Reliability-Weighted Averaging for Real-Time Neural Incident Radiance Caching

Dmitrii Klepikov
Karlsruhe Institute of Technology
Karlsruhe, Germany

Johannes Hanika
Karlsruhe Institute of Technology
Karlsruhe, Germany

Carsten Dachsbacher
Karlsruhe Institute of Technology
Karlsruhe, Germany

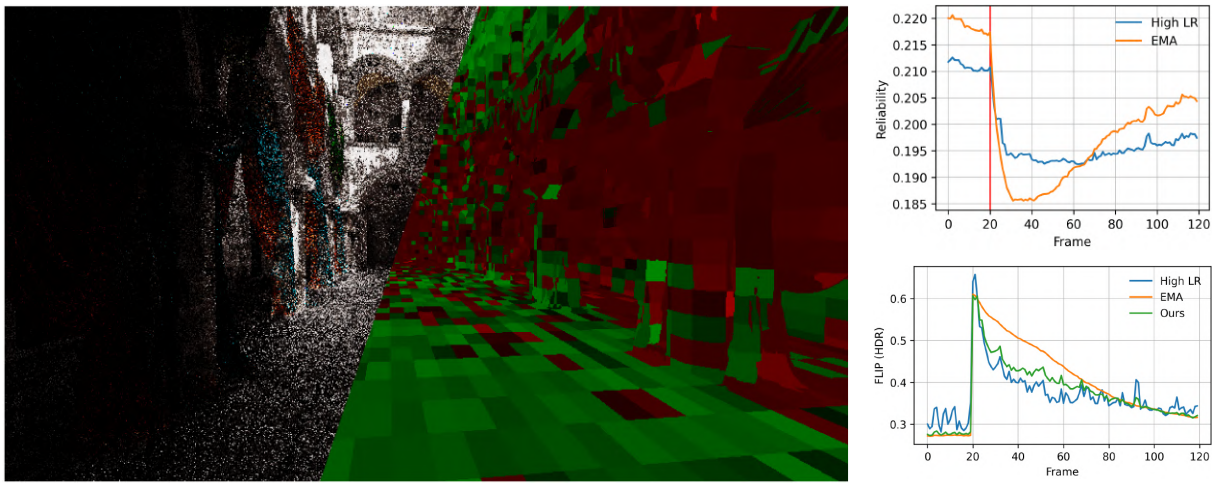


Figure 1: Two online predictors are run in parallel: a fast, adaptive predictor and its EMA-smoothed copy. Per-region reliabilities, computed from target–prediction statistics without extra rays, decide at query time whether to *switch* or *blend*. **Left:** preference map after the light rotation. Green indicates regions where the fast High-LR predictor is preferred, while red indicates regions where EMA is preferred. After a light rotation, the method switches to the High-LR predictor in regions affected by the lighting change. **Top-right:** reliability vs. frame index. Our method favors the High-LR model after the change and then returns to EMA in steady periods. **Bottom-right:** FLIP (error, lower is better). Our method adapts as quickly as the fast model immediately after the change and recovers the stability of EMA afterward, reducing flicker while matching steady-state quality.

ABSTRACT

Neural radiance caches remain challenging to use in highly dynamic scenes, such as when geometry or light sources move. The cache must adapt quickly to reduce bias after changes while preserving temporal stability in static regions to avoid flicker. We introduce a reliability metric based on exponentially weighted correlation between the cache prediction and the Monte Carlo target. The metric identifies regions where adaptation is needed and regions where a low-variance prediction is preferable. It does not require tracing additional rays. Using this signal, we construct a reliability-weighted scheme for real-time neural radiance caching. For each path-traced cache query, the method averages two global illumination predictors. One predictor adapts rapidly, while the other is temporally smoothed using exponentially averaged weights. Because the reliability weights are resolved spatially, the method responds quickly in changing regions while remaining stable elsewhere. At equal samples per pixel, our method reduces flicker in dynamic scenes, as measured by SMAPE and FLIP. Compared to the exponentially averaged neural incident radiance cache baseline, it also reduces temporal lag. This is achieved through data-driven per-query blending between the current predictor and the temporally smoothed predictor.

Keywords

Neural radiance caching, incident radiance, real-time path tracing, temporal stability, reliability estimation, global illumination.

1 INTRODUCTION

Neural radiance/incident radiance caches (NRC/NIRC) [9, 2] accelerate path tracing using online learning of the (incident) radiance field. Temporal stability is typically enforced by evaluating an exponential moving average (EMA) of the network weights. Adaptation speed is controlled by the learning rate (LR). A high learning rate adapts quickly but leads to flicker. Strong EMA stabilizes but introduces temporal lag after scene changes. A single global setting cannot satisfy both.

This work builds on a long line of (ir-)radiance caching methods, which reuse sparse lighting estimates through spatial interpolation [11]. Earlier cache methods also introduced gradients, adaptive placement, and clamping to improve interpolation robustness [10, 5].

We retain the NRC/NIRC abstraction and the EMA method and address the residual *temporal lag*. We run two predictors of the same cache in parallel: the training predictor and its EMA copy. At query time, we decide which predictor to trust. The decision is data-driven: a per-region reliability is computed from the same Monte Carlo training samples, with no extra rays.

