VIEW-STRUCTURED CONFORMAL PREDICTION FOR 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) renders novel views in real time, but an uncertainty heatmap only ranks pixels. It does not say how wide an interval must be for a view to meet a coverage target. We treat novel-view synthesis as structured regression and ask that, with probability at least $1-\alpha$, RGB prediction boxes cover at least a $1-\beta$ fraction of pixels in a new view. We propose View-Structured Conformal Prediction (VSCP). It splits the pre-calibration scale into a spatial shape from the renderer and a transferable view-difficulty factor, which predicts the smallest view-wise multiplier that shape needs. A held-out quantile over views (View-CP) then gives finite-sample validity even when transferring to new scenes. The same factorization makes the analysis exact. A conformity score is the ratio of oracle to predicted view difficulty, and excess width separates into a test-side and a calibration-side term. Across 13 real scenes, pixel-pooled calibration reaches 89.9% marginal pixel coverage but only 61.4% view-event coverage at a 90% target, while View-CP reaches 91.7–92.0%. At matched coverage VSCP cuts width by 22.1% against a constant scale, and matches a ten-model ensemble's 21.0% reduction using only one model per scene and four rather than ten rasterization passes per query. VSCP also improves on the closest single-model baseline, the 3DGS-U field, by 4.7 points ($p = 0.0225$). The view predictor transfers from bounded source families to all nine unbounded Mip-NeRF 360 scenes. There VSCP saves 23.5% width on average and beats the constant scale on all nine scenes. It also keeps a 20.6% saving under a different densification backbone and runs at 216–280 FPS on an RTX 4090.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) makes high-resolution novel-view synthesis practical in real time (Kerbl et al., 2023; Ren et al., 2026). Yet a rendering can look plausible exactly where it is wrong, and recent methods therefore render posterior variance, visibility, information, or residual-derived uncertainty (Goli et al., 2024; Wu et al., 2026; Galappaththige et al., 2026; Xue et al., 2026). Such fields are evaluated mainly by the area under the sparsification error (AUSE) and rank correlation, which read only the ordering of pixels. This ordering never says how wide an interval must be or how often it contains the unknown image.

The statistical unit matters just as much. A rendered image holds millions of dependent pixels, but a user looks at one view. Suppose two methods both miss 10% of all pixels. One misses 10% in every view, and the other misses 20% in half the views and none in the rest. Pooled coverage is the same, yet only the first gives every view 90% within-view coverage. We therefore ask a view-level question. With probability at least $1-\alpha$, does a new rendering cover at least a $1-\beta$ fraction of its pixels?

We propose **V**iew-**S**tructured **C**onformal **P**rediction (**VSCP**). Conformal prediction can certify this event once each held-out view has a score. Validity alone does not make the boxes useful. A constant scale is valid but far too wide in easy views and in well-observed regions. The real problem is efficiency under the spatial and angular structure of rendering. We address it with the scale

$$s_{vp} = a(z_v)\,\widetilde{b}_{vp}, \tag{1}$$

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.

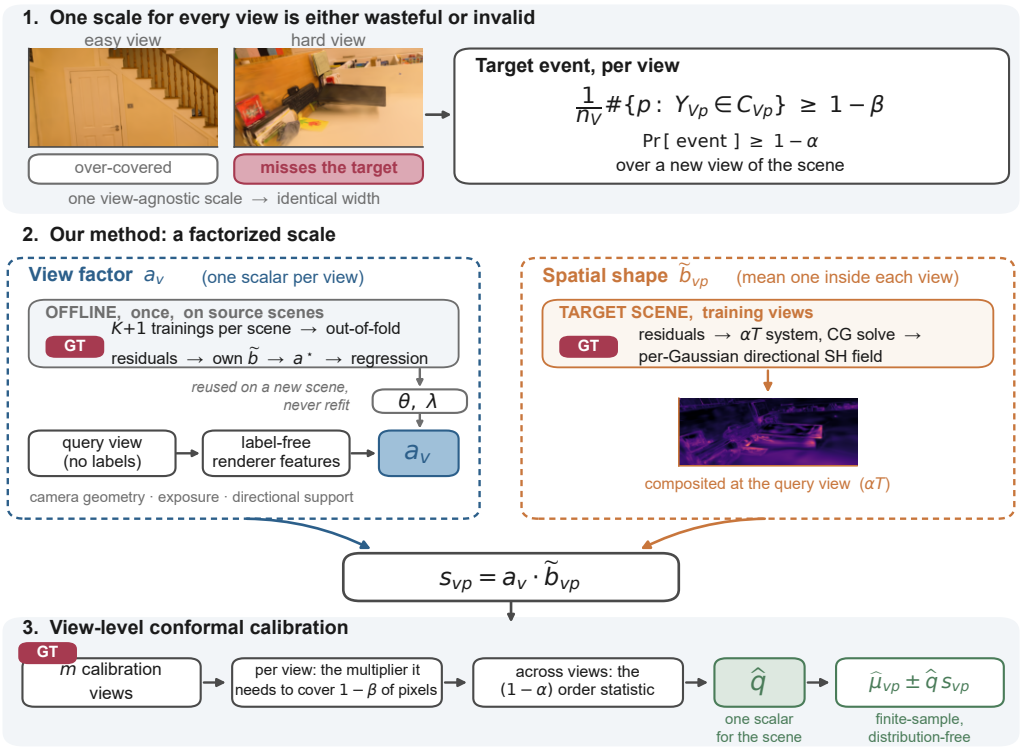


Figure 1: Overview of VSCP. A single scale can waste width on an easy view and miss the within-view target on a hard view. We factor the scale into a transferable view factor a_v and a target-scene spatial shape $\tilde{b}_{vp}$. Only the held-out calibration views enter the nested order statistics. They produce one multiplier $\hat{q}$ and finite-sample valid RGB boxes for a new view.

where z_v is a label-free descriptor of view v, $\tilde{b}$ is the relative spatial variation inside that view, and a is its overall difficulty. We fit $\tilde{b}$ from training residuals on view-dependent Gaussian primitives. We learn a on source scenes from label-free camera, rendering, exposure, and directional-support features.

The regression target for a is not an arbitrary error summary. For a fixed spatial shape $\tilde{b}$, we derive the smallest multiplier

$$a_v^\star(\tilde{b}) = Q_{1-\beta,p}\left(r_{vp}/\tilde{b}_{vp}\right) \tag{2}$$

that lets the view reach its required within-view coverage. This turns view-difficulty prediction into a direct efficiency problem. Since a enters only the width, a view factor learned on other scenes cannot break validity, so it transfers to a new scene or dataset family without refitting. Figure 1 summarizes our method.

This form also makes the analysis exact. A view's conformity score is exactly $a_v^\star/a_v$, which is the ratio of oracle to used view difficulty. The width ratio to the oracle then has a closed form. It equals the test view's own error over a low order statistic of the calibration errors. One underestimated calibration view therefore widens every interval, and average regression accuracy is the wrong efficiency diagnostic.

We evaluate VSCP on 13 real scenes. At a 90% target, pixel-pooled calibration covers 89.9% of all pixels. However, only 61.4% of views have at least 90% of their pixels covered. View-CP raises this rate to 91.7–92.0%. At matched coverage, VSCP reduces width by 22.1% compared with a constant scale. A ten-model ensemble gives a similar reduction of 21.0%, but VSCP needs only one model per scene. VSCP also beats the closest single-model baseline, 3DGS-U, by 4.7 points ($p = 0.0225$). To test transfer, we learn the view factor only on bounded scenes and apply it to all nine unbounded

Mip-NeRF 360 scenes. VSCP still beats the constant scale on every one of them, with a mean saving of 23.5%. Without refitting, VSCP keeps a 20.6% saving under a different densification backbone. The full query runs at 216–280 FPS on an RTX 4090.

Our contributions are as follows. (1) **We define reliable 3DGS rendering as a view-level coverage event and give a finite-sample guarantee for it.** View-CP calibrates on held-out views instead of pooled pixels, and the guarantee holds for any scale fixed in advance. (2) **We split the scale into a spatial shape from the renderer and a difficulty factor for each view.** For this split, we derive the oracle regression target for the view factor and prove an exact decomposition of the excess width. (3) **We learn the view factor once on source scenes and reuse it on new scenes without refitting.** It uses only label-free features, and per-view ranking metrics cannot detect it even though it strongly affects the interval width.

2 Related Work

Uncertainty for neural radiance fields. Early radiance-field methods model uncertainty in several ways. NeRF-W predicts a per-pixel uncertainty to down-weight transient objects in photo collections (Martin-Brualla et al., 2021). S-NeRF learns a probability distribution over radiance fields (Shen et al., 2021). Deep ensembles are a simple and strong baseline for predictive uncertainty (Lakshminarayanan et al., 2017), and density-aware NeRF ensembles add a term for regions that the training views did not observe (Sünderhauf et al., 2023). Bayes' Rays estimates uncertainty for a pretrained field with a post-hoc Laplace approximation (Goli et al., 2024), and a meta-calibrator calibrates NeRF uncertainty on unseen scenes without held-out images (Amini-Naieni et al., 2024). ActiveNeRF and FisherRF use uncertainty or Fisher information to choose new training views (Pan et al., 2022; Jiang et al., 2024).

Uncertainty for 3DGS. Variational and Bayesian 3DGS models estimate parameter and predictive uncertainty (Li & Cheung, 2024; Wu et al., 2026). Uncertainty also guides pruning, dynamic reconstruction, and RGB-D SLAM (Hanson et al., 2025; Guo et al., 2026; Tran & Kosecka, 2026). 3DGS-U fits a per-Gaussian photometric uncertainty from training residuals (Galappaththige et al., 2026), while Han & Dumery (2025) store view-dependent uncertainty as a spherical-harmonic feature of each Gaussian. GAVIS builds an anisotropic visibility field for active mapping (Xue et al., 2026). Rendering-aware Bayesian 3DGS reports posterior intervals together with their calibration error (Jia et al., 2026). Such uncertainty maps are often evaluated by sparsification, which removes the most uncertain pixels first and tracks the error of the rest (Ilg et al., 2018). The renderer is also view dependent. 3DGS models color with spherical harmonics of the view direction (Kerbl et al., 2023), and 6DGS couples spatial and angular coordinates more explicitly (Gao et al., 2025).

