SMART-GS: BLUR-GUIDED GAUSSIAN SPLATTING RECONSTRUCTION

Anonymous authors
Paper under double-blind review

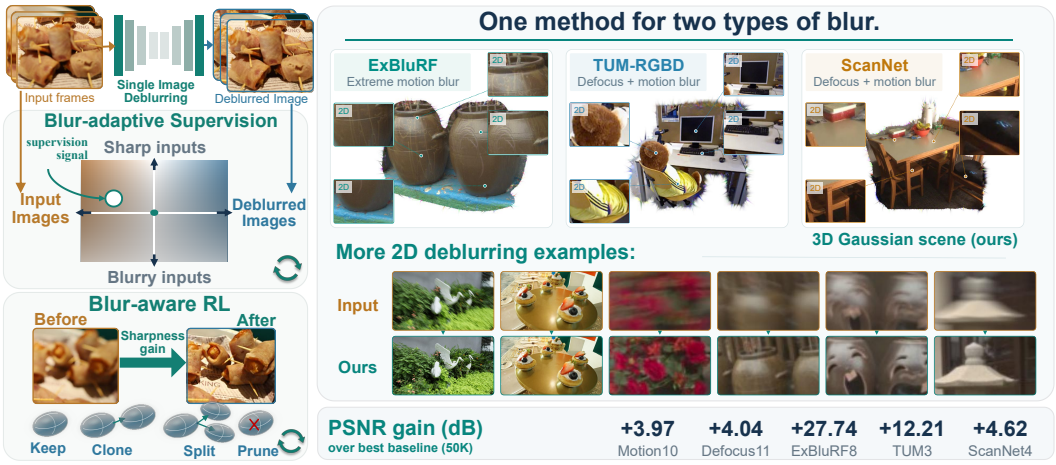

Figure 1: **Real-world Gaussian deblurring across motion and defocus blur.** Our method reconstructs a sharp 3D Gaussian splatting (3DGS) reconstruction from multiple blurred input images. Since standard 3DGS does not work well with blurry inputs, we use single-image deblurring as a prior to guide a blur-adaptive supervision of the 3DGS reconstruction, as well as to steer reinforcement learning (RL) refinement of the 3DGS scene representation, yielding superior performance on multiple defocus and motion-blur datasets.

ABSTRACT

A blurred image does not tell 3D Gaussian Splatting how to learn the sharp details, while a deblurring network could not add enough details that are consistent across views. Both problems are strongly related and a joint optimization by no means straightforward. We propose SMART-GS, an offline deblurring 3DGS method in which image blur guides the optimization process without assuming a particular type of blur. We use the amount of blur in the input image to define how much each observed frame and a corresponding deblurred image should supervise the optimization method. The blurriness of the current rendered image then measures how many sharp details have been recovered which we use to update the supervision loss for recovering more details in the blurred image. We also use the blurriness of both images to guide 3DGS densification so the optimizer learns both what to trust and where to grow in a complex input video sequence. One set of sharp 3D Gaussian representations explains captured and deblurring observations through ordered, learned blur formation. At inference, only this sharp representation remains. SMART-GS reaches state-of-the-art deblurring performance on five real-world benchmarks: motion blur, defocus blur, extreme motion blur, TUM-RGBD, and the ScanNet dataset. We achieve 27.74 dB PSNR higher than the state-of-the-art methods on the extreme motion blur sequences. We also study how supervision and capacity affect the optimization of large indoor scenes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

A useful 3D representation should stay sharp and consistent when the camera moves. This matters in virtual visits (Dutt et al., 2026), heritage inspection (Llull et al., 2023), and interactive 3D content (Panek et al., 2023; Jiang et al., 2024), where a viewer explores the scene from prior 3D map (Kerbl et al., 2023; Jamil & Brennan, 2025). The same need appears in robotics, where blur can hurt dense SLAM, navigation, and language- or manipulation-driven 3D queries in the prior map (Murai et al., 2025; Zhang et al., 2026; Chen et al., 2025; Kerr et al., 2023; Zheng et al., 2024; Qin et al., 2024). 3D Gaussian Splatting (Kerbl et al., 2023) makes these uses practical, but capture is often imperfect. Camera motion and finite depth of field can erase thin structures, boundaries, and small textures and add more 3D inconsistent noise in the views. We study how to use 2D and 3D clues to get sharp and consistent 3D reconstruction with various input sequences.

The blurriness of a frame creates three linked decisions: which image in the input video sequence should be assigned more weight during training? How to balance the learning of the deblurred and input image? Where should the method spend more Gaussian capacity? A standard 3D Gaussian Splatting (Kerbl et al., 2023) system assumes the input image must be sharp. A single image deblurring (Kong et al., 2025; Zamir et al., 2022) network can propose a relatively sharper target, but cannot recover all the details. If the optimization method makes some details that is not consist in 3D, the error would be introduced to 3D geometry. The single image may look sharp while the recovered scene is inconsistent in 3D.

We use one simple rule: let blurriness guide the three decisions. First, the blurriness of input images sets the trust in the sharp, reliable images and blurry images. This signal also sets the initial reliance on captured image and a deblurred prior. For unreliable images, we first render one sharp image from the Gaussian representation, then use the blur kernel to reblur the sharp image. We compare both to optimize the blur kernel and get the sharp details. Second, we inspect the current render. If a recovered edge is supported by the input, the optimization can rely on it more. If the edge is only a prior artifact, the loss limits its influence. This recovery signal changes the supervision during training. Finally, we also use this signal as a reward function for Reinforcement Learning to allocate and prune Gaussians. A delayed probe checks whether the action improved recovery on permitted training views. This delay connects an action to its later image effect. The image supervision changes continuous Gaussian parameters. The policy changes the capacity and 3D spatial resolution. Our central hypothesis is therefore direct: **image blur should guide what Gaussian Splatting learn and where they grow to get a 3D sharp reconstruction**. The hypothesis has two testable parts. Rendered recovery should improve the balance between captured and deblurred image supervision. The same recovery evidence should make capacity changes more useful than preset densification and pruning heuristics. To summarize, we make the following **contributions**:

(1) A blur-guided Deblurring Gaussian Splatting method. We connect captured evidence and deblurring prior, rendered recovery, and Gaussian capacity in one pipeline for mixed real captures without any assumption on blur labels. The system achieves state-of-the-art deblurring performance across 5 real-image deblurring datasets.

