VeilGS: Solving Darkness as a Veil of 2D Basis Maps over Retinex 3DGS

Mohab Kishawy[1,†] Chengzhou Tang[2] Shahram Shirani[1] Jun Chen[1]

[1]Department of Electrical & Computer Engineering, McMaster University, Hamilton, Ontario, Canada

[2]Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada

[†]Corresponding author: kishawym@mcmaster.ca

Project page: https://errorlogic1211.github.io/VeilGS/



(a) Visual results (b) Quantitative scores

Figure 1. **VeilGS treats low-light degradation as a veil drawn over a well-lit scene.** (a) Held-out views of Gaussian-DK [43]: the solved veil is high where light falls and low in shadow. (b) Mean scores, efficiency and user-study MOS; outward is better.

Abstract

In a dark capture, the darkness is not uniform. How much light reaches a surface depends on where it lies relative to the few light sources, and the camera compresses what remains with a tone curve. Most low-light 3D reconstruction methods either apply one curve to every pixel of a view, so a dark object beside a lamp and a bright one in an unlit corner receive the same correction, or store the darkness in the 3D representation, where it can absorb scene detail. We present VeilGS, a 3D Gaussian Splatting method for low-light novel view synthesis that treats the degradation as a veil drawn over a clear scene. Each Gaussian carries an albedo and a shading latent instead of spherical harmonics, separating surface colour from illumination. Each view is darkened by a spatially varying gain followed by a global tone curve, and rendering without this veil gives the restored scene. A small network predicts a coarse basis for the gain from the observation, and its coefficients are solved for each view rather than learned. Training through this solve discourages the Gaussians from absorbing what the veil can explain. VeilGS needs no paired normal-light supervision and stores no per-view parameters. On Gaussian-DK, its restored renders score best on no-reference NIQE and CLIP-IQA and are rated highest in a blind user study with 55 participants. On LOM, it achieves results competitive with the state of the art.

1. Introduction

Neural Radiance Fields (NeRF) [29] and 3D Gaussian Splatting (3DGS) [17] have brought unprecedented progress to novel view synthesis, with applications from virtual reality to autonomous driving and robotics. However, both methods require well-exposed inputs. In settings such as nighttime driving, surveillance and robotic exploration of unlit spaces, the captured images are dark, where standard NeRF and 3DGS pipelines fail to produce usable novel views.

3D representations such as NeRF and 3DGS store the colour of each point as it appears to the camera, which causes the darkness to be baked into the representation itself. The darkness is also spatially non-uniform, since surfaces are located at different positions relative to the light sources. Enhancing each image with a 2D low-light method [4, 13, 25] before reconstruction is ineffective [5, 6, 38, 45], because each image is enhanced on its own, which breaks the multi-view consistency that 3D reconstruction relies on, creating floaters and colour drift.

To overcome this challenge, several methods handle the degradation inside the 3D representation. LLNeRF [38] decomposes each 3D point into view-independent and view-dependent components and Aleth-NeRF [5] explains darkness with concealing fields, but both inherit the slow training and rendering of NeRF. On 3DGS, Gaussian-DK [43] decodes a lightness map from per-Gaussian light features and the exposure metadata of each image, Luminance-GS [6] learns enhanced Gaussian colours from training

views corrected by a per-view colour transform and tone curve, and LITA-GS [45] renders an illumination map from per-Gaussian features, guided by illumination-invariant structure priors. These models have two limitations. First, a curve applied uniformly over a view, as in Luminance-GS, maps each observed intensity to the same output wherever it occurs. Unless the darkening is global, a dark object beside a lamp and a bright object in an unlit corner may record the same value, yet need different corrections (Fig. 1). Second, a spatially varying degradation stored in the Gaussians, as in Gaussian-DK and LITA-GS, has the same resolution as the scene, so it can become entangled with scene detail.

These limitations raise a key question: *can low-light degradation be modelled as a spatially varying, physically motivated quantity that is inferred from each view rather than learned for it, yet coarse enough to leave the scene's texture to the Gaussians?*

To answer it, we propose VeilGS, which treats low-light degradation as a *veil* drawn over a well-lit scene (Fig. 1): the Gaussians hold the clean scene, and the veil holds the darkness. First, we replace the spherical harmonics of each Gaussian with an albedo and a shading latent, so that the effect of light is separable from the physical colour of the object, following Retinex theory [20]. Second, we model the veil as a spatially varying gain followed by a global camera tone curve, and span the gain with a coarse basis that a small network reads off the observation. Third, rather than learning per-view coefficients for this basis, we solve for them from each observation with a differentiable bilinear rank-one solver. Finally, we train the scene through the solve, which discourages the Gaussians from absorbing what the veil can explain, in the spirit of variable projection [10]. VeilGS needs no paired normal-light supervision, stores no per-image parameters, and renders the restored scene without the veil.

Our contributions are summarized as follows:

- We propose VeilGS, an unsupervised 3DGS framework for low-light novel view synthesis that models the degradation as a veil drawn over a well-lit scene.
- We replace per-Gaussian spherical harmonics with an albedo and a shading latent, with priors on each that encourage separating surface colour from the effect of light, producing a restored radiance for the veil to act on.
- We represent the veil as a coarse, spatially varying gain kept outside the Gaussians and solved from each observation, and train the scene through this solve, in the spirit of variable projection. To our knowledge, VeilGS is the first low-light 3DGS method that solves, rather than stores, its per-view degradation.
- Extensive experiments on the LOM and Gaussian-DK benchmarks and a user study with 55 participants demonstrate that VeilGS compares favourably with state-of-the-art NeRF- and 3DGS-based low-light methods.

2. Related Work

2.1. Novel View Synthesis under Challenging Lighting

2D low-light methods either build on the Retinex model [4, 14, 20, 42] or learn to enhance without paired data [13, 15, 25]. Applied to each view of a capture on its own, they break the multi-view consistency that 3D reconstruction relies on, so recent work models the lighting inside the 3D representation.

NeRF-based methods. NeRF [29] renders an implicit field by volume rendering and assumes well-exposed, photometrically consistent inputs. When the lighting changes across views, NeRF-W [28] absorbs the change into per-view appearance embeddings, and BilaRF [40] into per-view bilateral grids that mimic the camera pipeline. Both reproduce the lighting of the inputs rather than restore it. RAW-NeRF [30] recovers dark scenes, but only from RAW data. For dark sRGB captures, later methods model the darkness itself. LLNeRF [38] decomposes each point into view-independent and view-dependent components, and Aleth-NeRF [5] explains darkness with concealing fields that are removed at test time. RoSe [21] estimates an illumination transition in 3D, and I^2-NeRF [23] models light–media interactions with a unified radiative equation. All of them inherit the slow training and rendering of implicit fields.

3DGS-based methods. 3DGS [17] replaces the implicit field with explicit anisotropic Gaussians and tile-based rasterization, reaching comparable quality in real time. To handle appearance variance, WildGaussians [19] brings the per-view appearance embeddings of NeRF-W to 3DGS. Low-light 3DGS methods model the darkness in one of two ways. Luminance-GS [6] learns enhanced Gaussian colours from training views corrected as a whole by a per-view colour transform and tone curve, and Luminance-GS++ [7] adds pixel-wise residual refinement. Gaussian-DK [43] and LITA-GS [45] instead let the darkness vary across the image through features stored in each Gaussian. Gaussian-DK decodes a lightness map from these features and each image's exposure metadata, and LITA-GS renders an illumination map from them, guided by illumination-invariant structure priors and a progressive denoiser. Beyond these, LL-Gaussian [35] guides a reflectance–illumination decomposition with a diffusion prior, and RestorGS [32] restores several degradations in one depth-aware framework. VeilGS takes a third route: its veil varies across the image but stays coarse and outside the Gaussians, and is solved from each view's image rather than learned.

2.2. Learned Bases and Differentiable Solvers

A common way to keep a per-image quantity simple is to write it in a small basis. Camera responses are well described by a low-dimensional linear basis [12], and HDR-

Net [9] infers a coarse grid of local affine coefficients from a downsampled input. BA-Net [36] and Tang *et al*. [37] go further and solve for the coefficients rather than predict them. Solving for the linear coefficients in closed form and optimizing only the remaining parameters is classical variable projection [10, 33]. Gradients can pass through such a solve by argmin differentiation [11], optimization layers [1] or closed-form ridge regression [2], so the networks around it can be trained end to end. These solvers are, however, typically trained with ground-truth supervision. VeilGS places such a solve inside the optimization of a 3D scene, where neither the scene nor its degradation has ground truth.

3. Method

Given N posed low-light sRGB images $\{I_n\}_{n=1}^{N}$ of a static scene without normal-light reference, VeilGS recovers 3D Gaussians whose render J is the scene's clean radiance (Fig. 2). For a training view n, we rasterize J_n from an albedo–shading decomposition (Sec. 3.2), and a basis network reads I_n to propose a subspace for the spatially varying gain of the low-light forward model (Sec. 3.3). The coefficients in that subspace are solved to explain I_n (Sec. 3.4), the reconstruction loss is back-propagated through the solve (Sec. 3.5), and weak global priors select what the observations leave open (Sec. 3.6).

3.1. Preliminaries: 3D Gaussian Splatting

3DGS [17] represents a scene explicitly as a set of anisotropic 3D Gaussians $\{G_i\}_{i=1}^{M}$. Each Gaussian is defined by a centre $\mu_i \in \mathbb{R}^3$ and a covariance $\Sigma_i \in \mathbb{R}^{3\times3}$,

$$G_i(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^\top \Sigma_i^{-1}(\mathbf{x}-\mu_i)}, \tag{1}$$

where Σ_i is factored into scaling S_i and rotation R_i,

$$\Sigma_i = R_i S_i S_i^\top R_i^\top, \tag{2}$$

which, with positive scales, keeps Σ_i positive definite. Each Gaussian also carries an opacity $o_i \in [0,1]$ and a view-dependent colour c_i, represented by spherical harmonics coefficients. To render a view, the Gaussians are projected onto the image plane, and the colour of a pixel $\mathbf{p}$ is obtained by alpha-blending the depth-sorted Gaussians that overlap it,

$$C(\mathbf{p}) = \sum_i c_i\, \omega_i(\mathbf{p}) \prod_{i'<i} \big(1-\omega_{i'}(\mathbf{p})\big), \tag{3}$$

where $\omega_i(\mathbf{p})$ is the opacity o_i weighted by the projected 2D Gaussian at $\mathbf{p}$. The Gaussians are optimized with an ℓ_1 and D-SSIM loss, with adaptive densification and pruning [17]. VeilGS keeps this geometry, rasterization and densification unchanged, and changes only what c_i represents (Sec. 3.2) and how the rendered radiance is transformed into the observed low-light image (Secs. 3.3–3.5).