Conformal prediction. Conformal prediction gives distribution-free coverage in finite samples when the data are exchangeable (Vovk et al., 2005). Split conformal prediction calibrates a fixed model on held-out data (Lei et al., 2018). Normalized scores divide each residual by an estimate of its scale, so the interval width follows the input (Papadopoulos et al., 2008; Lei et al., 2018). Conformalized quantile regression learns intervals whose width depends on the input (Romano et al., 2019). Other methods learn input-adaptive thresholds or weights, for example by image-specific threshold regression (Luo et al., 2026), approximate conditional predictive distributions (Plassier et al., 2025), or latent-domain weighting (Kong et al., 2026). Recent analyses bound the efficiency of conformalized regression in finite samples (Yao et al., 2026). Risk-controlling prediction sets bound the expected loss with high probability (Bates et al., 2021), and conformal risk control extends split conformal prediction to monotone losses (Angelopoulos et al., 2024).

Structured and image-valued conformal prediction. Image-to-image regression calibrates pixel-wise intervals so that the expected fraction of uncovered pixels stays below a chosen level (Angelopoulos et al., 2022). For diffusion models, a convex program finds short pixel-wise intervals under the same kind of risk constraint (Teneggi et al., 2023). Principal uncertainty quantification calibrates regions along the principal components of the posterior, which captures the spatial correlation between pixels (Belhasin et al., 2024). Kandinsky calibration groups similar pixels to calibrate segmentation models (Brunekreef et al., 2024). Conformal structured prediction uses task structure

to handle large output sets (Zhang et al., 2025). Functional conformal prediction treats a whole curve as one response and gives prediction bands for new curves (Lei et al., 2015).

3 Problem Setup: View-Structured Coverage

3.1 3DGS as functional regression

Fix a scene and a trained 3DGS renderer. A view v contains camera extrinsics, intrinsics, and the associated image-formation conditions. The renderer maps this view to an RGB image $\widehat{\mu}_v = \{\widehat{\mu}_{vp} \in [0,1]^3 : p \in \mathcal{P}_v\}$, while the response is the unknown image $Y_v = \{Y_{vp}\}_{p\in\mathcal{P}_v}$. This is a functional regression problem. One covariate v produces a structured, image-valued response. Pixels may be arbitrarily dependent. The statistical observations used by conformal calibration are views.

Let $w_{vpi} = \alpha_{vpi}T_{vpi}$ be the alpha-transmittance contribution of Gaussian i to pixel p. Suppressing background terms, the point rendering has the form

$$\widehat{\mu}_{vp} = \sum_i w_{vpi}\, c_{vi}, \tag{3}$$

where c_{vi} is the spherical-harmonic color of Gaussian i evaluated for camera v.

3.2 RGB boxes and the view event

We use the scalar RGB residual

$$r_{vp} = \|Y_{vp} - \widehat{\mu}_{vp}\|_\infty. \tag{4}$$

For a positive scale s_{vp} and multiplier q, the output set is the axis-aligned RGB box

$$C_{vp}(q,s) = \prod_{c=1}^{3} [\widehat{\mu}_{vpc} - qs_{vp},\ \widehat{\mu}_{vpc} + qs_{vp}]. \tag{5}$$

The sup-norm makes $Y_{vp} \in C_{vp}$ exactly equivalent to $r_{vp} \le qs_{vp}$. We evaluate the unclipped half-width qs_{vp}. Intersecting the box with $[0,1]^3$ preserves containment but changes the width functional.

For user parameters $\alpha, \beta \in (0,1)$, define

$$E_v(q,s) = \left\{ \frac{1}{n_v} \sum_{p\in\mathcal{P}_v} \mathbb{I}\{Y_{vp} \in C_{vp}(q,s)\} \ge 1-\beta \right\}, \qquad n_v = |\mathcal{P}_v|. \tag{6}$$

Our target is

$$\mathbb{P}\{E_V(\widehat{q},s)\} \ge 1-\alpha \tag{7}$$

for a new view V exchangeable with the calibration views. Thus $1-\beta$ is the required fraction of covered pixels inside a successful view, while $1-\alpha$ is the probability that a new view is successful.

4 View-Structured Conformal Prediction

VSCP combines a factorized scale with view-level conformal calibration (View-CP). We apply the same calibration layer to every baseline.

4.1 Data roles and factorized scale

The method uses four disjoint roles. Scene training views fit the target-scene 3DGS and its residual field. Source scenes learn a transferable view predictor and select fixed shrinkage parameters. Target calibration images enter only the final conformal quantile, and target test images only the evaluation.

The scale before conformal calibration is

$$s_{vp} = a_v\, \widetilde{b}_{vp}, \tag{8}$$

$$a_v = (1-\lambda_a) + \lambda_a \widehat{a}(z_v), \qquad \widetilde{b}_{vp} = (1-\lambda_b) + \lambda_b \frac{b_{vp}}{\bar{b}_v}, \tag{9}$$

where $\widehat{a}$ is the view predictor of Section 4.3, $\overline{b}_v = n_v^{-1} \sum_p b_{vp}$, and $\lambda_a, \lambda_b \in [0, 1]$ are fixed using source data. We normalize the positive raw view predictions to have mean one over the unlabeled query batch, and Equation (9) gives $n_v^{-1} \sum_p \widetilde{b}_{vp} = 1$, resolving the multiplicative non-identifiability between a and b (Lemma 1 in Appendix B).

4.2 Renderer-derived spatial shape

Let $e_{tp} = \|Y_{tp} - \widehat{\mu}_{tp}\|_\infty$ be a residual on a 3DGS training view t. Following the residual-attribution construction of 3DGS-U (Galappaththige et al., 2026), each Gaussian carries a spherical-harmonic residual field $g_i(d) = C_i^\top \phi(d)$. For view t, define the linear rendering operator

$$(\mathcal{A}_t C)_p = \sum_i w_{tpi}\, C_i^\top \phi(d_{ti}), \tag{10}$$

where d_{ti} is the unit vector from Gaussian i to camera t. We fit the coefficients on the target-scene training views $\mathcal{D}^{\mathrm{GS}}$ by solving

$$\widehat{C} = \arg\min_C \sum_{t \in \mathcal{D}^{\mathrm{GS}}} \|\mathcal{A}_t C - e_t\|_2^2 + \tau \|C - C_{\mathrm{prior}}\|_2^2, \tag{11}$$

where $\tau > 0$ is a ridge penalty and C_{prior} represents a constant high-uncertainty field. We use conjugate gradients with exact forward/adjoint rasterization operators. At a query view, $b_{vp} = \max\{(\mathcal{A}_v \widehat{C})_p, \epsilon_0\}$ is rendered like an additional color channel.

4.3 Angularly aware, risk-derived view factor

The view descriptor z_v contains only quantities available without query labels. Camera features measure nearest-neighbor distance, angular distance, and local density relative to training cameras. Render features summarize accumulated alpha, depth, and RGB gradient. The angular part of the descriptor comes from per-Gaussian observation support

$$E_i = \sum_{t,p} w_{tpi}, \qquad D_i = \sum_t \Big(\sum_p w_{tpi} \Big) d_{ti}. \tag{12}$$

For query direction d_{vi}, we render summaries of exposure $\log(1 + E_i)$, angular extrapolation $\arccos\big(d_{vi}^\top D_i / \|D_i\|\big)$, and directional spread $1 - \|D_i\|/E_i$, summarizing each map by its mean and lower quantiles. With the camera and render features and two view-level log-means of the spatial field, this gives a 24-dimensional descriptor. It explicitly models whether visible Gaussians were observed, from which directions, and with what directional concentration.

For a fixed $\widetilde{b}$, the regression target is

$$a_v^\star(\widetilde{b}) = Q_{1-\beta,p}\Big(r_{vp} / \widetilde{b}_{vp} \Big), \tag{13}$$

where $Q_{1-\beta,p}$ takes the $\lceil n_v(1-\beta) \rceil$-th smallest value over the pixels $p \in \mathcal{P}_v$. It needs labels, so we compute it only on source scenes. Each source scene uses its own $\widetilde{b}$ (Section 4.2), and its r_{vp} comes from $K = 10$ fold models that did not see view v. We fit a ridge-regularized log-linear predictor

$$\log \widehat{a}(z_v) = \theta^\top \frac{z_v - \mu_z}{\sigma_z} + \theta_0, \tag{14}$$

standardizing features and centering targets within each source scene. The ridge penalty is selected by leaving out source scenes. This small model keeps the angular signals auditable and suits the number of source views. A target scene needs no fold models and does not update θ.

4.4 View-level conformal calibration

For any fixed positive scale s, define the score of calibration view v as

$$R_v(s) = Q_{1-\beta,p}(r_{vp} / s_{vp})\,. \tag{15}$$

Given m calibration views, set

$$k_\alpha = \lceil (m+1)(1-\alpha) \rceil, \qquad \widehat{q} = \begin{cases} Q_{k_\alpha}(\{R_1, \ldots, R_m\}), & k_\alpha \le m, \\ +\infty, & k_\alpha > m, \end{cases} \tag{16}$$

where Q_{k_α} takes the k_α-th smallest of the m calibration scores. For a new view V we render the center $\widehat{\mu}_V$, form its label-free scale, and return

$$C_{Vp} = \prod_{c=1}^{3} \left[\widehat{\mu}_{Vpc} - \widehat{q}\, a_V \widetilde{b}_{Vp},\ \widehat{\mu}_{Vpc} + \widehat{q}\, a_V \widetilde{b}_{Vp} \right]. \tag{17}$$

The half-width varies with the pixel through $\widetilde{b}_{Vp}$ and with the view through a_V, while $\widehat{q}$ is a single scalar for the scene. Algorithm 1 (Appendix C) collects the whole procedure of VSCP.

5 Validity and Efficiency

We condition on the trained renderer, the source scenes, and every choice used to build the scale. The exchangeable units are the labeled calibration and test views. We build the scales from the unlabeled covariates of the whole query batch, before any calibration/test split. This construction is permutation equivariant, so the scores stay exchangeable (Proposition 4, Appendix B).

Everything below rests on one simple fact. Because $s_{vp} > 0$, the within-view coverage fraction is the empirical distribution function of $\{r_{vp}/s_{vp}\}$. The view event in Equation (6) therefore holds at multiplier q if and only if $q \ge R_v(s)$ (Lemma 2, Appendix B). This is also why both quantiles must be discrete order statistics. An interpolated quantile would break the equivalence.

Theorem 1 (Finite-sample view-event validity)**.** *Suppose s is fixed before target calibration labels are observed and the m calibration scores together with the new-view score are exchangeable. Then the boxes calibrated by Equation* (16) *satisfy*

$$\mathbb{P}\left\{ \frac{1}{n_V} \sum_{p} \mathbb{I}\{Y_{Vp} \in C_{Vp}(\widehat{q}, s)\} \ge 1 - \beta \right\} \ge 1 - \alpha. \tag{18}$$

If scores have no ties and $k_\alpha \le m$, the probability is at most $1 - \alpha + 1/(m+1)$.

The proof is the standard split-conformal rank argument after Lemma 2 (Appendix B). View-CP is also conformal risk control (Angelopoulos et al., 2024) with the binary loss $1 - \mathbb{I}\{E_V(q, s)\}$.