(2) Adaptive supervision optimization. The level of blurriness of the input image affects the optimization weights of different image in the video sequence. It also initialises the trust between captured and deblurred observations. Rendered recovery then updates their trusts and the supported-detail loss during the learning process.

(3) Recovery-guided capacity optimization. A blur-conditioned controller combines local reconstruction gradients, scene-level blurriness, and delayed probe quality to choose keep, clone, split, and prune actions.

2 RELATED WORK

Image and video deblurring priors. Single-image deblurring methods predict a deblurred image from one blurred image (Kong et al., 2025). By design, these methods do not output camera poses or shared scene geometry. Therefore, for multiple views they cannot by itself guarantee cross-view consistency or synthesize a geometrically consistent novel view. Video deblurring methods use temporal history across neighboring frames (Ghasemabadi et al., 2024; Rao et al., 2024); temporal evidence is still not a 3D representation. These methods provide useful proposals.

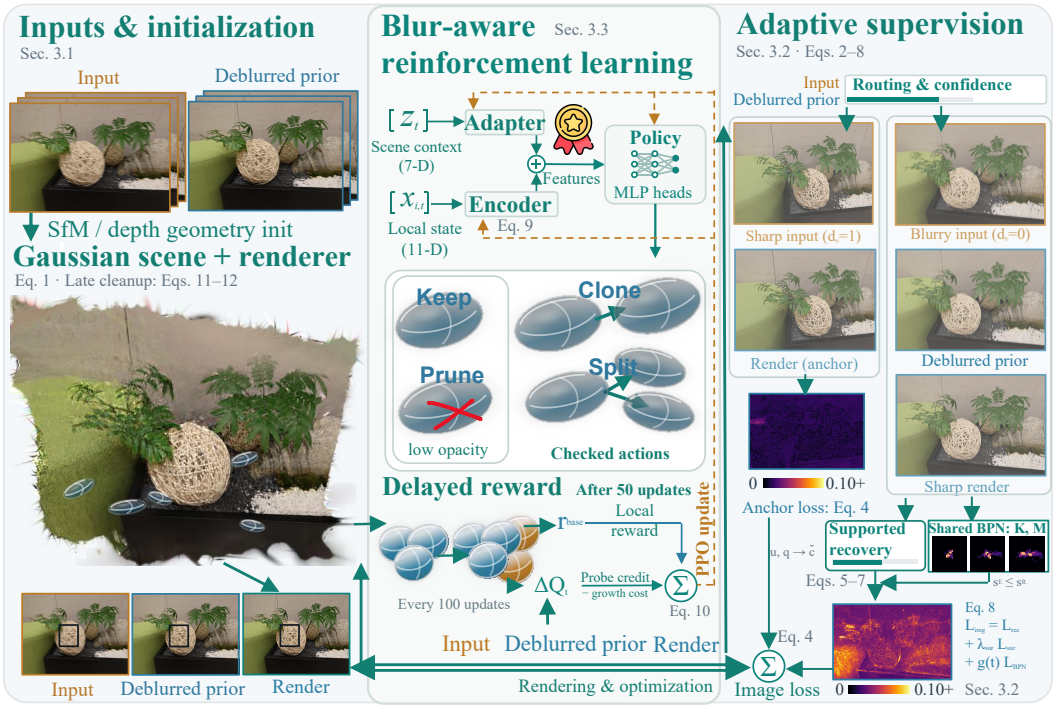

Figure 2: **Overview of SMART-GS.** Our blur-aware 3DGS reconstruction pipeline has two major buidling blocks: **Adaptive supervision** (Sec. 3.2): sharp frames supervise the render directly; for blurry frames, a shared blur proposal network (BPN) reblurs the render to explain both the input and its deblurred image, and the detail the render has recovered decides how much each of the two supervises the scene. **Blur-aware reinforcement learning** (Sec. 3.3): the same recovery signal gives delayed rewards to a policy that chooses to keep, clone, split, or prune, and checks on training views to decide which structural proposals are executed.

3D reconstruction with explicit blur models. Deblur-NeRF and DP-NeRF learn a sharp radiance field through an explicit image-formation model (Ma et al., 2022; Lee et al., 2023a). Gaussian methods extend this idea to motion and defocus. Motion-aware methods model camera motion with exposure samples or trajectories (Lee et al., 2025b; Matta et al., 2026). Defocus-aware methods use circle-of-confusion or aperture models (Lee et al., 2025a; Wang et al., 2025).

Label-free 3D deblurring. BAGS models mixed blur with learned multi-scale kernels and a quality mask (Peng et al., 2024). Unblur-SLAM couples feed-forward deblurring with multi-view SLAM and 3D blur modeling (Zhang et al., 2026).

3 SMART-GS

SMART-GS fits a single Gaussian scene to sharp frames and to pairs of blurry input and deblurred images (Figure 2). Sharp frames supervise the render directly. For a blurry frame, the render is reblurred by learned blur kernels, following BAGS (Peng et al., 2024), and compared with both the input and its deblurred version; how much detail the render has already recovered decides how supervision is split between the two images. The same recovery signal rewards a reinforcement-learning policy that chooses keep, clone, split, or prune for each Gaussian. Adaptive supervision thus optimizes the Gaussian parameters, and the policy controls their number and placement. Sec. 3.1 defines the inputs, Sec. 3.2 the supervision, and Sec. 3.3 the policy.

3.1 Inputs and Initialization

The inputs are calibrated images of a static scene; blur-type labels are not required. For training view v with camera C_v, I_v is the captured input image and I'_v is its deblurred version from a single-image

deblurring network (Kong et al., 2025), which we call the *deblurred prior* (or simply the prior). A binary label $d_v \in \{0, 1\}$ marks sharp views. Sharp views are *direct anchors* ($d_v = 1$) and are supervised by I_v alone; blurry views ($d_v = 0$) use both I_v and I'_v. Sharp labels are those from views with high scores by NIMA (Talebi & Milanfar, 2018), a no-reference image-quality predictor. Sharp views are sampled more often during training. Deblurred images and labels are precomputed and fixed.

