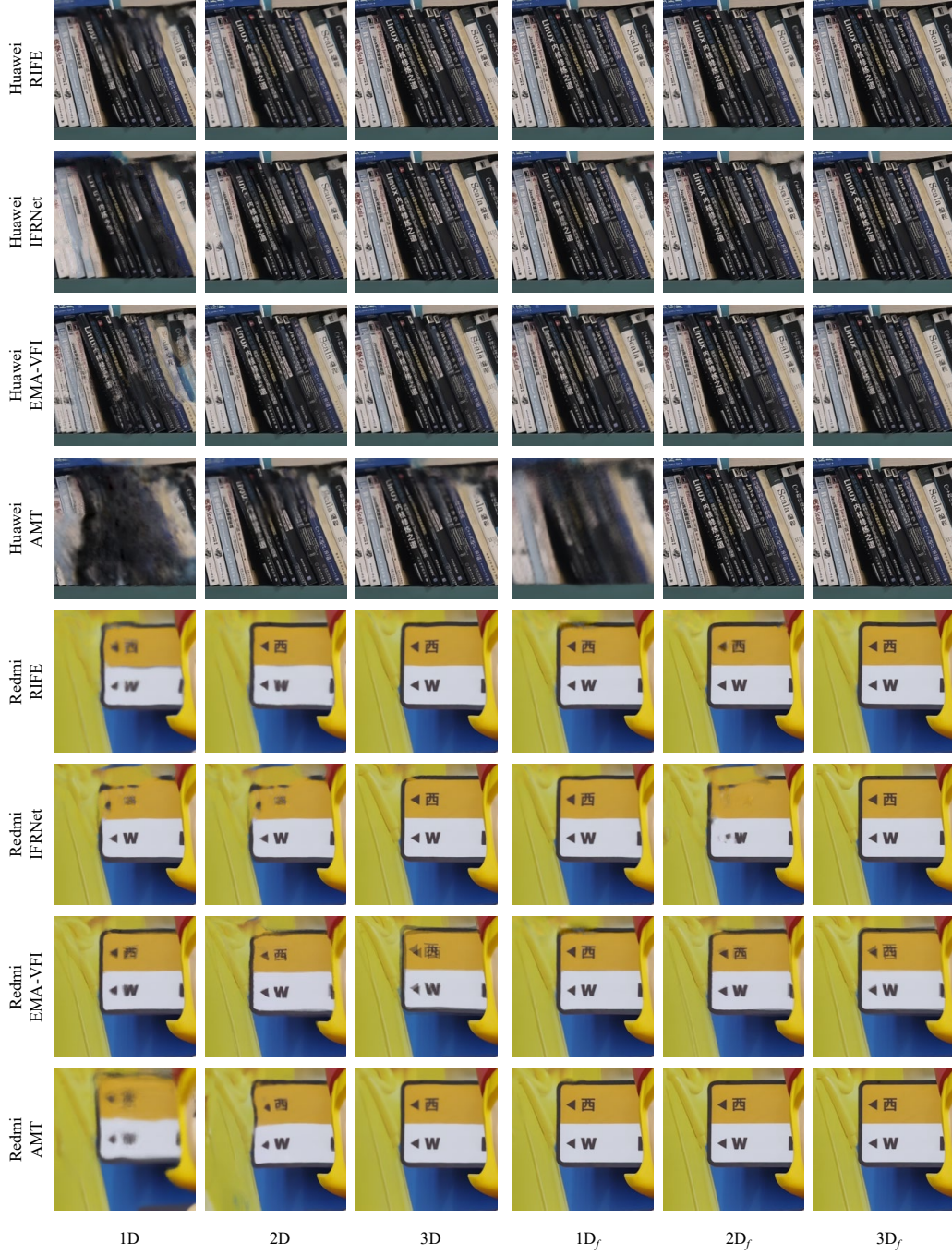


Figure 12: Qualitative comparisons on Huawei (top four rows) and Redmi (bottom four rows). While 1D and 2D models exhibit noticeable blurring and detail loss across networks, 3D_f effectively restores fine details, producing sharper structures and improved local contrast.