Corollary 1 (Transfer)**.** *Let $\mathcal{F}$ collect every object fixed before target calibration labels are observed, including the trained renderer, source data, and fitted scale construction. If the target calibration and test views are exchangeable given $\mathcal{F}$, then Equation* (18) *holds for every $\mathcal{F}$. Mismatch between $\mathcal{F}$ and the target can change the interval width, but not the coverage guarantee.*

We now give the two results that govern width. Because $\widetilde{b}$ has within-view mean one, the mean half-width of view i is $W_i = \widehat{q}\, n_i^{-1} \sum_p s_{ip} = \widehat{q}\, a_i$.

Proposition 1 (Oracle view factor and score identity)**.** *Fix a positive $\widetilde{b}$. For every view v and every scalar $c > 0$,*

$$R_v(c\widetilde{b}) = \frac{a_v^\star(\widetilde{b})}{c}, \tag{19}$$

and $a_v^\star(\widetilde{b})$ is the smallest scalar in the family $\{a\widetilde{b} : a > 0\}$ that attains the within-view event at $q = 1$.

A conformity score is therefore a ratio of oracle to used view difficulty. This makes $\log a_v^\star$ the natural regression target. Appendix A adds its identification under a separable residual model (Proposition 3), and a uniform error bound.

Proposition 2 (Exact excess-width decomposition)**.** *Assume $k_\alpha \le m$. Let $\delta_v = \log(a_v/a_v^\star)$ be the log error of the deployed post-shrinkage factor of Equation* (9)*, and let $\delta_{(1)} \le \cdots \le \delta_{(m)}$ be the ordered calibration errors. Set $j_\alpha = \lfloor (m+1)\alpha \rfloor$. For every test view i,*

$$\widehat{q} = e^{-\delta_{(j_\alpha)}}, \qquad \frac{W_i(a)}{W_i(a^\star)} = e^{\delta_i - \delta_{(j_\alpha)}}. \tag{20}$$

Consequently, with $\overline{(\cdot)}$ the average over the test views of a fixed split,

$$\frac{\overline{W}(a)}{\overline{W}(a^\star)} = \underbrace{\frac{\overline{a^\star e^{\delta}}}{\overline{a^\star}}}_{\text{test-side dispersion}} \underbrace{e^{-\delta_{(j_\alpha)}}}_{\text{calibration-side lower tail}} . \tag{21}$$

A common shift of every δ_v changes only $\widehat{q}$, not the width ratio, so global multiplicative bias cancels exactly (Lemma 1). At the smallest usable calibration size for $\alpha = 0.1$ we have $m = 9$ and $j_\alpha = 1$, so one underestimated calibration view sets the width inflation. Appendix A also gives an exact realizable-gain identity, and Appendix D audits all three identities numerically.

6 Experiments

6.1 Setup, protocols, and metrics

We evaluate 13 real scenes from Tanks & Temples (Knapitsch et al., 2017) (truck, train), Deep Blending (Hedman et al., 2018) (drjohnson, playroom), and Mip-NeRF 360 (Barron et al., 2022) (nine indoor and outdoor scenes). Full-benchmark experiments use FastGS (Ren et al., 2026) trained for 30k iterations. A controlled second-backbone study replaces only FastGS densification and pruning by vanilla 3DGS rules on three scenes. Everything else is held fixed.

The default event is $(\alpha, \beta) = (0.1, 0.1)$. We redraw the calibration/test split 200 times (100 for the native-calibration audit). These splits reuse the same views and quantify split randomization, not new-scene uncertainty. We therefore use scenes as the sampling unit for paired bootstrap intervals, win counts, and exact two-sided sign tests. We report pooled-pixel coverage, view-event coverage, and mean unclipped half-width. Saving is one minus the ratio of split-averaged widths to those of a constant-scale conformal predictor with the same center, calibration views, and event target.

Same-center baselines are a constant scale, the residual field of 3DGS-U (Galappaththige et al., 2026), the anisotropic visibility field of GAVIS (Xue et al., 2026) rendered in our rasterizer, and the pixel standard deviation of a ten-member deep ensemble (Lakshminarayanan et al., 2017). Each baseline field is scaled by one label-free scene constant, which keeps its cross-view magnitude. Per-view normalization is used only for our $\widetilde{b}$, where a_v carries that magnitude (Appendix C). All shrinkage and GAVIS concentration values come from source scenes, and no target test label is used. Systems with their own predictive center are reported separately, since a new center and a new scale answer different questions.

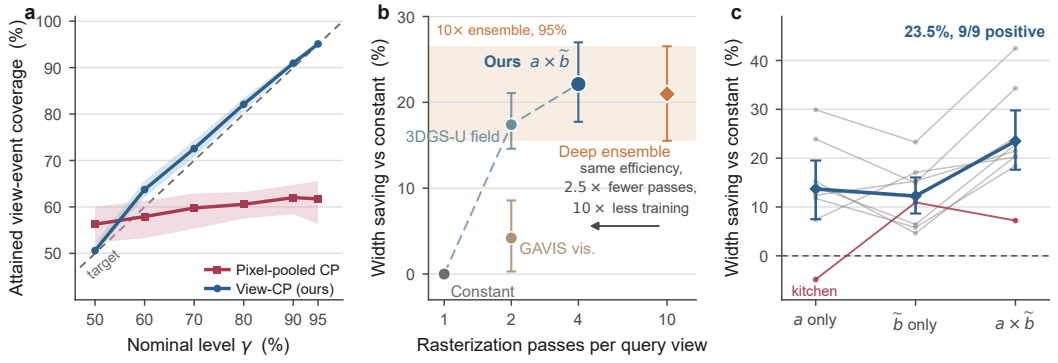


Figure 2: Validity, cost, and strict transfer. **(a)** View-CP tracks the requested view-event level $\gamma = 1 - \alpha = 1 - \beta$ (ten scenes, $m = 21$), while pixel-pooled CP stays near 60% despite 89.9% marginal pixel coverage at the 90% target. Curves are scene means, and bands are interquartile ranges. **(b)** Across 13 scenes, VSCP matches the ten-model ensemble scale (shaded 95% interval) using one model and four query passes instead of ten models and ten passes. **(c)** Under strict family holdout, the two factors remain complementary and their product improves over the constant scale on all nine target scenes. Diamonds show means with 95% scene-bootstrap intervals.

Table 1: Same center and View-CP comparison over 13 scenes. Models / passes denote trained models per scene and rasterizations per query view. Savings are relative to the constant scale at matched coverage, and intervals bootstrap scenes. The GAVIS row uses its visibility field in our shared rasterizer.

Scale	Models / passes	Saving (%)	95% interval	Event cov. (%)
Constant	1× / 1	0.0	–	91.7
3DGS-U spatial field	1× / 2	17.4	[14.6, 21.0]	91.6
GAVIS vis. field	1× / 2	4.2	[0.3, 8.5]	91.7
Ensemble std.	10× / 10	21.0	[15.5, 26.4]	91.4
Ours (VSCP) $a \times \widetilde{b}$	1× / 4	**22.1**	[17.7, 27.1]	91.5

Table 2: Factorization analysis over 13 scenes. $\overline{b}_v$ is the per-view mean of the raw spatial field. Scene-global b retains this cross-view magnitude but uses one shrinkage parameter for the full field.

Scale construction	Saving (%)
Within-view shape $\widetilde{b}$ only	9.9
Raw-field magnitude $\overline{b}_v$ only	12.6
Learned view factor a only	13.6
Scene-global b, one shrinkage (the 3DGS-U row of Table 1)	17.4
Full $a \times \widetilde{b}$ (separate shrinkage)	**22.1**

6.2 Does calibration control the right event?

Pixel pooling and View-CP read the same scale map differently. The former pools all calibration pixels, while the latter applies Equations (15)–(16). At the 90% target, pixel pooling attains 89.9% marginal pixel coverage but only 61.4% view-event coverage. View-CP attains 91.7–92.0% across three very different scale maps (Appendix Table 7). Without ties, the probability in Theorem 1 is exactly $k_\alpha/(m+1)$, 91.9% on average here, and View-CP stays within 0.6 points of this value at every level and m we test (Appendix D). The failure persists across operating points. Pooled calibration stays nearly flat, while View-CP tracks the requested level (Figure 2(a)).

6.3 Matched-coverage scale efficiency

Table 1 fixes the FastGS RGB center, L_∞ residuals, calibration views, and View-CP layer. Only the positive scale map changes. This is the clean comparison of uncertainty structure.

The paired gap to the ensemble scale is 1.2 points [-4.0, 6.6], with 7/13 wins and $p = 1.0$. We therefore claim no efficiency advantage over it. Our claim is about cost. VSCP uses one trained model per scene and four passes per query view instead of ten models and ten passes (Figure 2(b)). The ensemble is also the strongest baseline on the metric we criticize, since it has the best per-view AUSE and Spearman of every scale we tried (Appendix Table 9). Against 3DGS-U, the closest single-model baseline, the gain is 4.7 points [1.6, 7.8], with 11/13 wins and $p = 0.0225$. This is the main result of the same-center comparison. The table studies mechanisms, not strict transfer, because eight of the 13 scenes also belong to the source pool used to select shrinkage. The other five save 23.2%, against 21.4% for these eight.

GAVIS visibility is directional, but interleaved test views are about as visible as training views, so it cannot single out views with large residuals (Appendix D.3).

6.4 Cross-family transfer

Trained only on two Tanks & Temples and two Deep Blending scenes, VSCP transfers without refitting to all nine Mip-NeRF 360 scenes. It saves 23.5% on average, beats the constant scale on all nine ($p = 0.0039$), and matches the source-selected ensemble scale at 21.5%, although that scale requires ten target-specific models. A same-size source pool containing Mip-NeRF 360 reaches

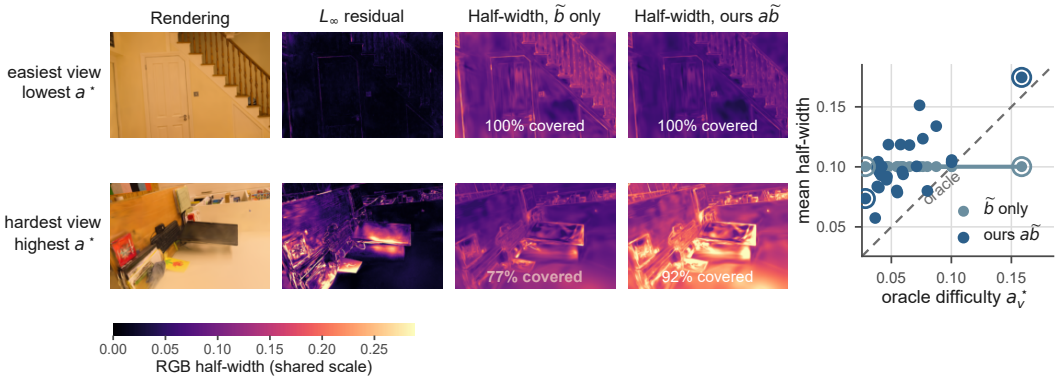


Figure 3: The view factor is invisible within a view and decisive across views. In the preselected playroom scene, rows show the easiest and hardest held-out views by oracle difficulty. Multiplication by a positive view factor preserves the pixel ranking, so the two scale maps in each row should look alike. Across views it raises hard-view coverage from 77% to 92% at a 90% target. Right: mean half-width over all 29 views, where $\tilde{b}$ alone is flat by construction.