We initialize the scene from SfM points and, where available, depth images. All views share a 3D Gaussian representation Θ_t at optimization step t. With $\mathcal{R}_{\mathrm{RGB}}$ the differentiable Gaussian rasterizer, the sharp render at camera C_v is

$$I^{\mathrm{3D}}_{v,t} = \mathcal{R}_{\mathrm{RGB}}(\Theta_t, C_v). \tag{1}$$

For each blurry view, an initial prior confidence $c_v \in [0, 1]$ combines four checks of I'_v against I_v: color fidelity (low-frequency colors agree), edge agreement (edge directions agree), clipping safety (few newly saturated pixels), and Laplacian gain (high-frequency detail added by I'_v). Anchors have $c_v = 1$. Sec. 3.2 revises this confidence during training. Details are in Appendix B.1. A small fixed set of training views, the *probe views*, is used to measure the effect of structural changes.

3.2 Adaptive supervision

Shared BPN. The deblurred image may retain residual blur, so we model both I'_v and I_v as blurred versions of the same sharp render. A blur proposal network (BPN) maps a learned per-view embedding to a nonnegative normalized kernel $K_{v,t}$ and two blur strengths. Two branches share $K_{v,t}$: the prior branch (superscript E, for enhanced) explains I'_v, and the input branch (superscript R, for real capture) explains I_v. With δ the identity kernel,

$$\begin{aligned} K^E_{v,t} &= \delta + s^E_{v,t}(K_{v,t} - \delta), \\ K^R_{v,t} &= \delta + s^R_{v,t}(K_{v,t} - \delta), \qquad 0 \le s^E_{v,t} \le s^R_{v,t} \le 1. \end{aligned} \tag{2}$$

Strength 0 leaves the image unchanged and strength 1 applies the full $K_{v,t}$; the ordering lets the prior retain some blur while the input is blurred at least as much. A shared mask $M_{v,t} \in [0, 1]$, predicted from I_v, I'_v, and their difference, sets where blur applies. With $*$ denoting 2D convolution and $\odot$ element-wise multiplication, the *formed images* reblur the sharp render $I^{\mathrm{3D}}_{v,t}$ of Eq. (1) inside the mask:

$$\widehat{I}^b_{v,t} = M_{v,t} \odot (K^b_{v,t} * I^{\mathrm{3D}}_{v,t}) + (1 - M_{v,t}) \odot I^{\mathrm{3D}}_{v,t}, \quad b \in \{E, R\}, \tag{3}$$

where the hat marks an image synthesized from the render: $\widehat{I}^E_{v,t}$ and $\widehat{I}^R_{v,t}$ are the model's predictions of I'_v and I_v.

All image comparisons use $\ell(X, Y) = 0.8\,\ell_1(X, Y) + 0.2\,\big(1 - \mathrm{SSIM}(X, Y)\big)$, where ℓ_1 is the mean absolute pixel difference and SSIM the structural similarity index (Wang et al., 2004). Anchors compare the unblurred render with I_v. Blurry views compare $\widehat{I}^E_{v,t}$ with I'_v and $\widehat{I}^R_{v,t}$ with I_v, mixed by the input-branch weight $w^R_{v,t} \in [0, 1]$:

$$\begin{aligned} w^R_{v,t} &= (1 - d_v)(1 - \widetilde{c}_{v,t})g(t), \\ \mathcal{L}_{\mathrm{rec},v,t} &= \begin{cases} \ell(I^{\mathrm{3D}}_{v,t}, I_v), & d_v = 1, \\ (1 - w^R_{v,t})\,\ell(\widehat{I}^E_{v,t}, I'_v) + w^R_{v,t}\,\ell(\widehat{I}^R_{v,t}, I_v), & d_v = 0. \end{cases} \end{aligned} \tag{4}$$

Here $\widetilde{c}_{v,t}$ is the effective prior confidence, which starts at c_v and is updated by Eq. (6), and the schedule $g(t) \in [0, 1]$ rises smoothly from 0 to 1 between 5% and 25% of the run. Input supervision thus grows as prior confidence falls. Appendix B.2 gives the BPN details.

Supported recovery. To revise c_v during training, we compare the detail in the render with that in the prior, measured only on a support region. The support keeps every Laplacian edge response of I_v and admits a response of I'_v only up to the strongest response of I_v within a 5×5 neighborhood, so prior edges with no nearby input edge receive no support. Let $\mathcal{E}_v(X)$ be the log mean squared Laplacian response of image X on this support, combined over three scales with weights 1, 0.5, and 0.25. The render's surplus over the prior is

$$u_{v,t} = \mathcal{E}_v(I^{\mathrm{3D}}_{v,t}) - \mathcal{E}_v(I'_v), \tag{5}$$

which is positive when the render contains more supported detail than the prior. Because a single value may be transient, each view keeps exponential moving averages of $u_{v,t}$ and $u_{v,t}^2$; the average minus one standard deviation, discounted for views with few updates, gives a local signal $h_{v,t} \in [0,1]$ that approaches one when view v has stably exceeded the prior. A scene-level signal $\gamma_t \in [0,1]$ averages this stable positive surplus over non-anchor views that have received enough updates. The confidence is then revised as

$$\begin{aligned} q_{v,t} &= (1-d_v)h_{v,t}\gamma_t, \\ \widetilde{c}_{v,t} &= d_v + (1-d_v)c_v(1-q_{v,t}). \end{aligned} \tag{6}$$

The recovery evidence $q_{v,t} \in [0,1]$ is the fraction of c_v removed because the render has recovered beyond the prior. Thus $\widetilde{c}_{v,t} = 1$ for anchors, and for blurry views it falls from c_v toward zero as $q_{v,t}$ grows, which raises $w_{v,t}^R$ and shifts supervision toward I_v. Both signals are detached from the rendering graph.