3.2. Albedo–Shading Decomposition

The spherical harmonics colour c_i of Eq. (3) describes how a point *looks*, which under a dark capture includes the darkness. We factorize it into an albedo $a_i \in \mathbb{R}^3$ and a shading $s_i \in \mathbb{R}^3_{>0}$ produced by a small MLP f_θ shared by all Gaussians,

$$s_i = f_\theta(z_i, d_i), \qquad c_i = a_i \odot s_i, \tag{4}$$

where $z_i \in \mathbb{R}^8$ is a per-Gaussian shading latent and d_i is the viewing ray direction from the camera centre to the Gaussian centre μ_i. Following Retinex theory [20], the albedo carries the surface colour, while the shading models the light it receives in the scene. Since the decomposition of $a_i \odot s_i$ is ambiguous, we regularize the decomposition with the priors introduced in Sec. 3.6. The decomposition matters because a spherical-harmonic colour can render a shadowed surface as a dark one, whereas here the albedo should be a reflectance in $[0,1]$ similar to its neighbours and the shading should be grey and bounded, so the shadows in the scene are separable from the surface colour. A Softplus output makes shading positive, so it can scale albedo but never invert it. View-dependent effects remain representable, but they now live in the shading.

The composited colour of Eq. (3) is the clean radiance $J \in \mathbb{R}^{H\times W\times3}_{\geq0}$, a canonical restored appearance that keeps the shading but not the per-view attenuation.

3.3. The Veil

We model a low-light image with a simple camera model, in which a pixel is the camera's tone curve applied to the light reaching the sensor [8, 12], and let that light be the clean radiance scaled by an illumination map, as in Retinex [14, 20]:

$$\hat{I} = \big(\alpha \odot J\big)^{\frac{1}{\gamma}}, \tag{5}$$

where $\alpha \in (0,1]$ is the gain, which we call the *veil*, and $1/\gamma$ is the exponent of the camera's tone curve [24], one number per scene. In all experiments, we average this curve with a luminance-preserving one (Sec. S1). The veil has one value per pixel, shared by the three colour channels, so before clipping it changes brightness but not colour. It varies across the image, since pixels near a lamp receive more light. The restored view is J itself, without the veil and the tone curve, clipped to $[0,1]$. Because scaling J up and α down leaves $\hat{I}$ unchanged, an exposure prior selects the brightness of J (Sec. 3.6).

Representing the veil. The gain is a global level times a local modulation, $\alpha = b_\alpha \exp(m)$, where $b_\alpha > 0$ is learned and m raises or lowers the level locally; the forward model clips α at 1. The map m should be smooth, which discourages it from absorbing scene texture. A small network h_ϕ reads the dark image at one-eighth resolution and outputs

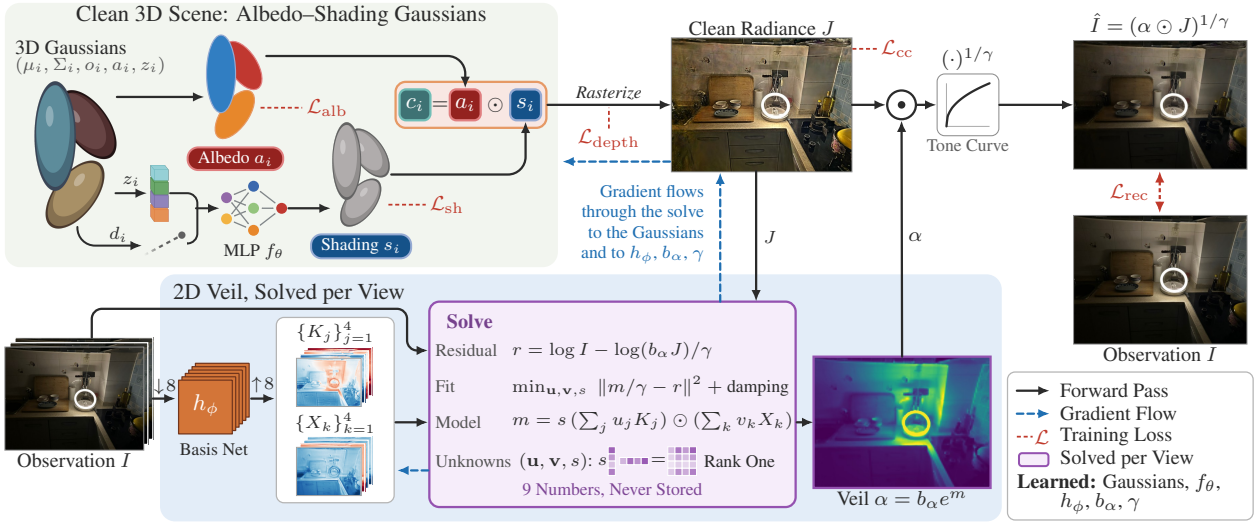


Figure 2. **Overview of VeilGS.** The Gaussians render a clean radiance J, the restored view. A basis network reads the observation I and proposes smooth basis maps, and a differentiable solve finds the coefficients under which the low-light model applied to J best explains I, giving the gain α and the prediction $\hat{I}$. The reconstruction loss is back-propagated through the solve. No clean image is used.

two sets of $d = 4$ coarse maps,

$$\{K_j\}_{j=1}^{d},\ \{X_k\}_{k=1}^{d} = \tanh\Big(\mathcal{U}_8\, h_\phi(\mathcal{P}_8\, I_n)\Big), \quad (6)$$

where $\mathcal{P}_8$ and $\mathcal{U}_8$ shrink and enlarge by a factor of 8. Following Tang *et al.* [37], m combines the two sets bilinearly, with a pixel-wise product where they use a convolution,

$$m = s\Big(\sum_j u_j K_j\Big) \odot \Big(\sum_k v_k X_k\Big) = \sum_{j,k} M_{jk}\, \Phi_{jk}, \quad (7)$$

with unit vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, a scale $s \geq 0$, $\Phi_{jk} = K_j \odot X_k$ and $M = s\,\mathbf{u}\mathbf{v}^\top$. Each view's veil is thus nine numbers $(\mathbf{u}, \mathbf{v}, s)$, computed at every step (Sec. 3.4) rather than stored. The reason behind choosing a rank-one product over a sum of the eight maps is that with a similar number of free coefficients per view (seven against eight), it draws on the 16 maps Φ_{jk} and lets one set scale the other. So the gain can take more shapes, while the rank-one constraint on M keeps it coarse, which discourages copying scene texture.

3.4. Solving for the Coefficients

For each view, $M = s\,\mathbf{u}\mathbf{v}^\top$ is chosen so that the prediction $\hat{I}$ of Eq. (5) matches the observation I, with the clean render J, the global level b_α, the tone-curve parameter γ and the basis maps held fixed.

The residual. Taking logs turns the multiplication by the veil into an addition. With $\alpha = b_\alpha \exp(m)$, the log of Eq. (5) at pixel $\mathbf{p}$ and colour channel ℓ is

$$\log \hat{I}(\mathbf{p}, \ell) = \underbrace{\tfrac{1}{\gamma} \log\big(b_\alpha J(\mathbf{p}, \ell)\big)}_{\text{known}} + \underbrace{\tfrac{1}{\gamma}\, m(\mathbf{p})}_{\text{veil}}. \quad (8)$$

Matching it to the observed $\log I(\mathbf{p}, \ell)$, the veil's part must equal the *residual*

$$r(\mathbf{p}, \ell) = \log I(\mathbf{p}, \ell) - \tfrac{1}{\gamma} \log\big(b_\alpha J(\mathbf{p}, \ell)\big), \quad (9)$$

the part of the observed log image that the clean render and the global level do not explain.

The fit. Since m is a weighted sum of the 16 maps, the nine numbers of the veil are found by a rank-constrained least-squares fit,

$$\min_{s,\mathbf{u},\mathbf{v}} \sum_{\mathbf{p},\ell} \Big(\tfrac{1}{\gamma} m(\mathbf{p}) - r(\mathbf{p}, \ell)\Big)^2 + \text{vec}(M)^\top \Lambda\, \text{vec}(M), \quad (10)$$

where m is the bilinear form of Eq. (7), so the fit is over the nine numbers rather than over the 16 entries of $M = s\,\mathbf{u}\mathbf{v}^\top$. If the maps Φ_{jk}/γ form the 16 columns of a matrix A, with one row per pixel and channel, the first term is $\|A\ \text{vec}(M) - r\|^2$. The second term is a Marquardt-style penalty [27]. It keeps the coefficients small, penalizing each entry of M in proportion to how strongly its map acts on the image, so the fit stays well conditioned as γ changes.

Solver. We solve Eq. (10) in three steps, adapting the solver of Tang *et al.* [37]. (1) Ignoring the rank constraint leaves a damped least-squares problem, which we solve in closed form with a QR factorization. (2) Power iteration approximates the rank-one matrix closest to that solution, which gives $\mathbf{u}$, $\mathbf{v}$ and s; it starts from a fixed vector rather than a random one. (3) Four Gauss–Newton steps adjust $\mathbf{u}$, $\mathbf{v}$ and s to fit Eq. (10) better. The three steps run on 64×64 downsized copies of I, J and the basis maps. As the veil is shared by the channels, only the channel mean of r matters,

giving 4,096 equations for 16 unknowns and a few small linear solves per training iteration. The resulting $(\mathbf{u}, \mathbf{v}, s)$ define the full-resolution veil through Eq. (7). Sec. S3 details all three steps.

3.5. Training Through the Solve

The solved coefficients depend on what we train: the Gaussians and the global level enter the residual r, and the basis network and γ enter A. Rather than detaching the solve, we back-propagate through it, so the gradient also includes how the coefficients change when the scene changes. A change the veil can undo is then partly cancelled by the re-fit, so the Gaussians receive less gradient to absorb it.

Without the rank constraint, the fit leaves the residual

$$r^\star = \Big(\mathbf{I} - A\big(A^\top A + \Lambda\big)^{-1} A^\top\Big)\, r, \tag{11}$$

which, without damping, is exactly the part of r that the veil cannot explain. In this idealized setting, solving for the coefficients and training the rest on $r^\star$ is variable projection [10]. Our fit is damped and rank-one, and the reconstruction loss differs from its objective, so this view only motivates training through the solve: Table 3 shows its benefit empirically, and Sec. S4 analyses the gradient.

Keeping the biases valid. Gradients through the solve can push b_α or γ out of their valid range. Once outside, the clipping in the forward model passes no gradient, so they never return and the veil stops training. After every optimizer step we therefore clip both back into range.

3.6. Training Objectives

We train the Gaussians, the shading MLP, the basis network and the physical biases end to end. Only the reconstruction loss compares $\hat{I}$ with I; the other three losses are priors, and none of them compares J with a reference image.

Reconstruction. Following 3DGS [17],

$$\mathcal{L}_{\text{rec}} = (1-\lambda_{\text{s}})\,\|\hat{I} - I\|_1 + \lambda_{\text{s}}\big(1 - \text{SSIM}(\hat{I}, I)\big), \tag{12}$$

with $\lambda_{\text{s}} = 0.2$. This loss sees J only through the veil, so it cannot fix the brightness or the albedo–shading split of J, and fixes its colour only up to γ.