24.1%, only 0.6 points higher. Corollary 1 keeps target-scene View-CP valid under such shifts, and view-event coverage is 90.6–93.4% on the nine scenes. Same-family and predictive-center controls are in Appendix Table 14.

6.5 Factorization, metric blind spots, and theory audit

Table 2 separates the value of the view signal from the value of the structure. The raw field magnitude already carries view information. Our learned factor is closer to oracle view difficulty than that magnitude (mean Spearman 0.719 against 0.677), and is a little better on its own (13.6% against 12.6%). The larger gain comes from the structure. Scene-global b uses one shrinkage parameter for cross-view magnitude and within-view shape together, and saves 17.4%. Estimating and shrinking the two parts separately saves 22.1%.

Figure 2(c) shows that the two factors stay complementary under strict holdout, with per-scene values in Appendix Table 15. The oracle factor reaches 41.6% there, so better view-difficulty prediction is still the main room for improvement.

Common ranking metrics cannot see this. By Lemma 3 (Appendix B), the view factor lies in the null space of per-view AUSE and Spearman correlation. Thus $a\tilde{b}$ and $\tilde{b}$ score identically on both, yet they save 22.1% and 9.9% (Appendix Table 9).

6.6 Robustness, labels, and query cost

Without refitting, VSCP retains a 20.6% saving after replacing FastGS density control with vanilla 3DGS rules, despite 5.6–7.7× more Gaussians. A finite View-CP threshold requires $m \geq 9$ at $\alpha = 0.1$ and $m \geq 19$ at $\alpha = 0.05$. Below this limit, the valid set is unbounded. Exact channel packing uses four passes and runs at 216–280 FPS. Target sensitivity, calibration-size sweeps, backbone results, and timing details are in Appendix D, and Appendix E discusses limitations.

7 Conclusion

Reliable rendering requires saying what should be covered, at what statistical unit, and at what cost. We treat a 3DGS view as a structured regression response and calibrate the event that most of its pixels are covered. Renderer-derived spatial and angular structure then makes the boxes efficient. The conformal quantile protects validity while the factorized scale controls width.

AI USE STATEMENT

In this work, we used generative AI tools for improving the language and presentation of this manuscript, including grammar correction, proofreading, and wording refinement. No AI tools were used to generate or develop the research ideas, methodology, experiments, analysis, results, or conclusions. The authors remain fully responsible for all content of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed method, training procedures, datasets, evaluation protocols, and implementation details in the main paper and supplementary material. All datasets used in our experiments are publicly available from their original sources. We will release the source code, configuration files, and scripts required to reproduce the reported experiments upon publication.

REFERENCES

Niki Amini-Naieni, Tomas Jakab, Andrea Vedaldi, and Ronald Clark. Instant uncertainty calibration of NeRFs using a meta-calibrator. In *European Conference on Computer Vision*, pp. 309–324, 2024.

Anastasios N. Angelopoulos, Amit Pal Kohli, Stephen Bates, Michael Jordan, Jitendra Malik, Thayer Alshaabi, Srigokul Upadhyayula, and Yaniv Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 717–730, 2022.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

Stephen Bates, Anastasios N. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Omer Belhasin, Yaniv Romano, Daniel Freedman, Ehud Rivlin, and Michael Elad. Principal uncertainty quantification with spatial correlation for image restoration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3321–3333, 2024.

Joren Brunekreef, Eric Marcus, Ray Sheombarsing, Jan-Jakob Sonke, and Jonas Teuwen. Kandinsky conformal prediction: Efficient calibration of image segmentation algorithms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4135–4143, 2024.

Chamuditha Jayanga Galappaththige, Thomas Gottwald, Peter Stehr, Edgar Heinert, Niko Sünderhauf, Dimity Miller, and Matthias Rottmann. Predictive photometric uncertainty in Gaussian splatting for novel view synthesis. In *European Conference on Computer Vision*, pp. 482–500, 2026.

Zhongpai Gao, Benjamin Planche, Meng Zheng, Anwesa Choudhuri, Terrence Chen, and Ziyan Wu. 6DGS: Enhanced direction-aware Gaussian splatting for volumetric rendering. In *International Conference on Learning Representations*, 2025.

Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20061–20070, 2024.

Fengzhi Guo, Chih-Chuan Hsu, Sihao Ding, and Cheng Zhang. Uncertainty matters in dynamic Gaussian splatting for monocular 4D reconstruction. In *International Conference on Learning Representations*, 2026.

Chenyu Han and Corentin Dumery. View-dependent uncertainty estimation of 3D Gaussian splatting. *arXiv preprint arXiv:2504.07370*, 2025.

Alex Hanson, Allen Tu, Vasu Singla, Mayuka Jayawardhana, Matthias Zwicker, and Tom Goldstein. PUP 3D-GS: Principled uncertainty pruning for 3D Gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5949–5958, 2025.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep Blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6), 2018.

Eddy Ilg, Özgün Çiçek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *European Conference on Computer Vision*, 2018.

Gaoxiang Jia, Vikram Appia, Junzhou Huang, and Xinlei Wang. Rendering-aware Bayesian 3D Gaussian splatting with native uncertainty and adaptive complexity control. *arXiv preprint arXiv:2607.05522*, 2026.

Wen Jiang, Boshu Lei, and Kostas Daniilidis. FisherRF: Active view selection and mapping with radiance fields using Fisher information. In *European Conference on Computer Vision*, 2024.

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.

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

Jingsen Kong, Wenlu Tang, Dezheng Kong, Linglong Kong, Guangren Yang, and Bei Jiang. Adaptive conformal prediction via mixture-of-experts gating similarity. In *International Conference on Learning Representations*, 2026.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 74(1):29–43, 2015.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

Ruiqi Li and Yiu-ming Cheung. Variational multi-scale representation for estimating uncertainty in 3D Gaussian splatting. In *Advances in Neural Information Processing Systems*, 2024.

Rui Luo, Jie Bao, Xiaoyi Su, Wen Jung Li, and Suqun Cao. Enhancing image-conditional coverage in segmentation: Adaptive thresholding via differentiable miscoverage loss. In *International Conference on Learning Representations*, 2026.

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*, pp. 7210–7219, 2021.

Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. ActiveNeRF: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, 2022.

Harris Papadopoulos, Alex Gammerman, and Vladimir Vovk. Normalized nonconformity measures for regression conformal prediction. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 64–69, 2008.

Vincent Plassier, Alexander Fishkov, Mohsen Guizani, Maxim Panov, and Eric Moulines. Probabilistic conformal prediction with approximate conditional validity. In *International Conference on Learning Representations*, 2025.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. FastGS: Training 3D Gaussian splatting in 100 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26094–26103, 2026.

Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3D representations. In *International Conference on 3D Vision*, pp. 972–981, 2021.

Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware NeRF ensembles: Quantifying predictive uncertainty in neural radiance fields. In *IEEE International Conference on Robotics and Automation*, 2023.

Jacopo Teneggi, Matthew Tivnan, J. Webster Stayman, and Jeremias Sulam. How to trust your diffusion model: A convex optimization approach to conformal risk control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33940–33960, 2023.

Anh Thuan Tran and Jana Kosecka. VarSplat: Uncertainty-aware 3D Gaussian splatting for robust RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Feng Wu, Tsai Hor Chan, Yihang Chen, Lingting Zhu, Guosheng Yin, and Lequan Yu. Horseshoe splatting: Handling structural sparsity for uncertainty-aware Gaussian-splatting radiance field rendering. In *International Conference on Learning Representations*, 2026.

Shangjie Xue, Jesse Dill, Dhruv Ahuja, Frank Dellaert, Panagiotis Tsiotras, and Danfei Xu. Uncertainty-driven 3D Gaussian splatting active mapping via anisotropic visibility field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5014–5026, 2026.

Yunzhen Yao, Lie He, and Michael Gastpar. Non-asymptotic analysis of efficiency in conformalized regression. In *International Conference on Learning Representations*, 2026.

Botong Zhang, Shuo Li, and Osbert Bastani. Conformal structured prediction. In *International Conference on Learning Representations*, 2025.

Appendix

View-Structured Conformal Prediction for 3D Gaussian Splatting

A Additional Efficiency Results

Section 5 treats $\log a_v^\star$ as the regression target without saying when it recovers real view difficulty. The following proposition answers that, and degrades gracefully when the assumption only holds approximately.

Proposition 3 (Identification of the oracle view factor)**.** *Fix the deployed shape $\widetilde{b}$ and assume the separable model $r_{vp} = A_v \widetilde{b}_{vp}\, \varepsilon_{vp}$, in which $A_v > 0$ is the difficulty of view v relative to that shape. Write $c_v = Q_{1-\beta,p}(\varepsilon_{vp})$ for the inner order statistic of the view's own noise. Then $a_v^\star(\widetilde{b}) = A_v c_v$, and:*

1. *if $c_v \equiv c_\beta$ across views, the oracle factor equals A_v up to a single global constant, which by Lemma 1 changes no interval;*
2. *if $|\log c_v - \log c_\beta| \le \eta$ and the deployed factor tracks A_v up to a constant, $|\log(a_v/A_v) - \bar{e}| \le \epsilon$ for some $\bar{e}$, then $|\delta_v - \delta_0| \le \epsilon + \eta$ with $\delta_0 = \bar{e} - \log c_\beta$, and Theorem 2 gives $W_i(a)/W_i(a^\star) \le e^{2(\epsilon+\eta)}$.*

The identity is immediate from positive homogeneity: dividing r_{vp} by $\widetilde{b}_{vp}$ leaves $A_v \varepsilon_{vp}$, whose $\lceil n_v(1-\beta)\rceil$-th order statistic is $A_v c_v$. Two things are worth noting. No pixel-independence assumption is used; what "view-invariant noise" must control is only the inner quantile c_v of each view, not its full noise law. And because $\widetilde{b}$ is normalized within each view, A_v absorbs the per-view magnitude of the underlying spatial field, which is why that magnitude alone already carries view information in Table 2.