Laplacian energy cannot tell a correctly placed edge from a displaced or inverted one, so we also compare signed responses pixel by pixel. All three terms use one target: the prior's response magnitude, scaled by a factor $\Lambda_{v,t} \geq 1$ that grows as prior confidence falls or recovery rises. $\mathcal{L}_{\text{floor},v,t}$ penalizes render responses (taken along the prior's sign) below this target, $\mathcal{L}_{\text{over},v,t}$ softly penalizes responses above it inside the support, and $\mathcal{L}_{\text{art},v,t}$ penalizes responses outside the support, i.e., likely artifacts:

$$\mathcal{L}_{\text{sur},v,t} = (1-d_v)\left(\widetilde{c}_{v,t}\mathcal{L}_{\text{floor},v,t} + \widetilde{c}_{v,t}^2\mathcal{L}_{\text{over},v,t} + \mathcal{L}_{\text{art},v,t}\right). \tag{7}$$

As $\widetilde{c}_{v,t}$ falls, the first two terms relax while the artifact term stays active. Appendix B.3 defines the moving statistics and the terms.

Total loss. The image objective is

$$\mathcal{L}_{\text{img}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{sur}}\mathcal{L}_{\text{sur}} + g(t)\mathcal{L}_{\text{BPN}}, \tag{8}$$

where $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{sur}}$ average Eqs. (4) and (7) over sampled views, $\lambda_{\text{sur}} = 0.1$, and $g(t)$ is the schedule of Eq. (4), so the BPN regularizer is phased in together with the input branch. The BPN regularizer $\mathcal{L}_{\text{BPN}} = \lambda_{\text{ctr}}\mathcal{L}_{\text{ctr}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{sm}}\mathcal{L}_{\text{sm}}$ keeps the mass of $K_{v,t}$ near its center so that the kernel models blur rather than a shift ($\mathcal{L}_{\text{ctr}}$), pulls $M_{v,t}$ toward the normalized $|I'_v - I_v|$ so that blur is placed where the two images disagree ($\mathcal{L}_{\text{mask}}$), and penalizes spatial variation of $M_{v,t}$ ($\mathcal{L}_{\text{sm}}$); forms and weights are in Appendix B.2.

3.3 Blur-aware reinforcement learning

A large residual may come from missing scene detail or from an unreliable target. We extend LeGS (Ning et al., 2026) with blur and recovery information in its state and reward to separate these cases. The policy has two heads: a growth head chooses to keep, clone, or split, and a low-opacity head, applied only to Gaussians below an opacity threshold, that chooses to keep or prune.

State. Each Gaussian i has different attributions (Kerbl et al., 2023). Its local features are the ten gradients of $\mathcal{L}_{\text{img}}$ with respect to these parameters and a blur-aware sensitivity score $s_{i,t}^A$ (the superscript distinguishes it from the blur strengths s^E, s^R), which estimates how strongly Gaussian i affects the renderings. It combines the LeGS sensitivity with a bounded, confidence-gated first-order estimate of the Gaussian's influence on Eq. (8). Stacking these eleven values for all N_t Gaussians gives $X_t \in \mathbb{R}^{N_t \times 11}$, and the local state is

$$\boldsymbol{x}_{i,t} = \left[\mathcal{Z}(X_t)_{i,:}\right]^\top \in \mathbb{R}^{11}, \tag{9}$$

where $\mathcal{Z}(\cdot)$ standardizes each column to zero mean and unit variance over the Gaussians and $(\cdot)_{i,:}$ takes row i; $\boldsymbol{x}_{i,t}$ thus compares Gaussian i with the current population.

Seven scene-level features, shared by all Gaussians, describe prior confidence, supported recovery, learned blur, and population pressure (the number of Gaussians relative to a reference count), giving an 18-dimensional input. A bias-free linear adapter, initialized to zero, adds these features to the encoded local state, so the policy starts identical to LeGS. Its output is scaled by $\alpha_t \in [0,1]$, which rises linearly from 0 at update 2K to 1 at 5K. Appendix B.4 defines the features.

Table 1: **Five-suite comparison at original and aligned (*) budgets.** All supplied source frames, including the marked evaluation frames, enter training. Metrics use those marked source frames; native preprocessing and pose estimation are retained where required. Each cell reports the terminal (final) checkpoint PSNR/SSIM/LPIPS triplet. Higher PSNR/SSIM and lower LPIPS are better.

Method	Motion10	Defocus11	ExBluRF8	TUM3	ScanNet4
Original training budgets					
PRISM3D	27.57 / 0.89 / 0.13	25.17 / 0.90 / 0.12	27.46 / 0.75 / 0.34	25.21 / 0.87 / 0.11	23.32 / 0.76 / 0.41
BAGS	32.36 / 0.95 / 0.05	26.48 / 0.90 / 0.10	20.68 / 0.54 / 0.50	18.94 / 0.58 / 0.21	20.17 / 0.65 / 0.33
Unblur-SLAM	27.52 / 0.89 / 0.10	26.40 / 0.89 / 0.10	24.41 / 0.73 / 0.34	26.12 / 0.86 / 0.09	19.07 / 0.52 / 0.30
DOF-GS	29.48 / 0.91 / 0.11	28.00 / 0.93 / 0.08	19.68 / 0.60 / 0.53	21.24 / 0.79 / 0.27	19.70 / 0.69 / 0.54
Aligned budgets: 50K updates; 25×25 image-space kernels where applicable					
PRISM3D*	33.28 / 0.97 / 0.03	29.73 / 0.94 / 0.06	29.35 / 0.79 / 0.21	28.24 / 0.91 / 0.07	27.10 / 0.82 / 0.30
BAGS*	32.58 / 0.96 / 0.05	26.39 / 0.89 / 0.10	20.23 / 0.52 / 0.47	19.64 / 0.62 / 0.19	20.91 / 0.64 / 0.30
Unblur-SLAM*	27.70 / 0.89 / 0.10	26.40 / 0.89 / 0.11	24.27 / 0.74 / 0.33	26.05 / 0.85 / 0.09	19.14 / 0.53 / 0.29
DOF-GS*	29.78 / 0.91 / 0.10	28.10 / 0.93 / 0.08	19.99 / 0.61 / 0.53	21.16 / 0.79 / 0.27	19.72 / 0.69 / 0.54
SMART-GS	**37.25 / 0.98 / 0.03**	**33.77 / 0.97 / 0.04**	**57.09 / 1.00 / 0.001**	**40.45 / 0.95 / 0.02**	**31.72 / 0.91 / 0.09**