Colour correction. Two priors set the brightness that $\mathcal{L}_{\text{rec}}$ leaves open and balance the colour,

$$\mathcal{L}_{\text{cc}} = \lambda_{\text{exp}}\,\mathcal{L}_{\text{exp}} + \lambda_{\text{gw}}\,\mathcal{L}_{\text{gw}}, \tag{13}$$

with $\lambda_{\text{exp}} = 0.3$ and $\lambda_{\text{gw}} = 0.05$. Inspired by Zero-DCE [13], the exposure term pulls the mean of J toward a target $E = 0.4$, as in RestorGS [32],

$$\mathcal{L}_{\text{exp}} = \Big|\frac{1}{3|\Omega|}\sum_{\mathbf{p}\in\Omega}\sum_{\ell\in\{R,G,B\}} J(\mathbf{p},\ell) \;-\; E\Big|, \tag{14}$$

where Ω is the set of pixels. Since a dark capture gives no reliable colour reference, the gray-world term follows the colour constancy loss of Zero-DCE and asks the channel means of J to be equal,

$$\mathcal{L}_{\text{gw}} = \sum_{(\ell,\ell')\in\mathcal{P}} \big(\bar{J}_\ell - \bar{J}_{\ell'}\big)^2, \quad \bar{J}_\ell = \frac{1}{|\Omega|}\sum_{\mathbf{p}\in\Omega} J(\mathbf{p},\ell), \tag{15}$$

where $\mathcal{P} = \{(R,G),(G,B),(R,B)\}$.

Retinex. Following Retinex theory [20], we constrain each factor of Eq. (4),

$$\mathcal{L}_{\text{retinex}} = \lambda_{\text{alb}}\,\mathcal{L}_{\text{alb}} + \lambda_{\text{sh}}\,\mathcal{L}_{\text{sh}}, \tag{16}$$

with $\lambda_{\text{alb}} = 0.5$ and $\lambda_{\text{sh}} = 0.02$. The albedo should lie in $[0,1]$ and be similar across neighbouring Gaussians,

$$\mathcal{L}_{\text{alb}} = \mathbb{E}_i\Big[\rho_{[0,1]}(a_i) + \mu_a \sum_{i'\in\mathcal{N}(i)} w_{ii'}\,\|a_i - a_{i'}\|^2\Big], \tag{17}$$

where $\mathcal{N}(i)$ holds the 8 nearest Gaussians in 3D, $w_{ii'}$ decays with distance, $\rho_{[l,h]}$ is zero inside $[l,h]$ and linear outside, and $\mu_a = 0.01$. The shading should be grey, so that colour lives in the albedo, and lie between $s_{\min} = 0.1$ and $s_{\max} = 1.5$:

$$\mathcal{L}_{\text{sh}} = \mathbb{E}_i\Big[\|s_i - \bar{s}_i\mathbf{1}\|^2 + \mu_s\,\rho_{[s_{\min},s_{\max}]}(s_i)^2\Big], \tag{18}$$

where $\bar{s}_i$ is the mean of the three channels of s_i and $\mathbf{1}$ is the all-ones vector, so that $\bar{s}_i\mathbf{1}$ is a grey shading with the same mean, and $\mu_s = 2.5$.

Depth. The inverse depth D, blended from per-Gaussian depths as in Eq. (3), follows a monocular estimate $\tilde{D}$ from Depth Anything 3 [22] on the dark images, both normalized to $[0,1]$ [18],

$$\mathcal{L}_{\text{depth}} = \|D - \tilde{D}\|_1. \tag{19}$$

Total loss. The total loss is

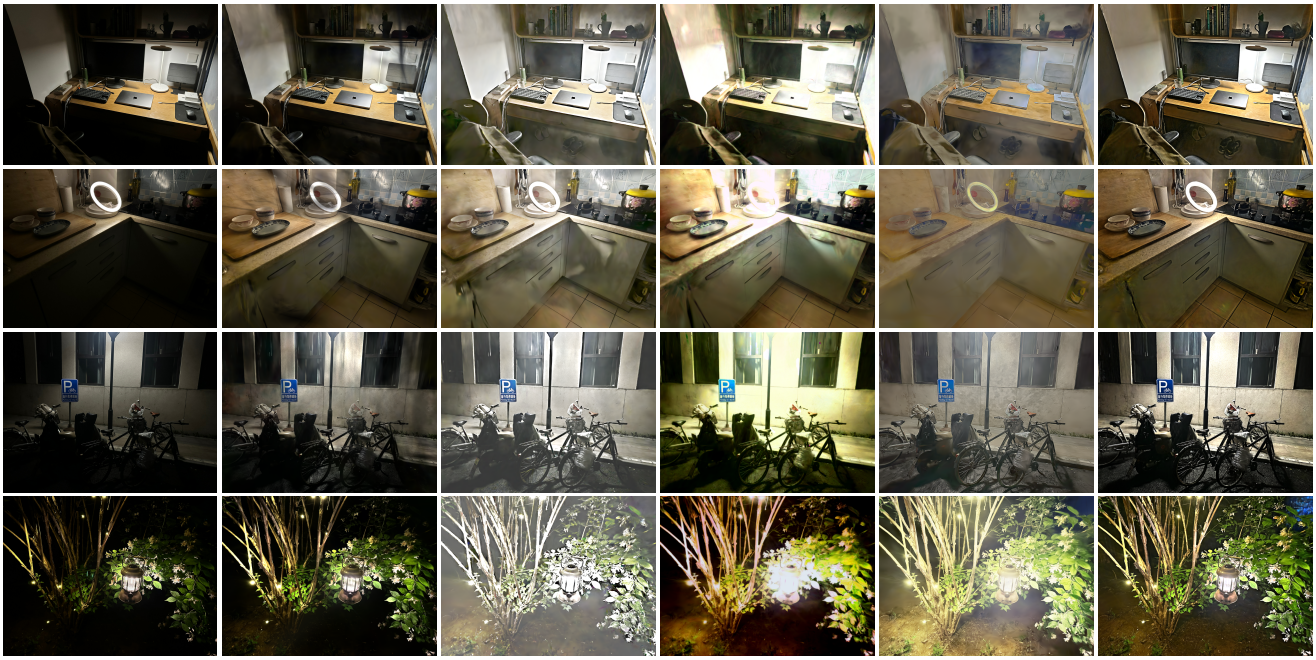
$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cc}} + \mathcal{L}_{\text{retinex}} + \lambda_{\text{d}}\,\mathcal{L}_{\text{depth}}, \tag{20}$$

with λ_{d} decaying from 0.2 to 0.01 during training. All weights, including E, are the same for every scene and both benchmarks, and no hyperparameter was tuned on normal-light frames.

4. Experiments

4.1. Experimental Setup

Evaluation Datasets and Metrics. We evaluate on two real low-light benchmarks. The Gaussian-DK night benchmark [43] has twelve scenes of 75–131 dark views with varying exposure and no normal-light reference. We train at quarter resolution and hold out 9–15 views per scene. We score the restored view, which for VeilGS



Dark input | 3DGS [17] | Luminance-GS [6] | LITA-GS [45] | Gaussian-DK [43] | VeilGS (ours)

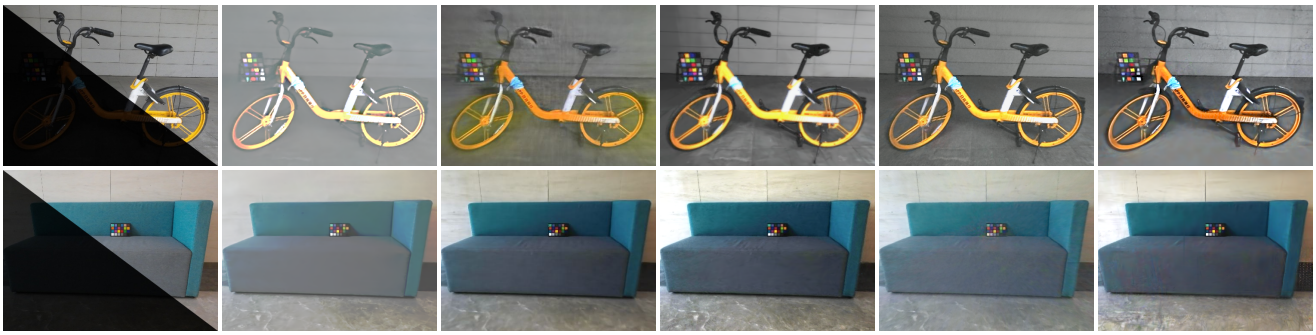
Figure 3. **Qualitative comparison on Gaussian-DK**: held-out views of DORMITORY, KITCHEN, STREET and TREE, top to bottom.

Table 1. **Restoration quality on Gaussian-DK**: per-scene no-reference scores and user-study MOS of the restored render. Red: best; yellow: second best.

Scene	3DGS [17]				LITA-GS [45]				Luminance-GS [6]				Gaussian-DK [43]				VeilGS (ours)			
	NIQE↓	MUSIQ↑	CLIP-IQA↑	MOS↑	NIQE↓	MUSIQ↑	CLIP-IQA↑	MOS↑	NIQE↓	MUSIQ↑	CLIP-IQA↑	MOS↑	NIQE↓	MUSIQ↑	CLIP-IQA↑	MOS↑	NIQE↓	MUSIQ↑	CLIP-IQA↑	MOS↑
ALLEY	2.98	64.73	0.384	2.47	2.99	52.25	0.227	2.65	2.96	65.56	0.274	2.80	3.06	65.56	0.357	3.60	2.21	67.22	0.442	4.65
BICYCLE	4.27	61.82	0.423	2.44	4.42	45.68	0.280	2.58	4.43	56.77	0.230	1.95	4.91	69.29	0.472	3.84	2.94	69.64	0.492	3.95
DORMITORY	4.44	69.03	0.450	2.84	4.10	50.55	0.329	1.95	4.58	69.66	0.372	2.62	4.95	73.56	0.439	3.29	3.01	72.58	0.525	4.45
FENCE	2.93	55.45	0.494	2.31	3.54	45.41	0.376	2.07	3.18	59.58	0.436	2.33	3.39	65.47	0.541	3.53	2.34	64.77	0.555	4.24
FLOWER	2.67	72.20	0.583	3.36	3.49	58.25	0.436	2.71	3.21	70.00	0.504	2.45	3.82	74.31	0.583	2.24	2.48	69.25	0.600	4.13
KITCHEN	3.90	70.30	0.422	3.20	3.94	53.23	0.292	2.47	4.20	69.34	0.389	2.69	5.18	73.86	0.454	2.40	2.58	72.37	0.521	4.78
LIVINGROOM	3.32	65.92	0.403	3.07	3.51	52.79	0.294	2.96	3.51	63.01	0.318	3.31	3.56	69.76	0.368	3.18	2.65	69.14	0.473	4.60
PIANO	4.73	52.72	0.441	1.82	4.65	42.54	0.299	2.24	5.11	49.11	0.278	1.82	5.44	62.18	0.528	4.13	2.64	65.69	0.542	3.51
SNOWMAN	4.19	57.17	0.417	3.16	4.44	46.62	0.371	2.00	4.28	62.79	0.402	1.98	4.66	62.03	0.447	4.45	3.13	64.12	0.464	2.98
STAIRCASE	5.41	58.56	0.384	2.20	3.85	39.37	0.307	2.29	4.21	60.65	0.279	1.80	4.74	69.51	0.526	3.36	2.92	69.55	0.546	3.95
STREET	2.48	64.76	0.411	2.33	3.30	48.08	0.284	2.96	2.78	61.01	0.334	2.11	3.06	72.19	0.541	3.73	1.95	66.71	0.491	4.29
TREE	3.24	64.96	0.523	2.98	3.39	55.35	0.365	2.51	3.44	64.02	0.363	1.87	3.05	68.40	0.585	3.56	2.69	69.35	0.565	4.25
Mean	3.71	63.14	0.445	2.68	3.80	49.18	0.322	2.45	3.83	62.62	0.348	2.31	4.15	68.84	0.487	3.44	2.63	68.37	0.518	4.15