Theorem 2 (Deterministic excess-width bound)**.** *Assume $k_\alpha \le m$. Fix $\widetilde{b}$ and suppose the post-shrinkage log errors $\delta_v = \log(a_v/a_v^\star)$ satisfy $|\delta_v - \delta_0| \le \epsilon$ for some constant δ_0 and all calibration and test views. For every test view i,*

$$e^{-2\epsilon} \le \frac{W_i(a)}{W_i(a^\star)} \le e^{2\epsilon}. \tag{22}$$

Hence the multiplicative excess width is at most $e^{2\epsilon} - 1 = 2\epsilon + O(\epsilon^2)$.

By Lemma 1, replacing a by $e^{-\delta_0} a$ changes no interval and shifts every δ_v by $-\delta_0$, so we may take $\delta_0 = 0$. This matters in practice, since the deployed factor has batch mean one while $a^\star$ is in RGB units. Equation (19) then gives calibration scores $e^{-\delta_v}$, so their order statistic lies in $[e^{-\epsilon}, e^{\epsilon}]$; multiplying by $a_i = a_i^\star e^{\delta_i}$ gives the result. The theorem is deterministic and needs neither a score-density lower bound nor an asymptotic calibration approximation. It also follows directly from the exact decomposition because $|\delta_i - \delta_{(j_\alpha)}| \le 2\epsilon$.

The same factorization identifies the realizable benefit of view adaptation. For the oracle factor versus a constant view factor,

$$\frac{\overline{W}(a^\star)}{\overline{W}(1)} = \frac{\overline{a^\star}_{\text{test}}}{Q_{k_\alpha}(\{a_v^\star : v \in \text{cal}\})}. \tag{23}$$

This is an exact finite-sample identity conditional on one calibration/test split: a perfect view model replaces an upper quantile of view difficulty by its mean, and absolute scene error cancels. It predicts larger gains when view difficulty is more heterogeneous and holds numerically to 6.2×10^{-7} over all 13 scenes.

B PROOFS

B.1 GAUGE INVARIANCE

Lemma 1 (Global scale invariance)**.** *Let s be a positive scale and $c > 0$ a constant. Then $R_v(cs) = R_v(s)/c$ for every view, hence $\widehat{q}(cs) = \widehat{q}(s)/c$ and $\widehat{q}(cs)\, cs_{vp} = \widehat{q}(s)\, s_{vp}$ for every pixel. Multiplying a positive scale by a global constant leaves every calibrated interval unchanged.*

Both claims follow from positive homogeneity of order statistics. Dividing every ratio r_{vp}/s_{vp} by c divides each inner order statistic by c, and therefore divides the outer order statistic of the calibration scores by c as well; the two factors cancel in the product $\widehat{q}\, s$. □

This one fact underlies four separate conventions in the paper: the multiplicative non-identifiability of a and b, the query-batch mean normalization of our raw view predictions, the scene-level normalization applied to every baseline scale map, and the exact cancellation of global multiplicative bias in Proposition 2.

B.2 PROOF OF LEMMA 2

Lemma 2 (View event as an order-statistic event)**.** *For a fixed view v and positive scale s, the event in Equation* (6) *holds at multiplier q if and only if $q \geq R_v(s)$.*

For positive s_{vp}, $Y_{vp} \in C_{vp}(q, s)$ if and only if $r_{vp}/s_{vp} \leq q$. Consequently the within-view coverage fraction is the empirical cumulative distribution function of $\{r_{vp}/s_{vp} : p \in \mathcal{P}_v\}$. It first reaches $1 - \beta$ at the $\lceil n_v(1-\beta) \rceil$-th order statistic, which is exactly $R_v(s)$. □

B.3 PROOF OF THEOREM 1

By Lemma 2, the new-view event is equivalent to $R_V \leq \widehat{q}$. Under exchangeability, the rank of R_V among the $m+1$ scores is uniform after random tie breaking and conservative without it. For $k_\alpha = \lceil (m+1)(1-\alpha) \rceil \leq m$,

$$\mathbb{P}\{R_V \leq R_{(k_\alpha)}\} \geq \frac{k_\alpha}{m+1} \geq 1 - \alpha. \tag{24}$$

If scores have no ties, equality holds in the first relation and $k_\alpha/(m+1) < 1 - \alpha + 1/(m+1)$. When $k_\alpha > m$, our convention is $\widehat{q} = +\infty$, so the lower bound remains true but the set is uninformative. □

B.4 LABEL-FREE BATCH CONSTRUCTION OF THE SCALE

Theorem 1 needs the calibration and test scores to be exchangeable, but our scales are not fixed functions of one view in isolation: they are built from the covariates of the whole query batch. The next proposition says this is harmless provided the construction ignores the order of the batch.

Proposition 4 (Covariate-transductive scale construction)**.** *Let $(Z_1, Y_1), \ldots, (Z_N, Y_N)$ be exchangeable view pairs, with $N > m$, and let the positive scales be produced jointly from all unlabeled covariates,*

$$(S_1, \ldots, S_N) = \Phi(Z_1, \ldots, Z_N; \mathcal{D}_{\text{source}}, \widehat{\mu}), \tag{25}$$

where $\mathcal{D}_{\text{source}}$ and the renderer $\widehat{\mu}$ are frozen before any target calibration label is observed and Φ is permutation equivariant, $\Phi_i(Z_{\pi(1)}, \ldots, Z_{\pi(N)}) = \Phi_{\pi(i)}(Z_1, \ldots, Z_N)$ for every permutation π. After this construction, draw a uniform calibration subset of size m, independently of the view pairs. Then the scores $R_i = R(Z_i, Y_i; S_i)$ are exchangeable, and View-CP satisfies Theorem 1 for a marginal test view from the complement.

Permuting the views permutes the covariate vector; by equivariance it permutes the scale vector the same way, hence the score vector the same way. A permutation-equivariant function of an exchangeable sequence is exchangeable. The uniform split is independent and permutation symmetric, so the m calibration scores and a marginal test score retain the rank symmetry used by Theorem 1. □

Three constructions in this paper satisfy Equation (25): normalizing our positive raw view predictions to mean one over the query batch, the scene constant c_U of Equation (30) applied to every baseline map, and any summary of camera poses or renderer features of the batch. What matters is that a single symmetric rule processes calibration and test views together. Computing separate normalizing constants on the calibration batch and on the test batch would break the argument. Our implementation does not do that: all batch statistics are formed over every query view of a scene, before the calibration/test split is drawn.

B.5 Proof of Corollary 1

Theorem 1 uses only exchangeability of the scores after all pre-calibration objects are fixed. It does not require that s estimate a conditional standard deviation or any true uncertainty function. Conditional on $\mathcal{F}$, the target calibration and test scores are exchangeable by assumption, so the rank argument is untouched. Changing $\mathcal{F}$ may alter the predictive center, the scale, and thus the score distribution and interval width, but not the conditional rank guarantee. □

B.6 Proof of Proposition 1

By homogeneity of order statistics,

$$R_v(c\,\widetilde{b}) = Q_{1-\beta,p}\left(\frac{r_{vp}}{c\,\widetilde{b}_{vp}}\right) = \frac{1}{c}Q_{1-\beta,p}\left(\frac{r_{vp}}{\widetilde{b}_{vp}}\right) = \frac{a_v^\star}{c}. \tag{26}$$

Lemma 2 with $q = 1$ says the event holds if and only if $c \geq a_v^\star$, proving both minimality and the identity. □

B.7 Ranking null space of the view factor

Lemma 3 (Ranking null space)**.** *Fix a view v and a positive shape $\widetilde{b}_v$. For every $a_v > 0$ the within-view ranking is unchanged,* $\mathrm{rank}_p(a_v\widetilde{b}_{vp}) = \mathrm{rank}_p(\widetilde{b}_{vp})$*, so any metric M_v that reads the scale only through that ranking satisfies $M_v(a_v\widetilde{b}_v, r_v) = M_v(\widetilde{b}_v, r_v)$. At the same time $R_v(a_v\widetilde{b}_v) = R_v(\widetilde{b}_v)/a_v$ by Proposition 1.*

Multiplication by a positive scalar is strictly increasing, so it preserves the order of $\{\widetilde{b}_{vp}\}_p$, and per-view AUSE and Spearman correlation are functions of that order alone. □

The two halves of the lemma are the two halves of Figure 3: the view factor is invisible to per-view ranking diagnostics (Table 9) and is exactly the quantity that divides the conformal width (Table 2).

B.8 Proof of Theorem 2

By Lemma 1 we may take $\delta_0 = 0$, so $|\delta_v| \leq \epsilon$ for every view. Equation (19) gives calibration scores $R_v(a_v\widetilde{b}) = e^{-\delta_v}$. Their k_α-th order statistic therefore lies in $[e^{-\epsilon}, e^{\epsilon}]$. Since $\widetilde{b}$ has mean one in every view,

$$W_i(a) = \widehat{q}\,a_i = \widehat{q}a_i^\star e^{\delta_i} \in [e^{-2\epsilon}a_i^\star, e^{2\epsilon}a_i^\star]. \tag{27}$$

For the oracle factor, every calibration score equals one, so $\widehat{q} = 1$ and $W_i(a^\star) = a_i^\star$. Dividing proves the result. □

B.9 Proof of Proposition 2

Equation (19) gives scores $e^{-\delta_v}$. Because the exponential is strictly decreasing, the k_α-th smallest score is the $(m - k_\alpha + 1)$-th smallest calibration error. The integer identity

$$m - \lceil (m+1)(1-\alpha) \rceil + 1 = \lfloor (m+1)\alpha \rfloor = j_\alpha \tag{28}$$

therefore gives $\widehat{q} = e^{-\delta_{(j_\alpha)}}$. Since $\widetilde{b}$ has within-view mean one,

$$W_i(a) = \widehat{q}\,a_i = a_i^\star e^{\delta_i - \delta_{(j_\alpha)}}. \tag{29}$$

The oracle has $W_i(a^\star) = a_i^\star$, proving the pointwise identity. Averaging numerator and denominator separately yields Equation (21). □

B.10 Proof of Equation (23)

Fix the calibration and test sets. By Proposition 1 the oracle factor gives every calibration view a score of one, so $\widehat{q} = 1$ and its mean half-width on test view i is $a_i^\star$. A constant view factor gives calibration scores $a_v^\star$, hence outer multiplier $Q_{k_\alpha}(\{a_v^\star : v \in \mathrm{cal}\})$ and that same mean half-width on every test view. Taking the ratio of test averages proves Equation (23), which therefore also supplies a direct numerical audit of the implementation. □

C Implementation and Protocol Details

C.1 Pseudo code of VSCP

Algorithm 1 View-Structured Conformal Prediction (VSCP).