Delayed reward. Every 100 updates, a *structural event* samples an action $a_{i,t} \in \{\text{keep}, \text{clone}, \text{split}, \text{prune}\}$ for each eligible Gaussian i, where t is the step of the event. Because a new Gaussian needs time to adapt, we evaluate the probe views before the event and again $d_{\text{RL}} = 50$ updates later. We compare three quantities: *target agreement*, the PSNR of the unblurred render against its target (I_v for anchors, I'_v otherwise); *input-formation agreement*, the PSNR of $\widehat{I}^R_{v,t}$ against I_v; and *supported surplus*, a confidence-weighted average of $\tanh(u_{v,t})$ over blurry probe views. Each change is normalized by its running scale, and the three are combined with weights set by the mean prior confidence and the mask coverage (Appendix B.5) into a scene-level signal ΔQ_t, positive when the probes improve. The reward $r_{i,t}$ for action $a_{i,t}$ is

$$r_{i,t} = r^{\text{base}}_{i,t} + \alpha_{t+d_{\text{RL}}} w_q \mathbb{I}_{a_{i,t}\neq\text{keep}} \Delta Q_t b_{i,t} - \alpha_{t+d_{\text{RL}}} w_c \mathbb{I}_{a_{i,t}\in\{\text{clone,split}\}} \kappa_{i,t}, \tag{10}$$

where $\mathbb{I}_{[\cdot]}$ is the indicator function. The base reward $r^{\text{base}}_{i,t}$, inherited from LeGS, is the delayed change in s^A of Gaussian i and its descendants (the Gaussians created from it by clone or split). The second term credits a non-keep action with ΔQ_t, distributed by a gate $b_{i,t} \in [0, 2]$ that favors Gaussians whose own base reward agrees in sign with ΔQ_t. The third term charges clone and split a growth cost $\kappa_{i,t}$, larger for weakly supported additions when the scene improves little. We use $w_q = 1$ and $w_c = 0.1$. Both blur-aware terms are scaled by α at step $t + d_{\text{RL}}$, when the reward is computed. PPO (Schulman et al., 2017) updates the policy. Appendix B.5 gives the probe aggregation, descendant tracking, and gates.

Actions and acceptance. The delayed reward reflects a whole event, not individual actions, so proposals are also checked. We keep the LeGS rules that remove Gaussians that are never visible, nearly transparent, or overly large. When the latest ΔQ is negative, we turn the 25% of clone and split proposals with the lowest action probability into keep. While the policy is trained with PPO (*policy refinement*, up to update 15K), candidate deletions are tested on up to four fixed training views, separate from the probe views: a deletion is rejected, and becomes keep, if it increases the image objective, the input-formation error, or the supported signed-detail error beyond a small tolerance.

After policy refinement, a cleanup every 3K updates, which samples no actions, tests Gaussians with opacity below 0.1 on all training views. Low opacity alone does not make a Gaussian redundant, so we render the scene without each candidate set $\mathcal{D}_t$ and compare two features with the current render: RGB colors ($F = \text{RGB}$) and finest-scale signed Laplacian responses ($F = \text{Lap}$). The sharp value y of each feature entry p in view v is unknown; we assume that $|y - m_v(p)| \le \rho_v(p)$ for a center $m_v(p)$ and radius $\rho_v(p)$, where $m_v(p)$ is the prior's feature and $\rho_v(p)$ is the absolute difference between the prior's and the input's features (anchors use the input feature with zero radius). With $F_v(p)$ and $F_v^{\setminus\mathcal{D}_t}(p)$ the features of the current render and of the render without $\mathcal{D}_t$, the worst-case increase in



Figure 3: **Blurred inputs and our renders in Exblurf (Lee et al., 2023b) and Deblur-NeRF sequences (Ma et al., 2022).** Top: input images. Bottom: our renders from the same cameras.

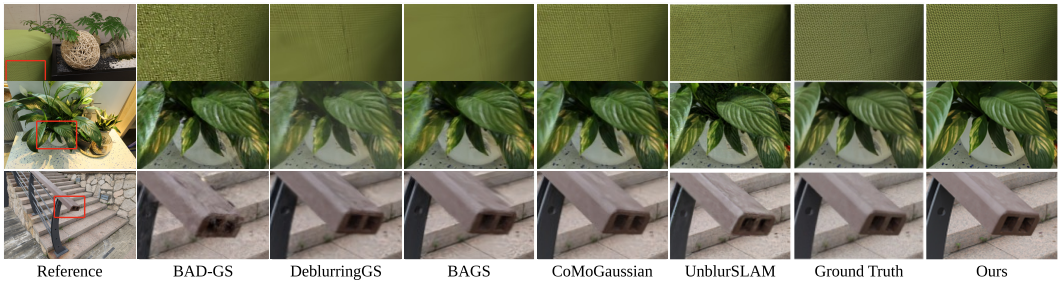

Figure 4: **Qualitative comparison on Deblur-NeRF motion blur.** The leftmost column shows the reference views; red boxes mark the enlarged regions, which highlight fabric texture, leaf boundaries, and railing geometry. Baseline panels are reproduced from Unblur-SLAM (Zhang et al., 2026); the last column shows our renders of the same regions.

squared error is

$$U_{F,v}(p) = \max_{|y-m_v(p)|\le\rho_v(p)} \Big[\big(F_v^{\setminus\mathcal{D}_t}(p) - y\big)^2 - \big(F_v(p) - y\big)^2 \Big]. \tag{11}$$

Averaging over valid entries of each view and then over views gives $\overline{U}_F(\mathcal{D}_t)$, the worst-case change in color error ($F = \mathrm{RGB}$) or edge error ($F = \mathrm{Lap}$). We accept the deletion only if neither error can increase:

$$\mathrm{accept}(\mathcal{D}_t) \iff \overline{U}_{\mathrm{RGB}}(\mathcal{D}_t) \le 0 \ \wedge\ \overline{U}_{\mathrm{Lap}}(\mathcal{D}_t) \le 0, \tag{12}$$

with both averages finite. This guarantee holds only if the intervals contain the true sharp values. If a set fails, we halve it, keeping the Gaussians with the smallest rendering contribution, and a cleanup may delete nothing. Appendix B.6 gives the search, and Appendix C.3 derives the bound.