is J rendered from the camera pose alone, with the no-reference metrics NIQE [31], MUSIQ [16] and CLIP-IQA [39]. As there is no normal-light reference, Gaussian-DK and RestorGS [32] evaluate reconstruction with PSNR, SSIM [41] and LPIPS [44] between each held-out dark view and its dark rendering. For VeilGS this reconstruction is target-conditioned: it computes the basis and the coefficients of the veil from the held-out view, with all trained parameters frozen. We therefore use it mainly for the ablations and give the comparison with other methods in the appendix (Table S3). LOM [5] has five scenes (BIKE, BUU, CHAIR, SHRUB, SOFA), each captured with paired low-light and normal-light frames at 500×375. We train on the low-light images and, on the official test split of 3–5 views per scene, compare J with the normal-light frames by PSNR, SSIM and LPIPS, without using the low-light test views. To get human feedback, we also ran a user study in which 55 participants rated the restored renders of VeilGS and four other methods on both benchmarks, blind and on a 1–5 scale, and we report the mean opinion score (details, consent and ethics in Sec. S7).

Implementation Details. We build on the official 3DGS implementation [17] and train each scene for 35,000 iterations on a single NVIDIA H100 GPU, with densification until iteration 15,000 and a densification gradient threshold of 10^{-4}. Training, including periodic evaluation, takes 20–22 minutes per scene on LOM and 36–68 minutes on Gaussian-DK. On LOM, VeilGS averages 20.7 minutes and renders J at 407 FPS, against 5.1 minutes and 397 FPS for Luminance-GS on the same GPU and resolution (Fig. 1 and



Low / Normal light | Luminance-GS [6] | Aleth-NeRF [5] | LITA-GS [45] | I^2-NeRF [23] | VeilGS (ours)

Figure 4. **Qualitative comparison on LOM**: BIKE and SOFA; first column: low light (lower left) and normal light (upper right).

Table 2. **Restoration on LOM**, against the normal-light frames. "A + B": B enhances the renders of A; "B + A": B enhances the training images of A. MOS: user study. Red: best; yellow: second best.

Method	BIKE			BUU			CHAIR			SHRUB			SOFA			Mean			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	MOS↑
NeRF [29]	6.35	0.072	0.623	7.51	0.291	0.448	6.04	0.147	0.594	8.03	0.031	0.680	6.28	0.210	0.568	6.84	0.150	0.583	–
3DGS [17]	6.38	0.071	0.822	7.74	0.292	0.459	6.26	0.146	0.761	8.74	0.039	0.604	6.21	0.201	0.918	7.07	0.150	0.713	–
NeRF + SCI [25]	18.84	0.637	0.565	7.84	0.660	0.562	12.07	0.699	0.584	12.38	0.358	0.587	10.25	0.737	0.626	12.28	0.618	0.585	–
3DGS + SCI	8.93	0.364	0.554	18.33	0.869	0.184	11.51	0.631	0.406	12.63	0.382	0.277	12.98	0.709	0.603	12.88	0.591	0.405	–
NeRF + Zero-DCE [13]	10.16	0.468	0.557	17.81	0.833	0.357	12.44	0.684	0.547	12.58	0.282	0.540	14.43	0.787	0.539	13.48	0.611	0.508	–
Zero-DCE + NeRF	10.39	0.518	0.464	17.90	0.858	0.376	12.58	0.721	0.460	12.32	0.308	0.481	14.45	0.831	0.419	13.53	0.647	0.440	–
Zero-DCE + 3DGS	10.54	0.539	0.401	17.92	0.896	0.179	12.94	0.756	0.303	13.10	0.467	0.229	14.42	0.831	0.356	13.78	0.698	0.294	–
SCI + NeRF	13.44	0.658	0.435	7.76	0.692	0.525	19.77	0.802	0.674	18.16	0.503	0.475	10.08	0.772	0.520	13.84	0.685	0.526	–
3DGS + Zero-DCE	10.56	0.498	0.500	18.86	0.890	0.191	13.24	0.731	0.349	13.26	0.430	0.272	14.23	0.767	0.586	14.03	0.663	0.380	–
SCI + 3DGS	13.67	0.677	0.324	7.95	0.695	0.501	21.77	0.866	0.350	18.67	0.657	0.153	9.99	0.750	0.452	14.41	0.729	0.356	–
Luminance-GS [6]	18.27	0.749	0.411	18.09	0.877	0.193	19.82	0.835	0.367	15.40	0.666	0.241	20.12	0.871	0.259	18.34	0.800	0.294	2.23
Aleth-NeRF [5]	20.46	0.727	0.499	20.22	0.859	0.315	20.93	0.818	0.468	18.24	0.511	0.448	19.52	0.857	0.354	19.87	0.754	0.417	3.27
LITA-GS [45]	22.75	0.819	0.282	20.59	0.897	0.175	22.60	0.873	0.223	19.35	0.659	0.217	20.43	0.895	0.268	21.14	0.829	0.233	3.11
I^2-NeRF [23]	22.87	0.803	0.278	22.35	0.845	0.203	23.82	0.847	0.280	19.44	0.674	0.249	25.89	0.859	0.234	22.87	0.806	0.249	3.38
VeilGS (ours)	23.20	0.815	0.287	22.52	0.893	0.181	23.65	0.884	0.216	19.21	0.662	0.205	26.21	0.887	0.212	22.96	0.828	0.220	3.57

Table S2). The shading MLP f_θ has two hidden layers of width 128. Camera poses and SfM points come from the COLMAP reconstruction released with each benchmark; albedo is initialized to the colours of the SfM points and the shading latents to zero. Sec. S5 gives the training schedule.

4.2. Quantitative and Qualitative Comparison

Baselines. On Gaussian-DK we compare with 3DGS [17], Gaussian-DK [43], LITA-GS [45] and Luminance-GS [6], each trained with its official code on the same training split and evaluated on the same held-out views; the appendix (Table S3) adds the reported results of NeRF-W [28] and RestorGS [32]. On LOM we compare with NeRF [29], 3DGS, their combinations with Zero-DCE [13] and SCI [25], Aleth-NeRF [5], Luminance-GS [6], LITA-GS [45] and I^2-NeRF [23], using the results reported in [6, 23, 45].

Gaussian-DK. This benchmark has no clean ground truth, so we assess restoration with no-reference metrics and the user study. Table 1 scores the restored renders. VeilGS has the best NIQE on all twelve scenes and the best CLIP-IQA on ten, and is second in MUSIQ, where Gaussian-DK [43] and VeilGS are each best on six scenes. Plain 3DGS scores above LITA-GS and Luminance-GS on all three metrics and in MOS, likely because their over-exposure and haze (Fig. 3) cost more than its darkness. In Fig. 3, Gaussian-DK and Luminance-GS leave a grey haze over the scene, most visible on the walls and floor of DORMITORY and KITCHEN, and LITA-GS [45] over-exposes the surfaces near light sources, such as the lantern in TREE and the wall in STREET. VeilGS lifts the dark regions without saturating the lit ones, so the scene keeps its contrast between lit and unlit areas.

LOM. Table 2 compares the restored renders with the normal-light frames. VeilGS is competitive, with modest leads in mean PSNR and LPIPS: 22.96 dB against 22.87 dB for I^2-NeRF [23], and 0.220 against 0.233 for LITA-GS [45]. Its SSIM is second, 0.001 below LITA-GS, and it has the highest PSNR on three of the five scenes. In Fig. 4, Luminance-GS [6] washes out the colours and Aleth-NeRF [5] blurs fine structures such as the wall tiles in BIKE, while VeilGS, LITA-GS and I^2-NeRF recover sharp details close to the normal-light frames.

User study. According to the blind user study, VeilGS has the highest mean opinion score (MOS): 4.15 against 3.44 for Gaussian-DK [43] (Table 1), and 3.57 against 3.38 for I^2-NeRF [23] on LOM (Table 2). Both differences are significant in a two-sided Wilcoxon signed-rank test over participants ($p < 10^{-9}$ and $p = 0.001$; Sec. S7). On PIANO and SNOWMAN, however, participants preferred Gaussian-DK, although VeilGS has better NIQE, MUSIQ and CLIP-IQA there; these metrics appear to miss local colour arte-

Table 3. **How the veil coefficients are obtained**, over the twelve Gaussian-DK scenes. Red: best; yellow: second best.

Variant	Reconstruction			No reference		
	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	MUSIQ↑	PI↓
Learned per-view table	19.62	0.631	0.170	3.06	62.94	3.11
Additive basis, detached	24.74	0.798	0.122	2.90	65.37	2.98
Rank-one basis, detached	25.04	0.816	0.119	2.89	65.83	3.04
Additive basis, through	25.96	0.832	0.103	2.62	67.92	2.80
Rank-one basis, through (ours)	26.39	0.841	0.104	2.63	68.37	2.80

Table 4. **Components of VeilGS**, over the four scenes of Fig. 3, two indoor and two outdoor, sampled to limit training cost. Each row removes one part of the full model. Red: best; yellow: second best.

Variant	Reconstruction			No reference		
	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	MUSIQ↑	PI↓
w/o spatially varying veil	21.79	0.688	0.166	2.80	65.47	2.96
w/o damping	21.91	0.654	0.336	3.71	52.53	3.79
spherical harmonics colour	24.23	0.802	0.137	3.94	66.96	2.92
w/o Retinex terms	25.69	0.813	0.114	2.67	68.60	2.92
w/o depth prior	26.35	0.829	0.109	2.66	69.63	2.84
Full model	26.73	0.836	0.107	2.56	70.25	2.76

facts in its renders.

4.3. Ablation Results

We ablate VeilGS on Gaussian-DK in two parts: how the veil coefficients are obtained (Table 3), and what each component of the model contributes (Table 4). Both tables report the target-conditioned reconstruction of Sec. 4.1 and, besides NIQE and MUSIQ, the perceptual index (PI) [3].