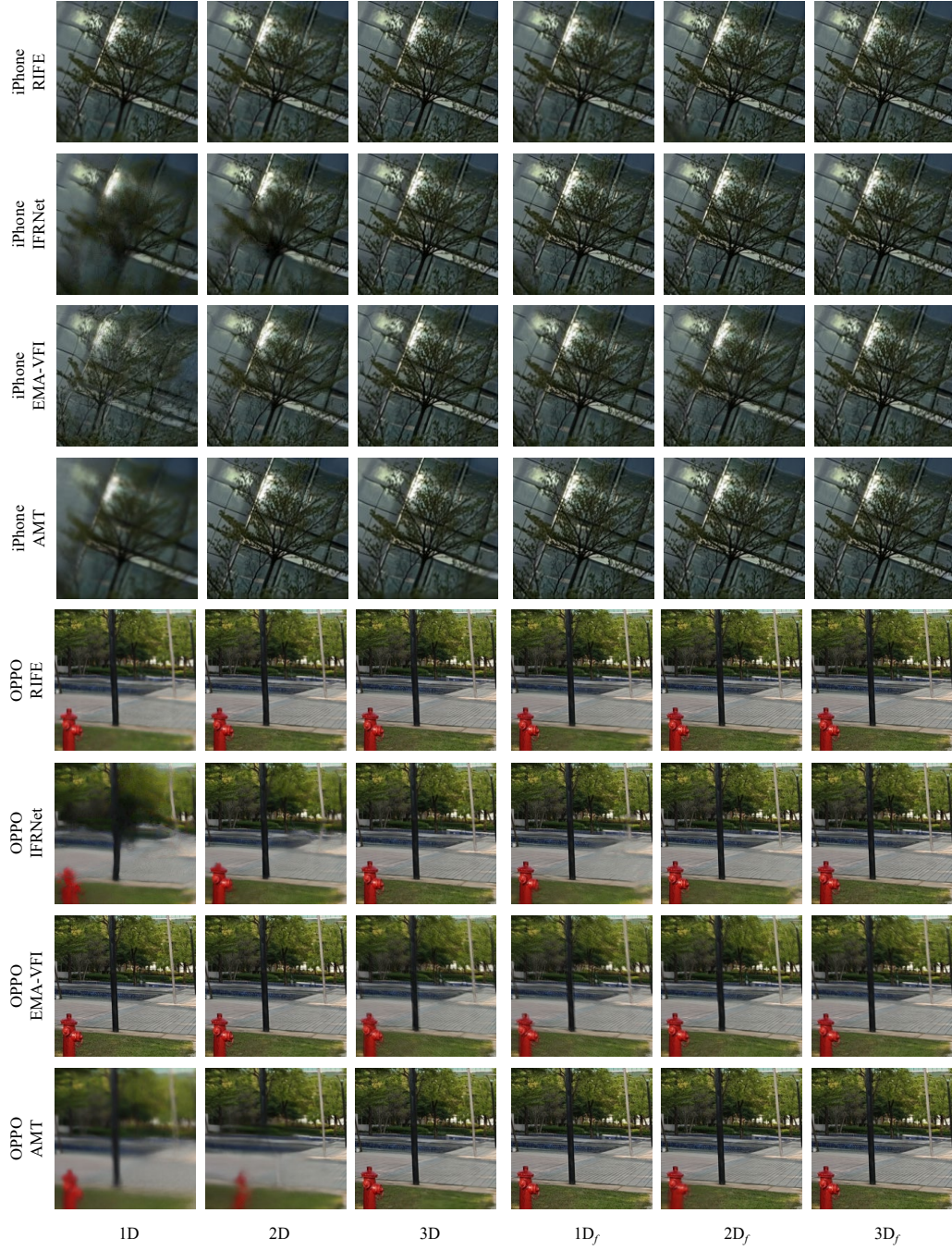


Figure 13: Qualitative comparisons on iPhone and OPPO test sets under rich structural conditions. While 1D and 2D variants exhibit edge instability and perceptual artifacts, $3D_f$ produces noticeably clearer results with sharper contours and reduced blurring, effectively preserving structural fidelity.

Qualitative comparisons. Beyond quantitative evaluation, we provide qualitative results in Figures 12 and 13, illustrating ZI outputs on Huawei, Redmi, iPhone, and OPPO devices across four FI networks. Each figure presents six variants per network: 1D, 2D, and 3D, along with their finetuned counterparts ($1D_f$, $2D_f$, and $3D_f$).

Several key observations emerge from these visualizations. First, the 3D configuration consistently produces sharper and more coherent results than its 1D and 2D counterparts, highlighting the

effectiveness of the 3D-TPR framework in real-world ZI tasks. Second, across all indexing paradigms, all finetuned models show clear perceptual improvements, demonstrating the broad applicability of our ZI dataset across diverse FI architectures.

Notably, $3D_f$ achieves the highest perceptual quality in nearly all cases, with fewer artifacts and enhanced texture fidelity. For example, on Redmi sequences, $3D_f$ recovers sharper character contours and fine line structures compared to other variants. On iPhone and OPPO, where scenes contain more intricate textures and geometric structures, 1D and 2D variants frequently introduce ghosting and edge artifacts, issues that are substantially mitigated by the $3D_f$ configuration.