Optimization and inference. Parameters are updated by the pretrained Learn2Splat optimizer (Pearl et al., 2026) for the first 2K updates and by Adam afterwards. After each structural change, optimization continues on the modified scene, whose renders update the prior confidence and inform the next event. Probe evaluations do not update the moving statistics. At inference, only the Gaussian scene is kept, and a novel camera is rendered as $I_{\mathrm{novel}} = \mathcal{R}_{\mathrm{RGB}}(\Theta_T, C_{\mathrm{novel}})$, where T is the final step.

4 Experiments

We first compare with existing methods, then ablate the supervision and capacity components, and finally analyze the Gaussian population.

Five real-world benchmark suites. We evaluate ten Deblur-NeRF motion scenes (Ma et al., 2022) and eleven Deblur-NeRF defocus scenes, PRISM3D-preprocessed ExBluRF severe-motion scenes (Lee et al., 2023b; Matta et al., 2026), three TUM-RGBD sequences (Sturm et al., 2012), and four fixed ScanNet sequences (Dai et al., 2017). Indoor source subsets and evaluation-frame mappings follow the released I^2-SLAM data (Bae et al., 2024).

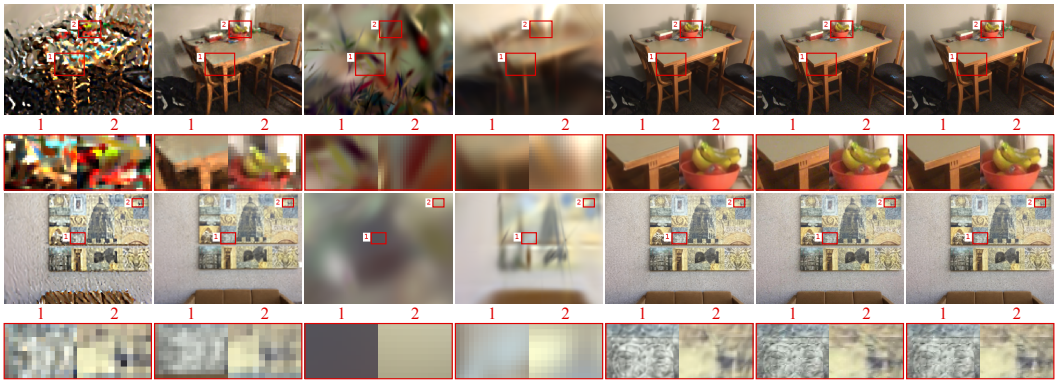

Figure 5: **Qualitative comparison on ScanNet.** For each example, the upper row shows full views and the lower row enlarges the red-boxed regions, highlighting a table joint and artwork texture. Ground Truth is the sharp reference image.

Table 2: **Cumulative module study on ScanNet and TUM.** Scene-macro averages over four fixed ScanNet and three fixed TUM sequences at 50K updates. M0–M4 are cumulative configurations; M5 denotes the full blur-aware system.

		ScanNet			TUM		
ID	Configuration	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
M0	Learn2Splat reference configuration	28.2859	0.8545	0.2133	29.9640	0.9232	0.0680
M1	+ input–prior confidence, single BPN	28.3415	0.8549	0.2100	30.0259	0.9233	0.0677
M2	+ coupled input/prior kernels	28.1135	0.8514	0.2073	29.8736	0.9117	0.0699
M3	+ recovery-adaptive confidence and surplus loss	28.1971	0.8514	0.2076	29.9843	0.9146	0.0684
M4	+ LeGS structural policy	30.8506	0.8985	0.1090	37.4996	0.9386	0.0395
M5	Full blur-aware system	**31.7221**	**0.9145**	**0.0887**	**40.4531**	**0.9540**	**0.0242**

Comparisons and metrics. The four comparison methods are PRISM3D (Matta et al., 2026), BAGS (Peng et al., 2024), Unblur-SLAM (Zhang et al., 2026), and DOF-GS (Wang et al., 2025). Table 1 reports native training budgets and aligned variants at 50K updates; 25×25 kernel alignment applies only to image-space blur kernels. All supplied source frames, including the marked evaluation set E, enter training. Final metrics are averaged over E within each scene and then equally across scenes; temporal partitions, where used, are pooled within the scene, with the update budget applied to each independently trained segment. PRISM3D on `scene0736_00` resumes saved Gaussians capped at 100K without restoring optimizer moments. The full input or reference rules and adapters are in Appendix A. Our results use the full method of Sec. 3. We report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). For our method, we report the final checkpoint at 50K updates; the ablations in Tables 2 and 2 use the 50K checkpoint. Implementation details, including the optimizer switch, kernel settings, and metric implementation, are given in Appendix B.

Qualitative results. Figure 3 pairs eight blurred input crops (top) with our renders from the same cameras (bottom). Figures 4 and 5 compare our renders with those of the baselines on Deblur-NeRF motion blur and on ScanNet, respectively, and Figure 7 shows fine-detail comparisons on two ExBluRF scenes. The baseline panels of Figures 4 and 7 are reproduced from Unblur-SLAM (Zhang et al., 2026) and PRISM3D (Matta et al., 2026), respectively, and our renders are shown for the same views and crops. These two figures therefore also include methods that are not evaluated in Table 1 (BAD-GS, DeblurringGS, Restormer, ExBluRF (Lee et al., 2023b), and Deblur-GS), which serve only as visual references. Figure 8 visualizes the reconstructed Gaussian scenes. Figure 6 shows, for two fixed cameras, the input image I_v, the deblurred image I'_v, and our render.