How the coefficients are obtained. Table 3 compares ways of obtaining the veil coefficients, and Fig. 5 shows them on a held-out view. Learning a coefficient table per training view is the weakest on every metric. A held-out view has no entry in the table, so it uses the average of the learned coefficients, and its PSNR, 6.77 dB below our model, partly reflects this; its no-reference scores, which depend only on J, are also the worst. Training through the solve improves on detaching its gradient for both bases and on all six metrics: by 1.35 dB PSNR and 2.5 MUSIQ with the rank-one basis and by 1.22 dB and 2.6 MUSIQ with the additive one, which is consistent with the argument of Sec. 3.5. Rank-one is ahead of additive by 0.43 dB PSNR through the solve and by 0.30 dB when detaching it. Our configuration, rank-one trained through the solve, is best in PSNR, SSIM and MUSIQ, ties on PI, and is within 0.01 of the additive basis in NIQE and LPIPS.

Components. Table 4 removes one part at a time, and every removal worsens all six metrics; Fig. 6 shows the same variants on a held-out view. Solving a single gain per view costs 4.94 dB PSNR, and removing the damping of the solve costs 4.82 dB. Fig. 6(c, d) shows that the undamped veil absorbs scene detail instead of illumination, so the prediction still reproduces the observation while the restored field has dissolved. Replacing the albedo and shading by spherical harmonics costs 2.50 dB PSNR and raises NIQE by 1.38, the largest rise in NIQE, and leaves the shadows dark while over-brightening the lit areas (Fig. 6(e)). Removing the Retinex terms costs 1.04 dB. Without the depth prior, PSNR drops by 0.38 dB and MUSIQ by 0.6.

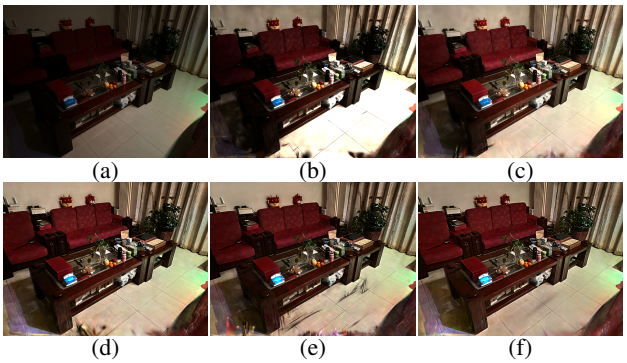


Figure 5. **How the veil coefficients are obtained**, on a held-out view of LIVINGROOM. (a) the dark observation; the restored field with (b) a learned per-view coefficient table, (c) an additive basis detached, (d) a rank-one basis detached, (e) an additive basis through the solve and (f) our rank-one basis through the solve.

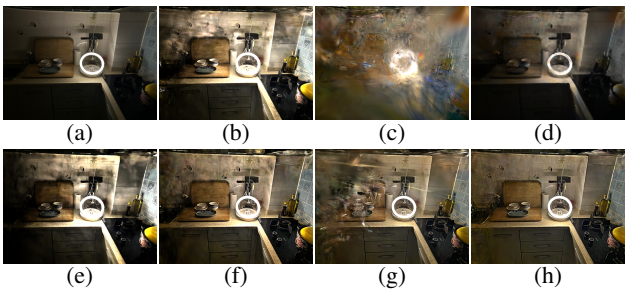


Figure 6. **Components of VeilGS** on a held-out view of KITCHEN. (a) the dark observation; the restored field (b) w/o spatially varying veil, (c) w/o damping, (e) with spherical harmonics colour, (f) w/o Retinex terms, (g) w/o depth prior and (h) of the full model; (d) the prediction $\hat{I}$ of the undamped model, *i.e.* (c) under its veil.

5. Conclusion

We presented VeilGS, which treats low-light degradation as a veil drawn over a well-lit 3D Gaussian scene. Each Gaussian carries an albedo and a shading latent instead of spherical harmonics, and each view is darkened by a spatially varying gain, spanned by a coarse basis read off the observation, followed by a global tone curve. Rather than learning parameters for every view, VeilGS solves the veil from each observation and trains the scene through that solve, which discourages the Gaussians from absorbing what the veil can explain. On Gaussian-DK, its restored renders give the best NIQE and CLIP-IQA and are rated highest in our user study; on LOM, it achieves results competitive with the state of the art. However, the veil is a single achromatic gain, so it cannot remove colour casts from coloured light

sources. Training through the solve can also become unstable on low-texture scenes. Finally, the albedo–shading decomposition is imperfect and can leave a green cast in the albedo, mostly in dark regions.

LLM Usage Statement

We used Claude Opus 5.5 (Anthropic) to assist with running the baseline methods, rendering result videos and editing code, and to polish the writing of the manuscript and its appendix. We also used it to format tables, compute summary statistics from our evaluation and user-study results, and write the script that draws the plot in Fig. 1. The authors reviewed all outputs and take full responsibility for the content of this work.

References

[1] Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning (ICML)*, 2017. 3

[2] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[3] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018. 8

[4] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage Retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12504–12513, 2023. 1, 2

[5] Ziteng Cui, Lin Gu, Xiao Sun, Xianzheng Ma, Yu Qiao, and Tatsuya Harada. Aleth-NeRF: Illumination adaptive NeRF with concealing field assumption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 1, 2, 6, 7, 14, 16

[6] Ziteng Cui, Xuangeng Chu, and Tatsuya Harada. Luminance-GS: Adapting 3D Gaussian splatting to challenging lighting conditions with view-adaptive curve adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 6, 7, 14, 15, 16

[7] Ziteng Cui, Shuhong Liu, Xiaoyu Dong, Xuangeng Chu, Lin Gu, Ming-Hsuan Yang, and Tatsuya Harada. Unifying color and lightness correction with view-adaptive curve adjustment for robust 3D novel view synthesis. *arXiv preprint arXiv:2602.18322*, 2026. 2

[8] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, pages 369–378, 1997. 3

[9] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics*, 36(4), 2017. 3

[10] Gene H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10 (2):413–432, 1973. 2, 3, 5, 12

[11] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016. 3

[12] Michael D. Grossberg and Shree K. Nayar. What is the space of camera response functions? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 2, 3

[13] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1780–1789, 2020. 1, 2, 5, 7

[14] Xiaojie Guo, Yu Li, and Haibin Ling. LIME: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. 2, 3

[15] Yifan Jiang, Xinyu Gong, Ding Liu, Yuheng Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. EnlightenGAN: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021. 2

[16] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5148–5157, 2021. 6

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2, 3, 5, 6, 7, 14, 15

[18] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3D Gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4), 2024. 5, 13

[19] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. WildGaussians: 3D Gaussian splatting in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[20] Edwin H. Land. The Retinex theory of color vision. *Scientific American*, 237(6):108–128, 1977. 2, 3, 5

[21] Ze Li, Feng Zhang, Xiatian Zhu, Meng Zhang, Yanghong Zhou, and P. Y. Mok. Robust low-light scene restoration via illumination transition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6188–6197, 2025. 2

[22] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 5

[23] Shuhong Liu, Lin Gu, Ziteng Cui, Xuangeng Chu, and Tatsuya Harada. I^2-NeRF: Learning neural radiance fields under physically-grounded media interactions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 7, 14, 16

[24] Feifan Lv, Yu Li, and Feng Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *International Journal of Computer Vision*, 129(7): 2175–2193, 2021. 3

[25] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5637–5646, 2022. 1, 2, 7

[26] Radosław Mantiuk, Rafał Mantiuk, Anna Tomaszewska, and Wolfgang Heidrich. Color correction for tone mapping. *Computer Graphics Forum*, 28(2):193–202, 2009. 11

[27] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 4, 11

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7210–7219, 2021. 2, 7, 15

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 7

[30] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16190–16199, 2022. 2

[31] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. 6

[32] Yuanjian Qiao, Mingwen Shao, Lingzhuang Meng, and Kai Xu. RestorGS: Depth-aware Gaussian splatting for efficient 3D scene restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 5, 6, 7, 15

[33] Axel Ruhe and Per-Åke Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22(3):318–337, 1980. 3, 12

[34] Christophe Schlick. Quantization techniques for the visualization of high dynamic range pictures. In *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, pages 7–20, 1994. 11

[35] Hao Sun, Fenggen Yu, Huiyao Xu, Tao Zhang, and Changqing Zou. LL-Gaussian: Low-light scene reconstruction and enhancement via Gaussian splatting for novel view synthesis. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2025. 2

[36] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[37] Chengzhou Tang, Yuqiang Yang, Bing Zeng, Ping Tan, and Shuaicheng Liu. Learning to zoom inside camera imaging pipeline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17552–17561, 2022. 3, 4, 11

[38] Haoyuan Wang, Xiaogang Xu, Ke Xu, and Rynson W. H. Lau. Lighting up NeRF via unsupervised decomposition and enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2

[39] Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the aesthetics and perceptual quality of images. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2555–2563, 2023. 6

[40] Yuehao Wang, Chaoyi Wang, Bingchen Gong, and Tianfan Xue. Bilateral guided radiance field processing. *ACM Transactions on Graphics*, 43(4):1–13, 2024. 2

[41] 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, 2004. 6

[42] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep Retinex decomposition for low-light enhancement. In *British Machine Vision Conference (BMVC)*, 2018. 2

[43] Sheng Ye, Zhen-Hui Dong, Yubin Hu, Yu-Hui Wen, and Yong-Jin Liu. Gaussian in the dark: Real-time view synthesis from inconsistent dark images using Gaussian splatting. *Computer Graphics Forum*, 43(7), 2024. 1, 2, 5, 6, 7, 14, 15

[44] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6

[45] Han Zhou, Wei Dong, and Jun Chen. LITA-GS: Illumination-agnostic novel view synthesis via reference-free 3D Gaussian splatting and physical priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 6, 7, 14, 15, 16

VeilGS: Solving Darkness as a Veil of 2D Basis Maps over Retinex 3DGS

Supplementary Material

This supplementary material gives details and results that complement the main paper. Secs. S1–S4 describe the image formation model, the veil, the solver and the effect of training through the solve. Sec. S5 gives further implementation details, Sec. S6 the evaluation protocol and additional results, and Sec. S7 the user study. Numbers without an "S" refer to the main paper.

S1. Low-Light Image Formation

Tone curve. Eq. (5) applies the camera's tone curve to each colour channel of $x = \alpha J$. In practice, we average this per-channel curve with a curve on the luminance,

$$\hat{I}(\mathbf{p}) = \eta\, x(\mathbf{p})\, Y\big(x(\mathbf{p})\big)^{\frac{1}{\gamma}-1} + (1-\eta)\, x(\mathbf{p})^{\frac{1}{\gamma}}. \quad \text{(S1)}$$

The second term is the per-channel curve of Eq. (5). The first term applies the curve to the luminance $Y(x) = 0.299\, x_R + 0.587\, x_G + 0.114\, x_B$ and keeps the ratios between the channels, following Schlick [34]. A per-channel curve fades colours [26], while a ratio-preserving curve can shift hue. We therefore average the two, with $\eta = 0.5$ in all experiments. The forward model clips α to $[10^{-3}, 1]$, γ to $[1, 4]$ and $\hat{I}$ to $[0, 1]$.