Phase 1: meta-training on source scenes (offline, once)
for each source scene **do**
  fit its 3DGS and its residual field $\widehat{C}$ on that scene's training views; accumulate E_i, D_i
  fit K fold models, the k-th without the training views of fold k
  for each fold k, and each training view v held out of fold k **do**
    $z_v \leftarrow$ descriptor of v built from the cameras and support of the *other* folds
    $\widetilde{b}_v \leftarrow$ *that scene's own* shape at v ▷ Equation (9)
    $r_{vp} \leftarrow$ residual of fold model k at v
    $a_v^\star \leftarrow Q_{1-\beta,p}(r_{vp}/\widetilde{b}_{vp})$ ▷ needs a label; source scenes only
  end for
end for
center $\log a^\star$ within each scene, pool, fit θ, and select λ_a, λ_b
$\theta, \lambda_a, \lambda_b$ are frozen from here on and are the only objects that cross the scene boundary.

Phase 2: target scene, training views
fit the 3DGS $\widehat{\mu}$ and the residual field $\widehat{C}$; accumulate E_i, D_i
compute the feature mean and scale μ_z, σ_z on these views

Phase 3: target scene, whole query batch $\mathcal{V}$, no labels
for every held-out view $v \in \mathcal{V}$ **do**
  render $\widehat{\mu}_v$ and b_{vp}, and form $\widetilde{b}_{vp}$ ▷ Equation (9)
  build z_v and evaluate $\widehat{a}(z_v)$ ▷ Equation (14)
end for
normalize $\widehat{a}$ to mean one **over all of** $\mathcal{V}$; set a_v and $s_{vp} = a_v\widetilde{b}_{vp}$

Phase 4: target scene, calibration and output
draw m calibration views from $\mathcal{V}$; the remainder are test views
for each calibration view v **do**
  $R_v \leftarrow Q_{1-\beta,p}(r_{vp}/s_{vp})$ ▷ the only use of a target label
end for
$\widehat{q} \leftarrow Q_{k_\alpha}(\{R_1, \ldots, R_m\})$ ▷ Equation (16)
for each test view V **do**
  output $\widehat{\mu}_{Vp} \pm \widehat{q}\, a_V \widetilde{b}_{Vp}$ ▷ Equation (17)
end for

Rendering cost. The RGB prediction uses one rasterization. Seven per-Gaussian auxiliary scalars are packed into the three RGB channels of three additional rasterizations. Channels are composited independently with the same αT weights, so packing is an algebraic reorganization rather than an approximation. The complete query therefore uses one backbone and four rasterization passes. Spatial-field fitting and support accumulation are one-time post-training operations.

C.2 PROTOCOL MANIFEST

Table 3 lists every scene with its view counts. Renderer training views fit the target-scene 3DGS and its residual field. The V held-out views are every eighth image, as in the standard 3DGS split. They are split into $m = \max(11, \lfloor V/3 \rfloor)$ calibration views and $V - m$ test views. We redraw that split 200 times with seed 0, and 100 times for the native-calibration audit of Table 7. Repeated splits reuse the same views, so every bootstrap interval, win count and sign test in the paper resamples *scenes*, never splits.

Table 3: Per-scene manifest. V is the number of held-out views and m the calibration size. Gaussians are counted in the seed-0 model.

Scene	Dataset	Train	V	m	Test	Gauss. (k)
truck	Tanks & Temples	219	32	11	21	276
train	Tanks & Temples	263	38	12	26	220
drjohnson	Deep Blending	230	33	11	22	388
playroom	Deep Blending	196	29	11	18	246
bicycle	Mip-NeRF 360	169	25	11	14	858
bonsai	Mip-NeRF 360	255	37	12	25	251
counter	Mip-NeRF 360	210	30	11	19	197
flowers	Mip-NeRF 360	151	22	11	11	662
garden	Mip-NeRF 360	161	24	11	13	511
kitchen	Mip-NeRF 360	244	35	11	24	284
room	Mip-NeRF 360	272	39	13	26	210
stump	Mip-NeRF 360	109	16	11	5	629
treehill	Mip-NeRF 360	123	18	11	7	804

Source pools. Shrinkage selection uses eight source scenes: truck, train, drjohnson, playroom, bicycle, counter, garden, and room. The same-family transfer in Table 14 learns the view regressor on those eight and tests on the five Mip-NeRF 360 scenes they do not contain, namely bonsai, flowers, kitchen, stump, and treehill. The strict holdout uses only truck, train, drjohnson, and playroom as sources, and tests on all nine Mip-NeRF 360 scenes. No Mip-NeRF 360 view enters the strict source pool in any role.

Query batch. The batch is every held-out view of the target scene, that is all V views of Table 3, and it is formed before the calibration/test split is drawn. Only unlabeled covariates are read from it: camera poses and renderer-derived maps. This is the map Φ of Proposition 4.

View descriptor. The deployed descriptor has 24 entries. Five are camera geometry (`g_dnn`, `g_dk`, `g_ang_min`, `g_ang_k`, `g_dens`). Eight are render statistics (`r_alpha` and `r_grad`, each as mean, 10th and 50th percentile, plus `r_depth_mean` and `r_depth_cv`). Nine are support statistics (`sup_logE`, `sup_ang`, `sup_spread`, each as mean, 10th and 50th percentile). Two are view-level log-means of spatial fields (`f_gsu_logmean`, `f_train_logmean`). A 25th feature, the log-mean of an out-of-fold residual field, exists in the source-side schema but is excluded from the deployed schema, because it would need fold models at target inference. The two schemas are compared entry by entry when a fit is loaded, and a mismatch stops the run.

Table 4 removes or keeps one feature group at a time and refits the cross-scene regressor on the eight source scenes. The two field log-means carry most of the signal, which matches the raw-field row of Table 2. Among the other groups, support is the strongest on its own, and the full descriptor is the best.

Hyperparameters. We select λ_a and λ_b jointly on the source pool over $\{0, 0.25, 0.5, 0.75, 1\}$, rebuilding the regression target at each λ_b so that it always matches the shape that will be deployed. The ridge penalty is chosen over $\{0.1, 1, 10, 100, 1000\}$ by leave-one-scene-out Spearman correlation against the oracle factor. Baseline λ_U uses the same grid and is scored once on the whole pool, rather than by averaging per-scene minima, because such an average can fall between grid points. The selected values are $\lambda_a = 1.0$ and $\lambda_b = 0.5$ for our scale, $\lambda_U = 0.5$ for the 3DGS-U and ensemble maps, and $\lambda_U = 0.5$ with $\kappa = 64$ for GAVIS. The strict-holdout ridge is 100.

Table 4: Feature-group ablation of the view regressor. Entries are leave-one-scene-out Spearman correlations with the oracle factor over the eight source scenes. The groups are correlated, so a small drop after removing one group means redundancy rather than irrelevance.

Feature set	Dim.	Spearman
All (deployed)	24	0.705
Without camera	19	0.673
Without render	16	0.675
Without support	15	0.668
Without field log-means	22	0.457
Camera only	5	0.292
Render only	8	0.254
Support only	9	0.445
Field log-means only	2	0.616

Label access. Table 5 states what each group of views may supply. The last row is the one that matters for Theorem 1: target test views contribute camera poses and renderer features to the query batch, and never contribute an image.

Table 5: What each group of views may supply. "Selection" covers the view regressor, the ridge penalty, and every shrinkage constant. Target calibration images enter only the outer order statistic of Equation (16).

View group	RGB image	Camera pose	Renderer features	Selection
Target-scene renderer training	yes	yes	yes	no
Source-scene meta-training	yes	yes	yes	yes
Target calibration	yes	yes	yes	no
Target test	no	yes	yes	no

C.3 Feature construction

For camera centers o_v and forward axes h_v, geometry features include the nearest and mean five-neighbor distances (normalized by scene extent), minimum and mean five-neighbor angular differences, and local camera density. Render features aggregate alpha, image gradient, and normalized depth. Support features aggregate the three per-Gaussian fields in Equation (12). Each map contributes its mean and selected lower quantiles; the spatial field contributes view-level log-mean summaries. Source and deployment schemas are checked to exclude any feature that requires fold-model residuals at target inference.

Three packed auxiliary renders contain: (i) ones, depth, and log exposure; (ii) support angle, support spread, and the spherical-harmonic spatial field; and (iii) a training-residual summary. Together with RGB, this gives four passes. Packing is exact because each output channel uses the same weights but is accumulated independently.

C.4 Fair normalization of baseline scale maps

For a baseline map U_{vp}, we compute one label-free scene constant over the unlabeled query batch,

$$c_U = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{1}{n_v} \sum_{p} U_{vp}, \qquad s^U_{vp} = (1 - \lambda_U) + \lambda_U \frac{U_{vp}}{c_U}. \tag{30}$$

This keeps the baseline's cross-view magnitude. The construction is permutation equivariant (Proposition 4) and uses no image label, and its global constant is absorbed by the conformal multiplier (Lemma 1). We select λ_U on source scenes. In contrast, our spatial field is normalized inside each view because its cross-view magnitude is assigned to a_v. Applying that identifiability rule to a baseline would remove view-level information from its raw scale map.

This choice moves the main-table numbers, so Table 6 reports the whole grid rather than only the selected point. Per-view normalization divides each baseline map by its own view mean, which deletes the cross-view magnitude that the map carries. Scene-level normalization keeps that magnitude and fixes only the global unit, which Lemma 1 shows the conformal multiplier would absorb anyway. The $\lambda = 1$ row is the raw scale with no shrinkage. Scene-level normalization is better than per-view normalization for all three baselines at every $\lambda > 0$, so the reported baselines are the stronger of the two versions, not the weaker.

Table 6: Baseline normalization sensitivity. Entries are pool-mean relative width, so lower is better and $\lambda = 0$ is the constant scale by definition. "pv" normalizes each map inside each view; "sg" uses one scene constant. The last row is the 13-scene saving at the λ selected on the pool, marked *.

	3DGS-U field		Ensemble std.		GAVIS vis.	
λ	pv	sg	pv	sg	pv	sg
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.25	0.9326	0.8781	0.9078	0.8097	0.9878*	0.9579
0.50	0.9180*	0.8329*	0.8968*	0.7772*	0.9892	0.9416*
0.75	0.9505	0.8413	0.9505	0.8115	1.0063	0.9561
1.00	1.0831	0.9880	1.1352	0.9521	1.0470	1.0081
Saving (%)	9.9	**17.4**	10.5	**21.0**	1.4	**4.2**

The bold savings are the values reported in Table 1. Had we used per-view normalization instead, the three baselines would have scored 9.9%, 10.5% and 1.4%, and our 22.1% would have looked far stronger than it should. The gap is largest for the ensemble, whose cross-view magnitude is its most useful signal.