These results underscore the strength of OmniZoom as a unified, cross-device solution for plug-and-play zoom interpolation, offering robust perceptual quality across heterogeneous hardware platforms.

A.8 Upper bound analysis of 3D-TPR framework

To evaluate the performance ceiling of the 3D-TPR framework, we adopt ground-truth 3D geometry as the timestep map, thereby ensuring perfect consistency between training and inference. As shown in Table 7, we conduct a complementary ablation study under this setting for fair comparison.

Results indicate that utilizing consistent 3D similarity maps during both training and testing leads to substantial gains across all metrics and networks: PSNR improves by over 1.0 dB on average, SSIM consistently increases, and LPIPS is significantly reduced. These results reveal the latent capacity of the 3D-TPR framework when accurate geometry is available throughout.

This upper bound analysis provides valuable insight into the theoretical capacity of our method. While full 3D supervision may not be available in practice, our empirical findings suggest that leveraging lightweight geometry estimation modules or substituting with physically obtainable priors (e.g., depth from stereo, SLAM-based pose, or monocular depth) can approximate similar benefits. This observation offers practical guidance for future extensions that aim to balance reconstruction quality and computational feasibility.

Table 7: Upper bound evaluation of the proposed 3D-TPR framework on four FI networks using ground-truth 3D supervision. The 2D framework serves as a baseline [55]. All experiments on 3D use the 3D trajectory progress ratio as timestep map. 3D-t-m indicates using only 3D-TPR encoding; 3D-m adds texture-focus strategy; 3D-t includes mask penalty constraint; and 3D integrates all components. **Bold** indicates improvements over 2D baseline.

Network	RIFE			IFRNet			EMA-VFI			AMT		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
2D	27.4050	0.9010	0.0863	27.1276	0.8994	0.0780	24.7313	0.8514	0.0810	27.1726	0.9017	0.0807
3D-t-m	28.7510	0.9267	0.0773	28.2484	0.9225	0.0715	25.4078	0.8618	0.0780	28.4850	0.9269	0.0837
3D-m	28.7311	0.9264	0.0764	28.3486	0.9241	0.0690	25.4705	0.8631	0.0800	28.5942	0.9285	0.0767
3D-t	28.5518	0.9234	0.0764	28.1917	0.9206	0.0686	25.3557	0.8618	0.0781	28.5895	0.9277	0.0810
3D	28.6548	0.9255	0.0773	28.2967	0.9220	0.0656	25.4984	0.8633	0.0770	28.6271	0.9286	0.0768

A.9 Additional visual results of 3D-TPR framework

We present additional qualitative comparisons between the proposed 3D-TPR framework and conventional 2D baselines, as shown in Figure 14. The results demonstrate that 3D-TPR effectively restores degraded regions in 2D-based outputs, particularly in areas suffering from motion-induced blur where structural details are severely corrupted or entirely missing.

Moreover, 3D-TPR alleviates misalignment artifacts commonly encountered in 2D models when handling textures with high levels of repetition and self-similarity, such as linear structures or dot-like patterns, by incorporating trajectory-aware spatial priors. In addition to these localized corrections, our method consistently produces globally improved visual quality, exhibiting sharper details, better structural coherence, and a more realistic overall appearance. These findings further highlight the advantages of integrating 3D spatial awareness into temporal interpolation tasks.