Linearity in the veil. The veil α is one number per pixel, and Y is linear, so $Y(\alpha J) = \alpha\, Y(J)$. Hence $\alpha^{1/\gamma}$ factors out of both terms of Eq. (S1):

$$\hat{I}(\mathbf{p}) = \alpha(\mathbf{p})^{1/\gamma}\Big[\eta\, J\, Y(J)^{\frac{1}{\gamma}-1} + (1-\eta)\, J^{\frac{1}{\gamma}}\Big](\mathbf{p}). \quad \text{(S2)}$$

With $\alpha = b_\alpha e^m$, the log of Eq. (S2) at pixel $\mathbf{p}$ and channel ℓ is exactly

$$\log \hat{I} = \tfrac{1}{\gamma}\log b_\alpha + \log\Big[\eta\, J\, Y(J)^{\frac{1}{\gamma}-1} + (1-\eta) J^{\frac{1}{\gamma}}\Big] + \tfrac{1}{\gamma} m. \quad \text{(S3)}$$

The veil enters only through $\frac{1}{\gamma}m$. For $\eta = 0$, Eq. (S3) becomes Eq. (8) of the main paper,

$$\log \hat{I}(\mathbf{p}, \ell) = \tfrac{1}{\gamma}\big(\log b_\alpha + \log J(\mathbf{p}, \ell) + m(\mathbf{p})\big), \quad \text{(S4)}$$

which is the relation the solver uses. The blend changes only the part that does not depend on the veil. The slope $\frac{1}{\gamma}$ of m, and therefore the form of the fit, stay the same.

S2. The Veil

Basis. The basis network h_ϕ has a stem convolution, four residual blocks of width 32 and an output head. It runs at one-eighth resolution, and its output is enlarged bilinearly, so each map has 64 times fewer degrees of freedom than the image. A $\tanh$ keeps every map in $[-1, 1]$, so the scale clip of Sec. S3 bounds the modulation m. Tang *et al.* [37] combine their two bases by convolution. We multiply them pixel by pixel in Eq. (7), since the veil is a smooth brightness field rather than a blur. The factors $(\mathbf{u}, \mathbf{v}, s)$ are nine numbers, of which seven are free, because $\mathbf{u}$ and $\mathbf{v}$ have unit length.

What each part holds. Fig. S1 shows the intermediate quantities on two training views. The shading is close to grey, as Eq. (18) asks, so the colour sits in the albedo. The shading is below one over most of the image (median 0.29 on FLOWER and 0.45 on LIVINGROOM), so the albedo is brighter and more saturated than J. The veil, solved from the observation as in training, is highest where light falls and lowest in unlit regions. Under this veil, the prediction $\hat{I}$ matches the observation at 26.2 dB (FLOWER) and 29.3 dB (LIVINGROOM) PSNR.

S3. Solver Details

Problem. Substituting Eq. (7) into Eq. (S4) and stacking all pixels and channels gives the fit of Eq. (10) in matrix form,

$$\min_{s,\mathbf{u},\mathbf{v}} \big\| A\ \mathrm{vec}(M) - r \big\|^2 + \mathrm{vec}(M)^\top \Lambda\ \mathrm{vec}(M), \quad M = s\,\mathbf{u}\mathbf{v}^\top. \quad \text{(S5)}$$

Here r stacks the residuals of Eq. (9), and vec stacks the rows of M, so that $\mathrm{vec}(\mathbf{u}\mathbf{v}^\top) = \mathbf{u} \otimes \mathbf{v}$. The design matrix A has the entries

$$A_{(\mathbf{p},\ell),(j,k)} = \tfrac{1}{\gamma}\,\Phi_{jk}(\mathbf{p}) = \tfrac{1}{\gamma}\,K_j(\mathbf{p})\,X_k(\mathbf{p}). \quad \text{(S6)}$$

Each column is one product map $K_j \odot X_k$, scaled by $\frac{1}{\gamma}$ and repeated for the three channels. At the 64×64 solve resolution, A has $3 \cdot 64^2 = 12{,}288$ rows and 16 columns. The stages below use only $A^\top A$ (16×16) and $A^\top r$.

The three rows of a pixel are equal, so the fit sees r only through its channel mean $\bar{r}(\mathbf{p}) = \frac{1}{3}\sum_\ell r(\mathbf{p}, \ell)$. For any value q of the veil term,

$$\sum_\ell \big(q - r(\mathbf{p}, \ell)\big)^2 = 3\big(q - \bar{r}(\mathbf{p})\big)^2 + \sum_\ell \big(r(\mathbf{p}, \ell) - \bar{r}(\mathbf{p})\big)^2, \quad \text{(S7)}$$

and the last term does not depend on the coefficients. The fit therefore has $64^2 = 4{,}096$ distinct equations.

Damping. We use $\Lambda = \lambda\,\mathrm{diag}(A^\top A) + \epsilon \mathbf{I}$ with $\lambda = 4$ and $\epsilon = 10^{-4}$ [27]. The first term damps each coefficient in proportion to its own curvature. This keeps the conditioning independent of the scale of A, which changes with γ during training. The floor ϵ keeps the system invertible if a column of A vanishes. We form $H = A^\top A + \Lambda$ once and use it



Observation I | Albedo | Shading | Restored J | Veil α | Prediction $\hat{I}$ | Inverse depth

Figure S1. **What VeilGS stores and renders**, on training views of FLOWER (top) and LIVINGROOM (bottom). The albedo a_i and the shading s_i are each rasterized like a colour, with the shading divided by its 99th percentile, so their product only approximates J. The veil is solved from the observation and shown from its 1st to its 99th percentile, as in Fig. 1. The prediction applies the tone curve of Eq. (S1) to αJ.

in all three stages, so they all minimize the same damped objective.

Stage 1: joint solve. Without the rank constraint, the solution is $\mathrm{vec}(\tilde{M}) = H^{-1}A^\top r$. We compute it with a QR factorization of H.

Stage 2: rank-one initialization. Eight power iterations on $\tilde{M}^\top\tilde{M}$ give $\mathbf{v}$. Then $\mathbf{u} = \tilde{M}\mathbf{v}/\|\tilde{M}\mathbf{v}\|$ and $s = \mathbf{u}^\top\tilde{M}\mathbf{v}$, with the sign absorbed into $\mathbf{u}$ so that $s \geq 0$. This approximates the rank-one matrix closest to $\tilde{M}$ in the Frobenius norm. It is in general not the minimizer of Eq. (S5), so it only initializes the next stage. Tang *et al*. take a single iteration from a random vector. We start from the all-ones vector instead, which removes a source of randomness from the training graph.

Stage 3: Gauss–Newton refinement. Let $T = [\,\mathbf{u}\otimes\mathbf{I}_d \mid \mathbf{I}_d\otimes\mathbf{v}\,]$ be the Jacobian of $\mathbf{u}\otimes\mathbf{v}$ with respect to $(\mathbf{v},\mathbf{u})$. Each of four steps updates the factors by

$$\begin{bmatrix}\delta\mathbf{v}\\ \delta\mathbf{u}\end{bmatrix} = \big(s\,T^\top HT + \mu\mathbf{I}\big)^{-1}T^\top\big(A^\top r - s\,H(\mathbf{u}\otimes\mathbf{v})\big). \quad \text{(S8)}$$

After each step, $\mathbf{u}$ and $\mathbf{v}$ are normalized and their norms are folded into s. The ridge $\mu = 10^{-2}$ damps the step and is separate from the penalty Λ. It fixes the gauge $(\mathbf{u},\mathbf{v}) \mapsto (\kappa\mathbf{u}, \mathbf{v}/\kappa)$, which otherwise makes the system singular. After the initialization and after every step, s is clipped to $s_{\max} = 4$.

Resolution and cost. The solve uses bilinear 64×64 resizes of J, I and the basis maps. After the resize, I and $b_\alpha J$ are floored at 10^{-6} before the logarithms of Eq. (9). Each training iteration needs one 16×16 and four 8×8 linear solves. The veil of Eq. (7) is then composed from the full-resolution basis.

S4. Training Through the Solve

Gradient through the solve. The solved coefficients depend on the trained parameters through A (the basis and γ) and through r (the render J and the biases b_α, γ). We differentiate each linear solve $x = H^{-1}g$ implicitly. Its backward pass is one more solve with $H^\top$, so we never differentiate through the QR factorization. The power and Gauss–Newton iterations are unrolled. The damping Λ is computed from A inside the graph, so gradients also pass through it.

Variable projection. Stage 1 leaves the residual $r^\star = R_\Lambda r$ of Eq. (11), with

$$R_\Lambda = \mathbf{I} - A\big(A^\top A + \Lambda\big)^{-1}A^\top. \quad \text{(S9)}$$

Without damping, and for A of full column rank, R_0 is an orthogonal projector, and $r^\star$ is exactly the part of r that the basis cannot explain. Solving for the linear coefficients in closed form and optimizing the other parameters on what remains is variable projection [10]. In this picture, a detached solve corresponds to alternating minimization, and a learned coefficient table to joint descent. Ruhe and Wedin [33] analyse both schemes for separable least squares. With damping, R_Λ is no longer a projector. Let σ be a singular value of A after its columns are scaled to unit norm, and w its left singular vector. Neglecting ϵ, $R_\Lambda w = \frac{\lambda}{\sigma^2+\lambda}\,w$. Damping thus shrinks what the basis can explain instead of removing it. The shrinking is strong along directions that the basis represents well ($\sigma^2 \gg \lambda$) and weak along the others.

What reaches the scene. Take a parameter that enters only through r, such as the Gaussians, and a loss $\ell(r^\star)$ with gradient $\nabla\ell$. A detached solve passes $\nabla\ell$ to r unchanged. Training through the solve passes

$$\frac{\partial\ell}{\partial r} = R_\Lambda^\top\nabla\ell = R_\Lambda\nabla\ell. \quad \text{(S10)}$$

The solve thus filters the gradient with the same matrix that filters the residual. Without damping and the rank constraint, training through the fit gives the Gaussians, to first order, only the part of r that the veil cannot explain. With our damping ($\lambda = 4$), the part that the veil can explain is

Table S1. **Damping of the inner solve**, over the four scenes of Table 4. The first row is the *w/o damping* row of that table. Its NIQE and PI average over three scenes, since the undamped TREE render is black. Red: best; yellow: second best.