C.5 Calibration feasibility

The smallest calibration size is the first m satisfying

$$\lceil (m+1)(1-\alpha) \rceil \leq m. \tag{31}$$

For commonly used levels,

α	Target view-event probability	Minimum m
0.10	0.90	9
0.05	0.95	19

At smaller m, a finite threshold cannot carry the stated distribution-free guarantee. This discreteness must not be hidden by an interpolated quantile.

D Additional Experiments and Diagnostics

D.1 Native scores versus conformalized intervals

Table 7 compares three readings of the same uncertainty map. "Native" uses the published residual or Gaussian interpretation without a view-level correction. Pixel pooling selects one multiplier from all calibration pixels. View-CP uses Equations (15)–(16).

For completeness, Table 8 gives the values plotted in Figure 2(a).

D.2 The view event as a distribution

Table 7 reports two averages. Figure 4 shows the distribution behind them. The x axis is the fraction of a view's pixels that the interval covers. The curve is the empirical distribution function over held-out test views, pooled over the same 13 scenes, the same m, and the same 100 splits.

Table 7: Coverage averaged over 13 scenes at $(\alpha, \beta) = (0.1, 0.1)$ with $m = \max(11, \lfloor V/3 \rfloor)$, the calibration size used throughout the main results. Each entry is pooled-pixel / view-event coverage (%). View-CP satisfies the targeted view event; pooled calibration can match marginal pixels while leaving roughly 40% of views below target.

Scale map	Native interpretation	Pixel-pooled CP	View-CP
Ensemble standard deviation	30.5 / 0.0	89.9 / 59.7	94.8 / 92.0
3DGS-U spatial field	72.5 / 0.6	89.9 / 61.2	95.3 / 92.0
Ours: $a \times \widetilde{b}$	n/a	89.9 / 61.4	95.0 / 91.7

The guarantee is a corner, not a curve. Theorem 1 asks that at most α of views fall below $1 - \beta$, that is $F(1 - \beta) \leq \alpha$. Drawing both lines makes that checkable by eye. View-CP passes under the corner at 7.9%. Pixel-pooled CP misses it at 38.9%.

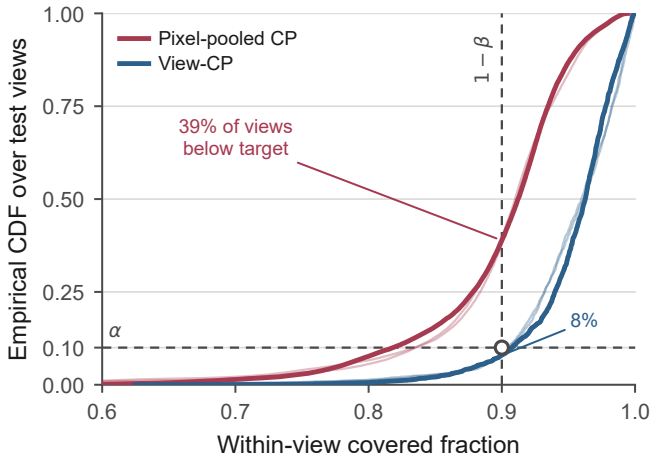


Figure 4: Within-view covered fraction at $(\alpha, \beta) = (0.1, 0.1)$, as an empirical CDF over 23,100 held-out (view, split) pairs from 13 scenes. The bold curves use the 3DGS-U field, the map of Figure 2(a). The faint curves behind them are the ensemble standard deviation and our scale. The dashed lines and the open circle mark the guarantee: a valid procedure passes below $(1 - \beta, \alpha)$.

At this operating point $\gamma = 0.9$, so the matched requirement $\beta = 1 - \gamma$ equals the fixed $\beta = 0.1$, and nothing in the figure depends on which convention is used. That is worth saying, because at other levels it does matter. With β pinned while γ moves, a pooled multiplier calibrated to a marginal rate below $1 - \beta$ cannot satisfy the view event for arithmetic reasons, and a flat pooled curve would then be forced rather than observed. Figure 2(a) matches β to the level for the same reason.

The three scale maps give almost the same pair of curves. Pooled CP fails on 38.0–40.0% of views and View-CP on 7.9–8.3%. The split is a property of the calibration unit, not of the scale map: Theorem 1 holds for each of the three.

Table 8: View-event calibration curve (%) for the 3DGS-U spatial field, over the ten scenes of Figure 2(a) at $m = 21$. Here $\gamma = 1 - \alpha = 1 - \beta$, so the within-view requirement moves with the level and pooled calibration is never arithmetically prevented from meeting it. The last row is the exact coverage of View-CP without ties.

Nominal γ	50	60	70	80	90	95
Pixel-pooled CP	56.3	57.9	59.8	60.6	62.0	61.7
View-CP	50.6	63.7	72.6	82.1	90.9	95.1
Exact $k_\alpha/(m+1)$	50.0	63.6	72.7	81.8	90.9	95.5

When a baseline publishes a score map rather than a calibrated RGB interval, we report the two objects separately. Reading the ensemble map as a Gaussian standard deviation gives only 30.5% marginal pixel coverage at nominal 90%; in the sweep run of Table 8, View-CP multiplies its native Gaussian radius by 4.84, while the 3DGS-U residual map starts at 72.5% pixel coverage and requires a 2.31× multiplier. These large corrections do not imply poor ranking. The ensemble has the best AUSE in Table 9. Instead, they show that ranking units are not automatically interval units. Constant, visibility, and our normalized scale have no native standard-deviation interpretation, so we do not manufacture one.

D.3 GAVIS VISIBILITY-FIELD SCOPE AND SANITY CHECK

We import the released degree-two spherical-harmonic visibility representation from GAVIS, but not its separate uncertainty-aware Bayesian rasterizer. For a Gaussian and view, we replace the released last-written-pixel transmittance by the footprint-weighted average

$$\overline{T}_{i,v} = \frac{\sum_p \alpha_{ivp} T_{ivp}}{\sum_p \alpha_{ivp}} \in [0,1], \tag{32}$$

then alpha-composite visibility in our shared rasterizer. We select $\kappa \in \{1, 4, 16, 64\}$ and shrinkage on source scenes; $\kappa = 64$ is best, while the released $\kappa = 1$ saturates under our much denser training-view regime. The directional field passes its intended sanity check:

Scene	Training dir.	Test dir.	Random dir.	Opposite dir.
truck	0.4332	0.4332	0.1341	0.2324
bicycle	0.3181	0.3220	0.0947	0.1224

Training directions are 3.2–3.4× more visible than random directions, but interleaved test views are as visible as training views. This supports the mechanism interpretation in Section 6 without claiming a full-system reproduction.

D.4 CALIBRATION-SIZE SWEEP

For the m sweep, we use the same seven scenes for every setting (bonsai, counter, drjohnson, kitchen, room, train, and truck) and hold the test set at five views so that only calibration size changes. On stump, which is too small for this sweep, increasing m from 9 to 11 changes saving only from 19.5% to 20.0%; the important cost is crossing the finite-sample feasibility threshold.

m	9	11	15	20	25
Saving (%)	20.7	21.4	23.0	17.1	18.1
Event cov. (%)	89.6	91.4	94.0	90.5	92.4
Exact $k_\alpha/(m+1)$ (%)	90.0	91.7	93.8	90.5	92.3

The non-monotonic saving is predicted by the discrete outer order statistic: for $m \in \{9, 11, 15\}$, $k_\alpha = m$ and calibration uses the maximum score; for $m \in \{20, 25\}$ it does not. Coverage follows the exact value in the last row for every m. More calibration labels buy feasibility and stability, not monotonic width reduction.

D.5 FACTORIZATION, RANKING METRICS, AND THEORY AUDIT

The fold models are needed only on source scenes. A target built instead from the deployed model's own residuals, which are in-sample, gives a 23.5% mean saving on the eight-source pool. The out-of-fold target gives 24.0% at eleven times the source training cost. It is better on 6/8 scenes, but on playroom it saves only 4.1% while the in-sample target saves 21.2%. There the out-of-fold factor ranks the test views better (Spearman 0.68 against 0.47), so a better ranking alone does not guarantee a narrower interval.

Table 9: Uncertainty ranking versus interval efficiency over 13 scenes. AUSE and Spearman are computed separately within each view.

Scale	AUSE ↓	Spearman ↑	Saving (%) ↑
GAVIS vis. ($\kappa = 64$)	0.4796	0.1302	4.2
Ensemble std.	**0.2331**	**0.4657**	21.0
Ours: spatial shape $\widetilde{b}$	0.2479	0.4604	9.9
Ours $a \times \widetilde{b}$	0.2479	0.4604	**22.1**

Finally, we numerically audit Proposition 2 over 13 scenes and 200 splits. The three identities in Equation (20) and Equation (21) hold to relative error below 1.1×10^{-6}. After centering the non-identifiable global error, the test-side factor is 0.975 and the calibration-tail factor is 1.420, yielding the observed oracle width ratio 1.384. Hence almost all excess width is paid for the lower tail of view-factor error. Here $j_\alpha = 1$ in all 13 scenes: one underestimated calibration view determines the global conformal inflation.

D.6 Sensitivity to the coverage targets

Table 10 changes one target at a time. Each α uses $m = \max(m_{\min} + 2, \lfloor V/3 \rfloor)$, with $m_{\min}$ from Appendix C.5. At $\alpha = 0.05$ this gives $m = 21$, which only nine scenes support, so the α sweep uses those nine (bicycle, bonsai, counter, drjohnson, kitchen, playroom, room, train, and truck).

Table 10: Sensitivity of saving and view-event coverage. In the β sweep, the view predictor remains the one trained for $\beta = 0.1$, deliberately testing scale misspecification. "Exact" is $k_\alpha/(m+1)$ averaged over the scenes. Each sweep recomputes its rows, so the $\beta = 0.1$ row can differ from Table 1 in the last digit.

Sweep	Setting	Scenes	Saving (%)	Event cov. / exact / target (%)
α	0.05	9	24.2	95.6 / 95.5 / 95
	0.10	9	21.4	91.7 / 91.9 / 90
	0.20	9	13.9	84.2 / 83.8 / 80
β	0.05	13	27.6	91.9 / 91.9 / 90
	0.10	13	22.0	91.7 / 91.9 / 90
	0.20	13	15.4	91.6 / 91.9 / 90

D.7 Cross-backbone transfer

We replace FastGS density control by vanilla 3DGS densification while keeping the rasterizer and all uncertainty code fixed. This produces 5.6–7.7× more Gaussians at comparable residual level. The view regressor and shrinkage parameters remain those learned on FastGS; they are not refit.

Table 11: Saving (%) and event coverage for the controlled vanilla-densification backbone. The full factorization remains better than either factor alone.