Which part of the loop matters? Tables 2 and 2 report cumulative ablations at 50K updates on four ScanNet scenes and the three TUM sequences. M0 is the Learn2Splat (Pearl et al., 2026) baseline trained on the input images. M1–M3 progressively add the adaptive-supervision (AS) components of Sec. 3.2: M1 adds the initial input–prior confidence and a single BPN, M2 couples the input and prior branches through the shared kernel of Eq. (2), and M3 adds the recovery-adaptive confidence and the signed surplus loss of Eqs. (6)–(7). M4 adds the original LeGS structural policy (Ning et al., 2026).

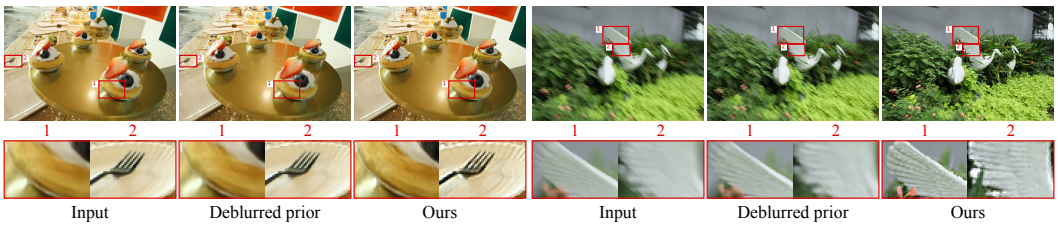

Figure 6: **Input, deblurred image, and render.** Rows show `defocus_cake` (frame 023) and `motion_blurheron` (frame 034); each row uses one camera. Left: input image I_v. Middle: deblurred image I'_v. Right: our render after 50K updates.

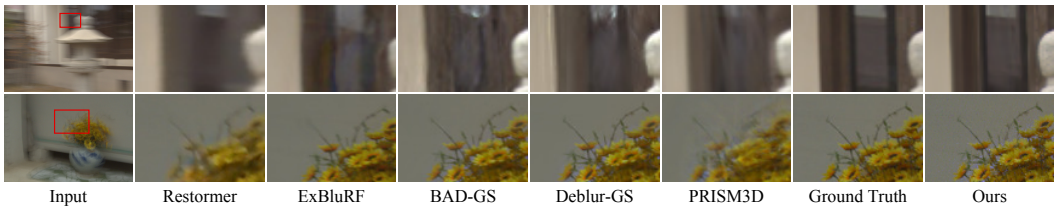

Figure 7: **Fine-detail comparisons on Stone Lantern and Sunflowers.** Top: Stone Lantern. Bottom: Sunflowers. Red boxes indicate the regions enlarged across methods. Baseline and ground-truth images are reproduced from PRISM3D (Matta et al., 2026); ExBluRF, BAD-GS, and Deblur-GS use the source paper's oracle initialization.



Figure 8: **Qualitative 3D Gaussian visualizations on Exblurf (Lee et al., 2023b).** From left to right: Dragon, Jars2, and Stone Lantern. For each scene, the top panel presents the reconstructed 3D Gaussian scene, while the bottom row shows two corresponding views rendered from our method. The irregular Gaussian fringes are retained to visualize the spatial extent of the reconstructed scene.

M5 is the complete system, which replaces this policy with the blur-aware reinforcement learning policy of Sec. 3.3, including its structural checks. M0–M3 do not add Gaussians during training and keep about 69K Gaussians on TUM, close to the initial population, so the supervision components change the metrics only slightly. The main gains appear once the population can grow (M4, 1.93M Gaussians on TUM) and when the policy uses the blur and recovery signals of AS (M5). Compared with M4, M5 improves PSNR by 0.87 dB on ScanNet and 2.95 dB on TUM, and SSIM and LPIPS also improve on both datasets.

5 Discussion and Conclusion

Blur affects both the evidence used to learn a scene and the places where the scene needs more capacity. SMART-GS makes these two decisions together. Input blur sets the first supervision weights. Rendered blur and recovered detail then revise the loss. The same state guides Gaussian growth and removal through delayed training probes. At inference, the learned sharp Gaussian representation is enough to render new sharp views.

Reproducibility Statement

The supplementary material specifies the update order, observation model, recovery statistics, policy feedback, and dataset protocols. Editable figures and a complete comparison-table skeleton accompany the source. The primary comparison uses all supplied observed views for scene construction and scores renders against the designated sharp references; it is an observed-camera reconstruction protocol, not a camera-disjoint novel-view benchmark. Final experimental configurations, input memberships, metric implementations, and run outputs accompany the completed evaluation. We would pulish our code after the acceptance.

AI Use Statement

Generative AI assisted writing, literature review, notation checks, experiment planning, and conceptual figure design. The editable figures reuse supplied experimental images; missing measurements and competitor outputs were not generated. The authors remain responsible for verifying the manuscript, experiments, and any required disclosure.

References

Gwangtak Bae, Changwoon Choi, Hyeongjun Heo, Sang Min Kim, and Young Min Kim. I^2-SLAM: Inverting imaging process for robust photorealistic dense slam. In *Computer Vision – ECCV 2024*, volume 15085 of *Lecture Notes in Computer Science*, pp. 72–89. Springer, 2024. doi: 10.1007/978-3-031-73383-3_5.

Timothy Chen, Ola Shorinwa, Joseph Bruno, Aiden Swann, Javier Yu, Weijia Zeng, Keiko Nagami, Philip Dames, and Mac Schwager. Splat-Nav: Safe real-time robot navigation in gaussian splatting maps. *IEEE Transactions on Robotics*, 41:2765–2784, 2025. doi: 10.1109/TRO.2025.3552348.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

Niladri Shekhar Dutt, Zifan Shi, Paul Guerrero, Chun-Hao Paul Huang, Duygu Ceylan, Niloy J. Mitra, and Xuelin Chen. Lost: Level of semantics tokenization for 3d shapes. *arXiv preprint arXiv:2603.17995*, 2026.