Variant	Reconstruction			No reference		
	PSNR↑	SSIM↑	LPIPS↓	NIQE↓	MUSIQ↑	PI↓
$\lambda = 0$ (w/o damping)	21.91	0.654	0.336	3.71	52.53	3.79
$\lambda = 8$	26.11	0.824	0.109	2.63	69.23	2.79
$\lambda = 4$ (ours)	26.73	0.836	0.107	2.56	70.25	2.76

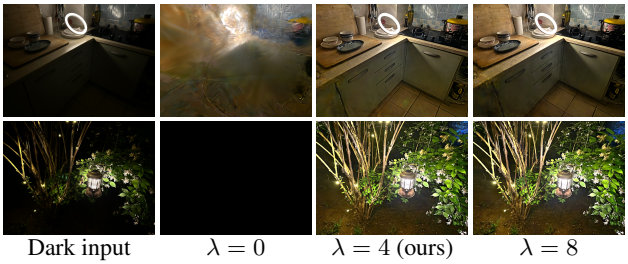


Figure S2. **Damping of the inner solve.** Restored renders of a held-out view of KITCHEN (top) and TREE (bottom). Without damping, the veil absorbs the scene: KITCHEN loses its structure, and TREE collapses to a black render.

shrunk rather than removed. In the same setting, the basis network, which enters through A, learns to cover the part of the observation that the scene does not explain.

Scope of this analysis. This view motivates training through the solve, but it does not fully explain its benefit. If the training loss were the objective of the fit and the fit were exact, the envelope theorem would give the same gradient for a detached solve. For example, without damping, $\ell = \frac{1}{2}\|r^\star\|^2$ has $\nabla\ell = r^\star$, which is already orthogonal to the range of A, so $R_0\nabla\ell = \nabla\ell$. In VeilGS, the two gradients differ for two reasons. First, $\mathcal{L}_{\text{rec}}$ is not the objective of the fit. It compares intensities at full resolution with an ℓ_1 and an SSIM term, after the tone curve of Eq. (S1) and clipping. The fit instead solves a damped least-squares problem in the log domain at 64×64. Second, the fit is not exact. The rank-one stages are nonlinear and stop after four steps. Even at convergence, their residual would be orthogonal only to the seven tangent directions of the rank-one model, not to all 16 columns of A. We therefore show the benefit of training through the solve empirically, in Table 3.

The role of damping. Damping sets how much of the residual the veil may claim. With $\Lambda = 0$, the veil claims every residual in the range of its basis. Early in training, J is still far from the scene, and this range covers most of the observation, so the scene receives little gradient. A large damping pulls the coefficients toward zero. The veil then reduces to its bias, and the spatial variation of the light moves into the scene. Table S1 and Fig. S2 show both effects on the four scenes of the component ablation. Removing the damping costs 4.82 dB PSNR and 17.7 MUSIQ, as much as removing the spatial variation of the veil. The veil then absorbs the scene instead of the light: on KITCHEN the render loses its structure, and on TREE training diverges to a black render. Doubling the damping to $\lambda = 8$ costs 0.62 dB. Our $\lambda = 4$ is best on all six metrics.

S5. Implementation Details

Training schedule. The physical biases start at $b_\alpha = 0.3$ and $\gamma = 2.2$. For the first 200 iterations, only the biases are trained. The basis network stays fixed until iteration 5,000, so that the global level and the geometry settle before the veil gains spatial structure. The veil is optimized with Adam. Its learning rate decays from 10^{-3} to 10^{-5}, and its gradients are clipped to norm 1.

Keeping the biases valid. Gradients through the solve also reach b_α and γ, and these updates can be large. The forward model clips α and γ, and a clip passes no gradient outside its range, so a bias that crosses its bound never returns. For example, if b_α becomes negative, α sits at its floor everywhere, and neither the bias nor the basis network receives gradient. We observed this on one night scene within the first 5,000 iterations. After every optimizer step, we therefore project the biases back into their ranges,

$$b_\alpha \leftarrow \max(b_\alpha, 10^{-3}), \qquad \gamma \leftarrow \text{clip}(\gamma,\, 1{+}10^{-3},\, 4{-}10^{-3}). \tag{S11}$$

This turns the clip into projected gradient descent and keeps the clips of the forward model inactive.

Albedo and shading. The albedo a_i is a free parameter, initialized to the colour of its SfM point. The colour $a_i \odot s_i$ of each Gaussian is clipped at zero before rasterization, so $J \geq 0$. The range term of Eq. (17) keeps the albedo close to $[0, 1]$, with $\rho_{[l,h]}(x) = \text{relu}(l - x) + \text{relu}(x - h)$ applied per channel. The shading prior of Eq. (18) is evaluated with the viewing directions of the current training view, so over training it covers all training views. In Eqs. (17) and (18), the range terms and the squared norm $\|s_i - \bar{s}_i\mathbf{1}\|^2$ are averaged over the three channels, and the neighbour sum over the $k = 8$ neighbours. The weights λ_{alb}, λ_{sh}, μ_a and μ_s refer to this normalization. The albedo smoothness uses the $k = 8$ nearest Gaussians in 3D, with weights $w_{ii'} = \exp(-d_{ii'}/0.1)$ for their distance $d_{ii'}$. The smoothness term of Eq. (17) and both terms of Eq. (18) use 500 randomly chosen Gaussians.

Depth. Following [18], we align each monocular depth map to the SfM points with a scale and an offset. A view is skipped if its scale is more than five times larger or smaller than the median scale. We then min-max normalize the rendered and the monocular inverse depths to $[0, 1]$ in each view. Eq. (19) averages their absolute difference over all pixels.

Efficiency. Table S2 compares the training time and rendering speed of the methods of Fig. 1 on LOM. All of them

Table S2. **Efficiency on LOM**: mean training time per scene and rendering speed, on one NVIDIA H100 at 500×375. Red: best; yellow: second best.

Method	Training (min)↓	FPS↑
Luminance-GS [6]	5.1	397
LITA-GS [45]	18.6	72
I^2-NeRF [23]	32.1	0.57
VeilGS (ours)	20.7	407

were run on one NVIDIA H100 at the LOM resolution of 500×375.

S6. Evaluation Details and Additional Results

Scoring. On both benchmarks, we average each metric over the test views of a scene and then over the scenes. On Gaussian-DK, PSNR, SSIM and LPIPS (AlexNet) use the metric code released with Gaussian-DK [43]. NIQE, MUSIQ, CLIP-IQA and PI use the pyiqa toolbox (v0.1.13). We score the restored render of every method with the same code.

Baselines. We train 3DGS [17], Gaussian-DK [43], LITA-GS [45] and Luminance-GS [6] with their official code on the standard Gaussian-DK split. The restored image is the light-up image for Gaussian-DK, the enhanced render for LITA-GS and Luminance-GS, and the render for 3DGS. For VeilGS, it is J, rendered from the camera pose.

Target-conditioned reconstruction. Gaussian-DK and RestorGS also evaluate how well a method reproduces the held-out dark views. Table S3 gives this comparison per scene. To render a dark view, VeilGS keeps all trained parameters fixed and solves the veil from the dark image of that view, as in training. Gaussian-DK uses the exposure settings of the view, and the other methods we trained render it from the camera pose. To compare all methods fairly, we therefore rely on no-reference metrics and the user study on Gaussian-DK (Table 1 of the main paper). These score the restored renders, and the restored render of VeilGS, J, comes from the camera pose alone. In Table S3, VeilGS has the highest PSNR on every scene and the best mean PSNR, SSIM and LPIPS.

More qualitative results. Fig. S3 shows four more Gaussian-DK scenes, and Fig. S4 the two LOM scenes not shown in the main paper.

S7. User Study

Setup. We ran a blind rating study on a web page. Each question shows five versions of one scene, labelled A to E, and hides the method behind each letter. The assignment of methods to letters and the order of the scenes are shuffled for every participant. The first part covers the twelve Gaussian-DK scenes with looping videos of 3DGS [17], Luminance-GS [6], LITA-GS [45], Gaussian-DK [43] and VeilGS, played in sync. In each video, the camera moves back and forth between two held-out views of the scene (on SNOWMAN, a held-out and a training view). The path is the same for every method, and the videos run at the quarter resolution of the benchmark and 30 frames per second for 8.9 s. The second part covers the five LOM scenes with still images of Luminance-GS, LITA-GS, Aleth-NeRF [5], I^2-NeRF [23] and VeilGS at 500×375. It uses the views of Figs. 4 and S4 and one test view of SHRUB. The Aleth-NeRF and I^2-NeRF images are the renders released by their authors. The other baselines are our runs of their official code.

Task. Participants rate every version for overall visual quality on a five-point scale: 5 Excellent (looks like a clean, well-lit photo or video), 4 Good (minor flaws you have to look for), 3 Fair (noticeable flaws, but the scene is clear), 2 Poor (obvious flaws that get in the way) and 1 Bad (hard to make out the scene). The instructions ask whether the scene is clearly visible, whether its colours and brightness look natural, and whether it is free of haze, blur, noise, floating blobs or flicker. Participants can enlarge any version and switch between the versions of a question. Several versions may receive the same score, and ratings can be changed until submission. Participants are asked to keep their screen at full brightness.

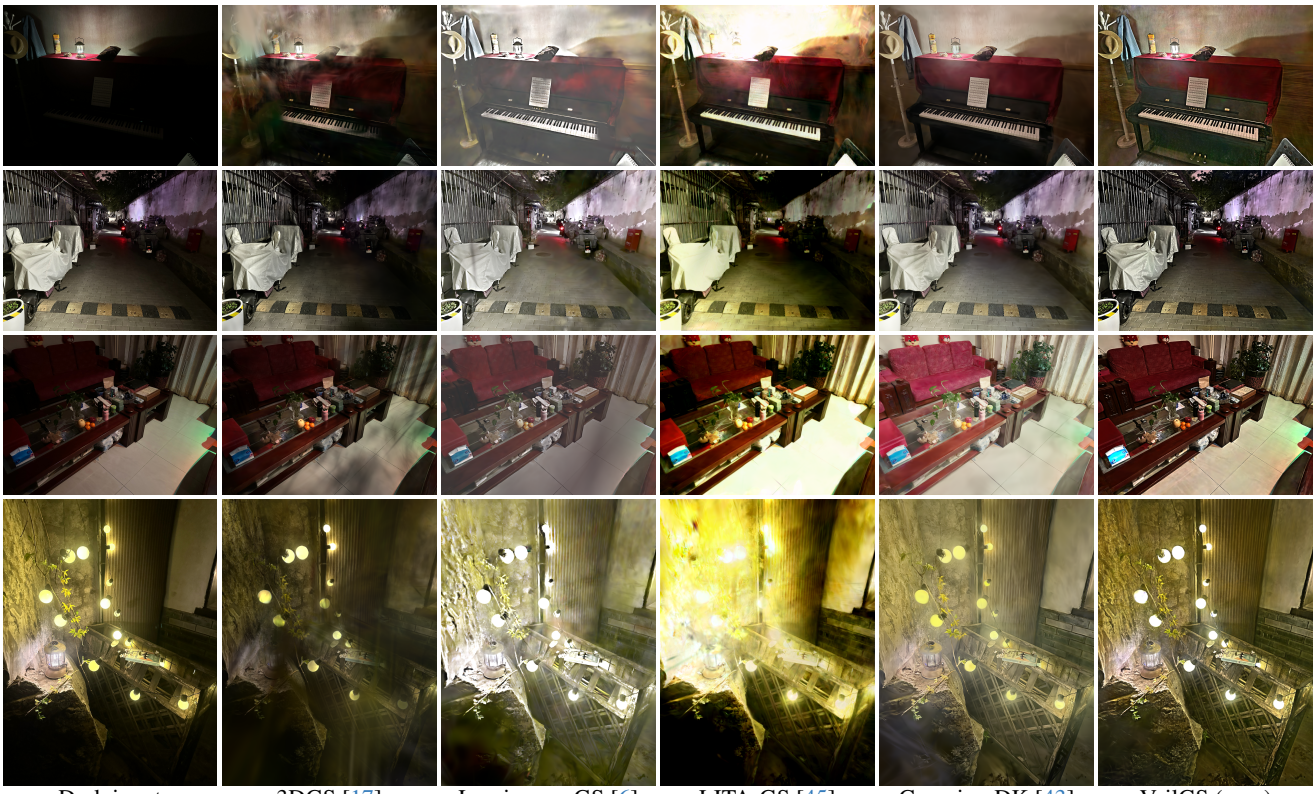
Consent and ethics. Before starting, participants agree that their anonymous ratings may be used for research. The study recorded no identifying information. No ethics review was carried out, as we judged the study to be of low risk.