Scene	a only	$\widetilde{b}$ only	Full	Oracle	Event cov.
truck	6.0	7.6	12.6	31.7	91.6
bicycle	19.2	4.4	23.2	43.1	91.3
room	13.3	15.4	26.1	52.6	92.5
Mean	12.8	9.1	**20.6**	42.5	–

Table 12: End-to-end query latency on an RTX 4090, including RGB prediction. Ours uses one target-scene model and four rasterization passes; the ensemble uses ten models and ten RGB passes.

Scene (MP)	Ours (ms)	Ours FPS	Ensemble (ms)	Speedup
truck (0.53)	3.57	280	8.67	2.43×
bicycle (1.02)	4.63	216	11.86	2.56×
room (1.62)	3.91	256	9.06	2.32×

D.8 Rendering latency and exact channel packing

Channel packing reduces the previous eight-pass implementation to four passes without changing predictions: the maximum relative difference is zero for the key alpha and spatial maps and 6.23×10^{-8} over the complete feature dictionary. Table 12 shows 216–280 FPS across 0.53–1.62 MP. These numbers are specific to FastGS and this hardware; they are not a cross-paper claim against differently measured real-time UQ systems. Building the GAVIS visibility field is cheaper than our conjugate-gradient residual field, but its width result in Table 1 shows that the performance gap is not caused by query cost.

D.9 What a calibrated interval looks like

Figure 5 shows the objects the paper is actually about: an interval, and the pixels it misses. We fixed the three scenes in advance to cover three regimes. Flowers is spatial-shape dominant, since $\widetilde{b}$ alone saves 17.1% there under strict holdout while a alone saves 7.4%. Kitchen is our weakest scene, where the view factor alone loses 4.8% and the full scale saves only 7.2%. Room uses the vanilla-densification backbone, with the view predictor and both shrinkages carried over from FastGS without refitting.

Within each scene we show the view of median oracle difficulty $a_v^\star$. Figure 3 already shows the two extremes of a scene, so repeating that rule would show the same thing twice. The median view answers the different question of what a typical deployment looks like. The displayed view is always excluded from the m calibration views that produce $\widehat{q}$.

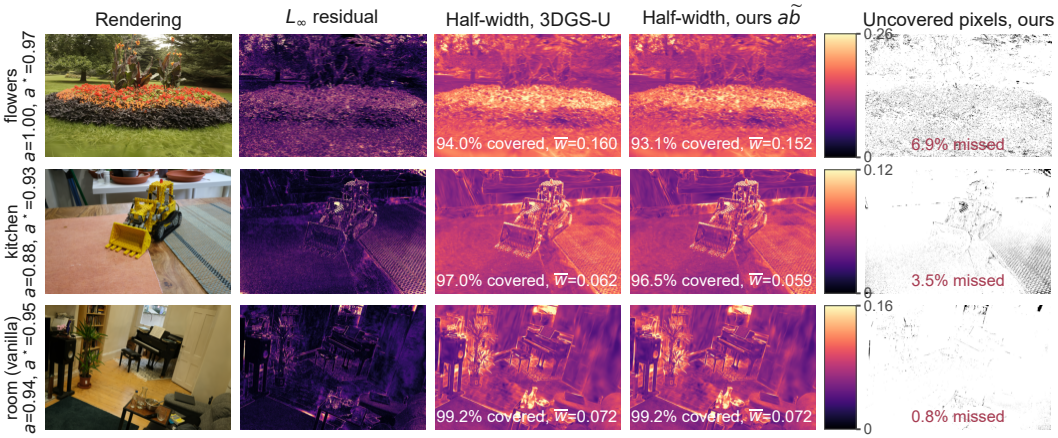


Figure 5: Calibrated intervals on one held-out view per scene, at $(\alpha, \beta) = (0.1, 0.1)$. Columns 2–4 of a row share one absolute color scale in RGB units, given by the bar. Rows do not share it: the three scenes sit at 0.26, 0.12 and 0.16 RGB units. Row labels give the deployed view factor a and the oracle $a^\star$, both relative to their batch mean, so that the two are comparable. Inside a single view the two half-width maps are close to affine-related, which is Lemma 3 once more, so they are meant to look alike. The comparison between them is the printed mean half-width $\overline{w}$, not the picture. The last column marks the pixels our interval misses. All three views clear the 90% within-view target, at 93.1%, 96.5% and 99.2%.

Three things in the figure are worth stating. First, the interval is everywhere much wider than the typical residual. It has to be: it must cover 90% of the pixels, so its width is set by the upper tail of the error and not by the average. Second, the misses are not spread evenly. They concentrate on thin structure and high-frequency texture, which is where a Gaussian primitive renders an edge as a soft ramp. Third, on the vanilla-backbone view the two scales happen to give the same mean half-width. That is a property of this one view, not of the scene: over the whole scene the full factorization saves 26.1% against a constant scale (Table 11).

D.10 Per-scene results behind Table 1

Table 13 gives the 13 individual scenes that produce the means in Table 1, together with the two paired differences. Neither aggregate is driven by a few scenes. Our scale beats 3DGS-U on 11 of 13, and the two losses are kitchen (-6.4) and train (-1.6). Against the ensemble the picture is genuinely mixed, which is why we claim parity rather than an advantage. We win on 7 of 13, and the column holds both the largest win in the table (bonsai, $+24.7$) and the largest loss (drjohnson, -19.5). The ensemble is very strong on drjohnson and room, while our scale is stronger on bonsai and train.

Table 13: Per-scene width saving (%) against the constant scale, for all 13 scenes of Table 1. The last two columns are paired differences on the same scene. View-event coverage stays within 90.0–93.2% across all 13 scenes and all five scales. Means and paired differences are computed before rounding.

Scene	3DGS-U	GAVIS	Ensemble	Ours	Ours−3DGS-U	Ours−Ens.
truck	14.6	−0.6	10.7	17.4	2.8	6.7
train	18.8	−2.3	5.7	17.2	−1.6	11.5
drjohnson	12.3	4.8	34.7	15.2	2.9	−19.5
playroom	13.3	11.0	25.4	22.4	9.1	−3.0
bicycle	15.7	−0.2	13.7	18.2	2.5	4.5
bonsai	35.3	−5.9	18.0	42.7	7.4	24.7
counter	18.3	1.9	20.4	24.0	5.7	3.6
flowers	19.4	6.7	20.5	21.0	1.6	0.5
garden	20.8	8.9	28.2	21.8	1.1	−6.4
kitchen	13.5	−6.3	4.9	7.1	−6.4	2.2
room	19.9	23.2	39.4	35.1	15.2	−4.3
stump	11.3	4.7	22.0	19.3	8.0	−2.7
treehill	13.0	8.8	29.1	26.2	13.1	−2.9
Mean	17.4	4.2	21.0	**22.1**	4.7	1.2
Wins					11/13	7/13
Sign p					0.0225	1.0

D.11 Cross-family transfer and predictive-center controls

Table 14 reports the aggregate transfer results omitted from the main text. In the same-family setting, the view regressor is learned on eight source scenes and tested on five unseen Mip-NeRF 360 scenes. In strict family holdout it is learned only on two Tanks & Temples and two Deep Blending scenes, then used without refitting on all nine Mip-NeRF 360 scenes. All hyperparameters remain source-selected.

Table 14: Cross-scene transfer with a shared single-model center. Savings are relative to a constant scale. Gap and wins compare VSCP with the source-selected ensemble scale.

Transfer	Ours (VSCP)	Ensemble scale	Gap [95%]	Wins	Sign p
Same family (5 scenes)	23.3	18.8	4.5 [−1.9, 15.2]	3/5	1.0
Strict family holdout (9)	**23.5**	21.5	2.0 [−3.0, 8.5]	4/9	1.0

Table 15: Per-scene width saving (%) in strict leave-Mip-NeRF-360-out meta-training. The source pool contains only Tanks & Temples and Deep Blending.

Scene	a only	$\widetilde{b}$ only	Full	Ensemble scale	Oracle
bicycle	15.2	4.6	20.4	13.9	41.1
garden	12.4	15.3	21.5	27.2	38.9
room	23.9	14.9	34.3	38.5	54.1
counter	13.4	12.0	24.2	20.0	37.7
flowers	7.4	17.1	20.2	20.3	29.5
stump	11.8	5.8	18.4	21.8	37.0
treehill	14.2	6.4	22.8	28.9	40.6
bonsai	29.9	23.3	42.5	18.3	57.5
kitchen	−4.8	10.9	7.2	4.8	38.3
Mean	13.7	12.3	**23.5**	21.5	41.6

A four-scene source pool of the same size that includes Mip-NeRF 360 gives a 24.1% saving, only 0.6 points above strict holdout. Thus the observed domain shift has little efficiency cost, while target-scene View-CP remains the safety layer. The strict predictor beats the constant scale on all nine target scenes ($p = 0.0039$) and matches an ensemble scale that requires ten models trained on each target scene. This comparison concerns how the view-difficulty signal is obtained, not a claimed width advantage over the ensemble.

The conclusion is unchanged when both scales use the ensemble mean as their predictive center. Under strict holdout, VSCP saves 22.3% and the native ensemble scale 19.3%; the paired gap is 3.0 points [−2.8, 10.2], with 5/9 wins and $p = 1.0$. In the same-family setting the gap is 5.5 points [−1.8, 17.0], with 4/5 wins and $p = 0.375$. When ten models are available, the ensemble mean and the VSCP scale can therefore be combined.

D.12 Scene-level strict-holdout results

The full method beats the constant scale on all nine scenes, with exact two-sided sign-test $p = 0.0039$. That is the smallest value attainable at $n = 9$, so it records a clean sweep of directions and not an effect size. The full method beats the ensemble scale on 4/9 scenes; the paired mean gap is 2.0 points [−3.0, 8.5] and $p = 1.0$. In the five-scene same-family test it beats the constant on 5/5 scenes, where the floor is $p = 0.0625$. Scene count, rather than repeated split count, limits all of these tests.

E Limitations

The guarantee is marginal over exchangeable views, not conditional coverage for every camera pose, and ordered camera paths can break exchangeability. The method needs labeled target-scene views that are held out. Using them to train 3DGS instead would need a different calibration argument. Only one held-out family is large enough to be useful, and family is confounded with bounded versus unbounded capture. The 13-scene results use FastGS. The vanilla-densification study covers three scenes and changes only the density rule, so it is not the official vanilla implementation. Our GAVIS row evaluates the released visibility field under a shared rasterizer, not its Bayesian uncertainty head. Systems such as Horseshoe Splatting change the center and the training objective, so we discuss them rather than place them in Table 1. Finally, our sets are axis-aligned RGB boxes with a shared radius. Richer color geometry, dynamic scenes, and time-correlated calibration units are out of scope.