Amirhosein Ghasemabadi, Muhammad Kamran Janjua, Mohammad Salameh, and Di Niu. Learning truncated causal history model for video restoration. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=cUGf2HaNcs.

Osama Jamil and AnnMarie Brennan. Immersive heritage through gaussian splatting: a new visual aesthetic for reality capture. *Frontiers in Computer Science*, 7:1515609, 2025. doi: 10.3389/fcomp.2025.1515609.

Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. VR-GS: A Physical Dynamics-Aware Interactive Gaussian Splatting System in Virtual Reality. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657448. URL https://arxiv.org/abs/2401.16663.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, July 2023. doi: 10.1145/3592433.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19729–19739, October 2023.

Lingshun Kong, Jiangxin Dong, Jinhui Tang, Ming-Hsuan Yang, and Jinshan Pan. Efficient visual state space model for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12710–12719, June 2025.

Dogyoon Lee, Minhyeok Lee, Chajin Shin, and Sangyoun Lee. DP-NeRF: Deblurred neural radiance field with physical scene priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12386–12396, June 2023a.

Dongwoo Lee, Jeongtaek Oh, Jaesung Rim, Sunghyun Cho, and Kyoung Mu Lee. ExBluRF: Efficient radiance fields for extreme motion blurred images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17639–17648, 2023b. URL https://openaccess.thecvf.com/content/ICCV2023/html/Lee_ExBluRF_Efficient_Radiance_Fields_for_Extreme_Motion_Blurred_Images_ICCV_2023_paper.html.

Jungho Lee, Suhwan Cho, Taeoh Kim, Ho-Deok Jang, Minhyeok Lee, Geonho Cha, Dongyoon Wee, Dogyoon Lee, and Sangyoun Lee. CoCoGaussian: Leveraging circle of confusion for Gaussian Splatting from defocused images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16101–16110, June 2025a.

Jungho Lee, Donghyeong Kim, Dogyoon Lee, Suhwan Cho, Minhyeok Lee, Wonjoon Lee, Taeoh Kim, Dongyoon Wee, and Sangyoun Lee. CoMoGaussian: Continuous motion-aware gaussian splatting from motion-blurred images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26415–26424, October 2025b.

Cristián Llull, Nelson Baloian, Benjamin Bustos, Kornelius Kupczik, Ivan Sipiran, and Andrés Baloian. Evaluation of 3d reconstruction for cultural heritage applications. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 1634–1643, 2023. doi: 10.1109/ICCVW60793.2023.00179.

Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. Deblur-NeRF: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12861–12870, June 2022.

Gopi Raju Matta, Trisha Reddypalli, Divya Madhuri Vemunuri, and Kaushik Mitra. PRISM3D: Probabilistic refinement and robust initialization for physically consistent scene modeling under extreme motion blur. In *Computer Vision – ECCV 2026*, 2026. URL https://arxiv.org/abs/2607.03855. Accepted at ECCV 2026; arXiv:2607.03855; final proceedings metadata pending.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zhenhua Ning, Xin Li, Jun Yu, Guangming Lu, Yaowei Wang, and Wenjie Pei. Beyond heuristics: Learnable density control for 3D Gaussian Splatting. arXiv:2605.00408, 2026. URL https://arxiv.org/abs/2605.00408.

Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Visual Localization using Imperfect 3D Models from the Internet . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13175–13186, Los Alamitos, CA, USA, June 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01266. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01266.

Naama Pearl, Stefano Esposito, Haofei Xu, Amit Peleg, Patricia Gschossmann, Lorenzo Porzi, Peter Kontschieder, Gerard Pons-Moll, and Andreas Geiger. Learn2Splat: Extending the horizon of learned 3DGS optimization. arXiv:2605.15760, 2026. URL https://arxiv.org/abs/2605.15760.

Cheng Peng, Yutao Tang, Yifan Zhou, Nengyu Wang, Xijun Liu, Deming Li, and Rama Chellappa. BAGS: Blur agnostic gaussian splatting through multi-scale kernel modeling. In *Computer Vision – ECCV 2024*, volume 15138 of *Lecture Notes in Computer Science*, pp. 293–310. Springer, 2024. doi: 10.1007/978-3-031-72989-8_17.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D Language Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024. URL https://arxiv.org/abs/2312.16084.

Chen Rao, Guangyuan Li, Zehua Lan, Jiakai Sun, Junsheng Luan, Wei Xing, Lei Zhao, Huaizhong Lin, Jianfeng Dong, and Dalong Zhang. Rethinking video deblurring with wavelet-aware dynamic transformer and diffusion model. In *European Conference on Computer Vision*, pp. 421–437. Springer, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Hossein Talebi and Peyman Milanfar. NIMA: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, August 2018. doi: 10.1109/TIP.2018.2831899.

Yujie Wang, Praneeth Chakravarthula, and Baoquan Chen. DOF-GS: Adjustable depth-of-field 3D Gaussian Splatting for post-capture refocusing, defocus rendering and blur removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21297–21306, June 2025.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, April 2004. doi: 10.1109/TIP.2003.819861.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. doi: 10.1109/CVPR52688.2022.00564. URL https://arxiv.org/abs/2111.09881.

Qi Zhang, Denis Rozumny, Francesco Girlanda, Sezer Karaoglu, Marc Pollefeys, Theo Gevers, and Martin R. Oswald. Unblur-SLAM: Dense neural slam for blurry inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 352–362, June 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, June 2018.

Yuhang Zheng, Xiangyu Chen, Yupeng Zheng, Songen Gu, Runyi Yang, Bu Jin, Pengfei Li, Chengliang Zhong, Zengmao Wang, Lina Liu, Chao Yang, Dawei Wang, Zhen Chen, Xiaoxiao Long, and Meiqing Wang. GaussianGrasper: 3d language gaussian splatting for open-vocabulary robotic grasping, 2024. URL https://arxiv.org/abs/2403.09637.