Analysis. In total, 55 participants finished the study and rated every version. This gives 4,675 ratings, given in a median of 8.4 minutes (interquartile range 5.8–14.2). If a version was rated more than once, the last rating counts. The ratings of one participant are correlated, so we compute every statistic over participants. The mean opinion score (MOS) of a method is the mean, over participants, of the average score that each participant gave it. Its 95% confidence interval is $t\,\sigma/\sqrt{n}$ over the $n = 55$ participant averages. We compare VeilGS with each baseline by a two-sided Wilcoxon signed-rank test on the same participant averages.

Results. Tables S4 and S5 give the MOS on each benchmark, and Table 1 of the main paper the MOS per scene on Gaussian-DK. Fig. S5 shows how the scores of the participants are distributed. On the twelve Gaussian-DK scenes, VeilGS has a MOS of 4.15, against 3.44 for Gaussian-DK. Of the 55 participants, 51 rate it higher than Gaussian-DK ($p < 10^{-9}$). It has the highest score on ten of the twelve scenes; on PIANO and SNOWMAN, Gaussian-DK scores higher. On LOM, VeilGS has the highest MOS, 3.57, against 3.38 for I^2-NeRF ($p = 0.001$) and 3.27 for Aleth-NeRF. It is best on two of the five scenes, Aleth-NeRF on BUU and SOFA, and I^2-NeRF on SHRUB. On both bench-

Table S3. **Target-conditioned reconstruction on Gaussian-DK**, per scene: each held-out dark view against its dark rendering. All methods use the standard test views of Gaussian-DK. †: results reported in [43] (NeRF-W) and [32] (RestorGS). Neither paper reports SNOWMAN, so the table covers the other eleven scenes; the last row averages all twelve, SNOWMAN included, for the methods we ran. Red: best; yellow: second best.

Scene	3DGS [17]			Luminance-GS [6]			LITA-GS [45]			NeRF-W [28]†			Gaussian-DK [43]			RestorGS [32]†			VeilGS (ours)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ALLEY	17.19	0.637	0.191	16.09	0.612	0.226	17.04	0.617	0.210	21.59	0.646	0.331	21.72	0.774	0.141	22.18	0.781	0.138	23.75	0.807	0.106
BICYCLE	18.69	0.601	0.226	19.91	0.598	0.222	20.64	0.617	0.189	23.56	0.717	0.238	24.90	0.806	0.129	24.93	0.811	0.127	26.52	0.860	0.114
DORMITORY	19.71	0.655	0.164	21.38	0.672	0.179	21.22	0.666	0.158	24.51	0.820	0.149	25.37	0.854	0.088	25.95	0.859	0.088	27.63	0.891	0.075
FENCE	19.01	0.583	0.225	20.26	0.609	0.227	20.67	0.631	0.229	23.97	0.724	0.214	25.13	0.801	0.129	25.05	0.803	0.128	26.38	0.816	0.116
FLOWER	17.68	0.586	0.182	17.89	0.577	0.195	19.34	0.622	0.187	21.66	0.662	0.223	21.70	0.735	0.133	21.97	0.742	0.127	22.95	0.760	0.117
KITCHEN	18.60	0.691	0.183	17.77	0.702	0.220	19.48	0.730	0.179	24.84	0.812	0.205	25.04	0.839	0.118	25.86	0.854	0.099	28.60	0.872	0.094
LIVINGROOM	20.52	0.734	0.159	19.35	0.706	0.196	21.26	0.742	0.189	22.43	0.749	0.243	24.53	0.825	0.126	24.73	0.830	0.088	26.10	0.842	0.103
PIANO	19.25	0.591	0.209	20.08	0.601	0.232	21.00	0.617	0.214	25.44	0.803	0.157	27.13	0.830	0.106	27.37	0.837	0.103	29.97	0.889	0.079
STAIRCASE	16.16	0.480	0.279	18.91	0.536	0.189	20.51	0.583	0.176	23.96	0.809	0.115	26.73	0.887	0.067	27.02	0.882	0.067	27.41	0.908	0.073
STREET	16.74	0.539	0.266	17.32	0.553	0.296	21.51	0.661	0.221	22.85	0.638	0.293	26.06	0.786	0.144	26.44	0.793	0.141	26.45	0.782	0.129
TREE	18.78	0.573	0.210	18.69	0.552	0.231	19.55	0.559	0.223	21.66	0.649	0.260	23.60	0.778	0.132	23.89	0.793	0.123	24.27	0.799	0.131
Mean (11 scenes)	18.39	0.606	0.209	18.88	0.611	0.219	20.20	0.641	0.198	23.32	0.730	0.221	24.72	0.810	0.119	25.04	0.817	0.112	26.36	0.839	0.103
Mean (12 scenes)	18.56	0.606	0.208	19.13	0.614	0.217	20.21	0.637	0.197	–	–	–	24.88	0.813	0.118	–	–	–	26.39	0.841	0.104



Dark input | 3DGS [17] | Luminance-GS [6] | LITA-GS [45] | Gaussian-DK [43] | VeilGS (ours)

Figure S3. **More qualitative results on Gaussian-DK.** Restored renders of held-out views of PIANO, ALLEY, LIVINGROOM and FENCE (top to bottom). Among the scenes not shown in Figs. 1 and 3, these four have the largest PSNR lead of VeilGS in Table S3. Each row shows the held-out view with the largest lead over the baselines we trained.

marks, the per-participant difference favours VeilGS against every baseline ($p \leq 0.002$).

Table S4. **Mean opinion score on Gaussian-DK** (1–5), with the 95% confidence interval over participants. Videos of all twelve scenes; Table 1 gives the scores per scene. Red: best; yellow: second best.

Method	MOS ± 95% CI
3DGS [17]	2.68 ± 0.27
Luminance-GS [6]	2.31 ± 0.20
LITA-GS [45]	2.45 ± 0.22
Gaussian-DK [43]	3.44 ± 0.17
VeilGS (ours)	4.15 ± 0.12



Figure S4. **More qualitative results on LOM.** BUU (top) and CHAIR (bottom), the two scenes of Table 2 not shown in Fig. 4. The first column shows the low-light input (lower left) and the normal-light frame (upper right).

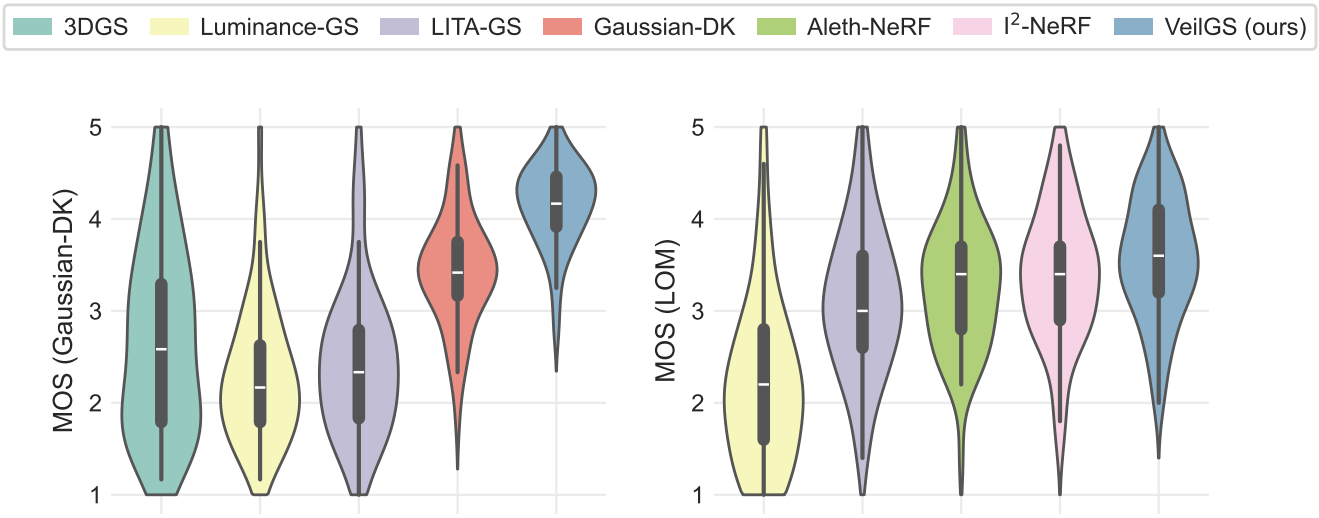


Figure S5. **Scores of the user study.** Each violin shows, for one method, the average score that each of the 55 participants gave it over the scenes of a benchmark. The box marks the quartiles and the white line the median. The means and their 95% confidence intervals are in Tables S4 and S5.

Table S5. **Mean opinion score on LOM** (1–5), per scene and on average, with the 95% confidence interval of the average over participants. Red: best; yellow: second best.

Method	BIKE	BUU	CHAIR	SHRUB	SOFA	Mean ± 95% CI
Luminance-GS [6]	1.82	2.76	2.24	1.91	2.42	2.23 ± 0.25
LITA-GS [45]	2.71	2.95	3.55	2.35	4.00	3.11 ± 0.22
Aleth-NeRF [5]	2.22	3.64	2.95	3.42	4.13	3.27 ± 0.19
I²-NeRF [23]	3.82	3.60	2.84	4.00	2.65	3.38 ± 0.20
VeilGS (ours)	4.05	3.42	3.85	2.78	3.75	3.57 ± 0.18