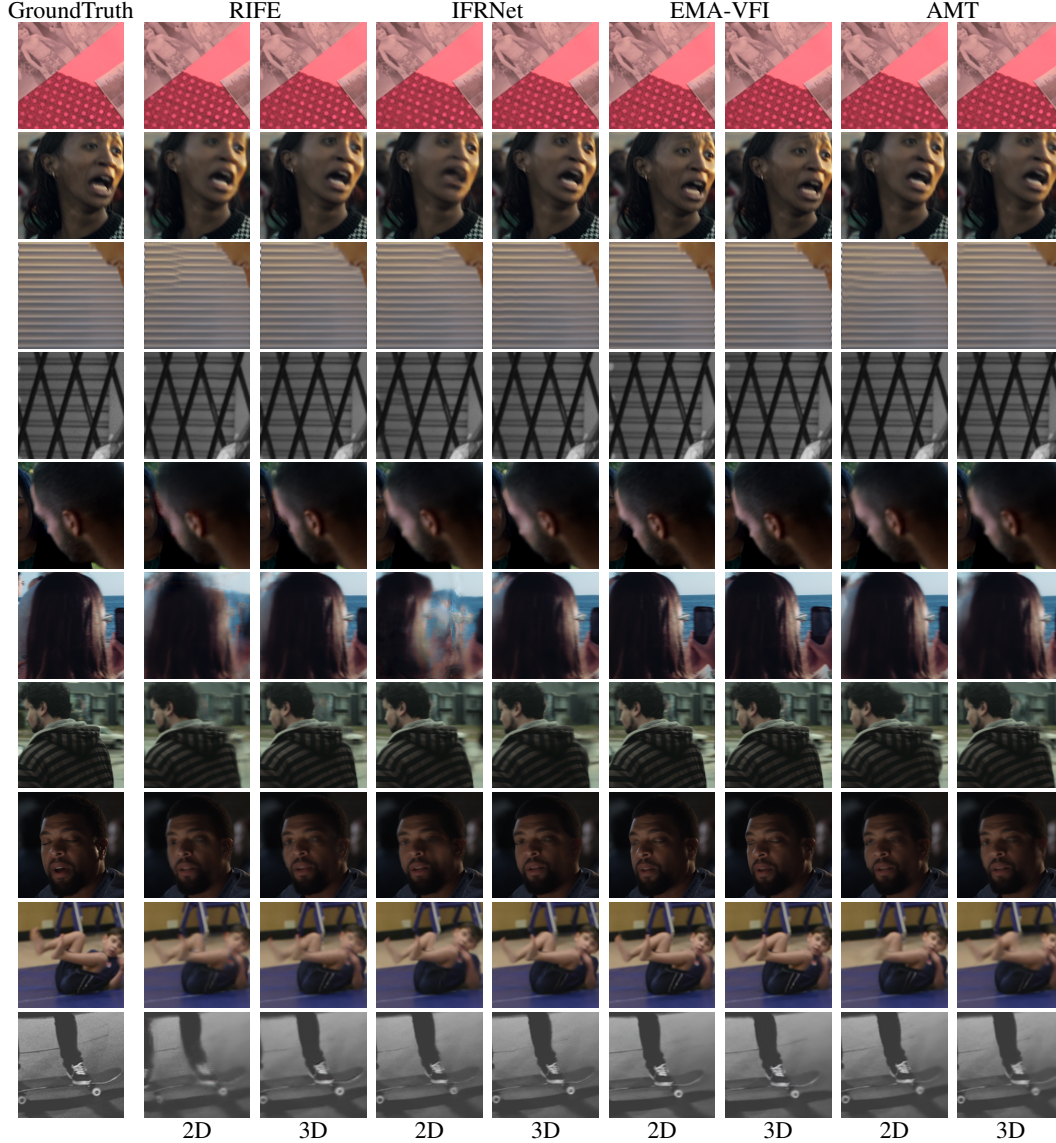


Figure 14: Visual comparison of 2D and 3D interpolation results across networks at the same timestep. The 3D method consistently yields less-blurred and neater outputs across all networks.

763 A.10 Limitations

764 While OmniZoom is designed to be plug-and-play across a variety of FI networks, achieving optimal
 765 performance may still require minor adjustments to the training schedule, such as tuning the learning
 766 rate warm-up strategy, rebalancing loss weights, or modifying batch size. These adjustments, although
 767 lightweight, reflect the need for architecture-aware training when applying our framework to diverse
 768 settings.

769 A.11 Broader impacts

770 This work is motivated by the goal of enhancing frame interpolation in zoom scenarios, with direct
 771 benefits for a wide range of applications in mobile photography, video conferencing, augmented
 772 reality (AR), and digital content creation. Our proposed framework improves the temporal smoothness
 773 and spatial consistency of interpolated video frames across heterogeneous smartphone platforms,
 774 contributing to better visual quality in real-world capture workflows. In particular, it enables smoother

775 zoom transitions without requiring any hardware-level changes due to zero cost, offering cost-effective
776 enhancements for existing consumer devices.

777 Such capabilities are especially impactful in mobile videography, where users often experience
778 abrupt visual discontinuities during zoom. Our method can help reduce motion artifacts and maintain
779 semantic coherence in challenging scenes, which is beneficial for video-based communication,
780 content sharing on social platforms, and immersive AR experiences that rely on high frame fidelity.
781 Furthermore, cross-device compatibility and plug-and-play integration facilitate broader adoption
782 across hardware vendors, making advanced interpolation accessible without retraining for each model.

783 Nevertheless, as with other generative video technologies, improvements in visual realism may pose
784 risks if misused, such as enabling the creation of misleading or manipulated video content. While
785 our framework is restricted to supervised frame interpolation under dual-camera settings and does
786 not support arbitrary video generation, we emphasize the importance of responsible use. To mitigate
787 misuse, we do not release general-purpose generative models, and our dataset is limited to constrained
788 zoom interpolation scenarios. We encourage future efforts to pair such technologies with appropriate
789 safeguards and transparency mechanisms to ensure ethical deployment.