We evaluate two decision rules at equal time: a binary switch and a convex blend. Our contributions are: First, we propose an online, per-region reliability estimate derived from exponentially weighted target-prediction statistics. Second, we use this estimate for query-time blending between the training network and its EMA version. Third, we analyze controlled scene changes by separating adaptation speed and temporal stability. Adaptation speed is evaluated using per-frame FLIP error [1], a perceptual image error metric for rendered images. Temporal stability is evaluated using frame-to-frame SMAPE (symmetrical MAPE) [6, 3].

2 PREVIOUS WORK

NRC [9] approximates the exitant radiance field with a tiny fully-fused multi-layer perceptron (MLP) trained online. Monte Carlo path samples terminate in the cache by querying the MLP. A sparse set of paths is extended by a few vertices and the resulting estimator is used for training. MLP evaluation uses an exponential moving average (EMA) of weights to suppress flicker from noisy training samples. EMA is known to improve prediction consistency and reduce optimization noise by averaging over stochastic training updates [7].

Stronger EMA improves temporal stability but also introduces temporal lag after scene changes. NIRC [2] follows the same idea but approximates incident radiance and uses a two-level Monte Carlo estimator with a balanced termination heuristic to control bias at fewer bounces. Like NRC/NIRC, we use a multiresolution hash encoding [8].

Classical and recent radiance-caching methods often rely on explicit spatial or spatio-directional data structures to decide where cached information is valid. For example, path guiding methods learn spatial and directional radiance distributions using adaptive structures such as SD-trees. Recent real-time radiance caches instead use surface-based or wavelet-based cache representations. In contrast, NRC and NIRC replace explicit cache-point interpolation with an online neural predictor.

3 METHODOLOGY

Following NRC [9], we evaluate the cache with an exponential moving average (EMA). The rapidly updated MLP weights lead to flicker caused by noisy targets and multiple gradient steps per frame. We train only the High-LR weights W_t and maintain a second set of dampened weights $\bar{W}_t$ using EMA

$$\bar{W}_t = \alpha \bar{W}_{t-1} + (1-\alpha) W_t.$$

This average is only used for inference. For inference we evaluate *both* predictors, $f_{\text{cur}}(\cdot; W_t)$ and $f_{\text{EMA}}(\cdot; \bar{W}_t)$, and blend them per query. Determining good blend weights that are adaptive both in time and space is the topic of the rest of this paper.

Relation to optimizer tuning.

Adam optimizer [4] hyperparameters, such as the learning rate and momentum terms, control the update dynamics of a single global predictor. They therefore select only one point on the stabilityâadaptation trade-off. Our method instead keeps two predictors with different temporal characteristics: High-LR and EMA. It blends between them locally at inference time. A single global optimizer setting cannot provide this behavior.

3.1 Per-region accumulators

To achieve spatial adaptivity, we maintain a data structure to store a few values indexed by world space position. We define a single-resolution, axis-aligned hashed voxel grid over the scene bounding box B. We map a query position x to a grid index in two steps. First, we linearly normalize x to the scene bounding box B. Then, we apply integer truncation. Hash collisions are ignored. Each active cell stores statistics from training samples (y, f), where $y \in \mathbb{R}^3$ is the Monte Carlo

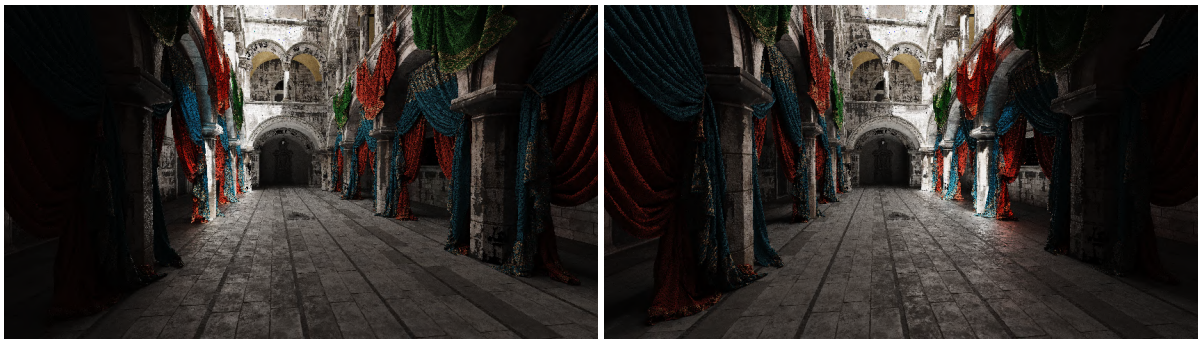


Figure 2: Scene change introduced by abruptly rotating the dominant directional light.

estimator for the target radiance (in RGB) and $f \in \mathbb{R}^3$ the MLP prediction of the same quantity (either f_{cur} or f_{EMA}). The EMA uses $\lambda \in (0,1)$ as parameter.

For every cell, we track the smoothed first moments M^y and M^f. These correspond to the Monte Carlo estimator and the prediction, respectively. We also track the corresponding second moments Q^y and Q^f. Additionally we store $C^{y,f}$, related to the covariance of y and f. All these statistics are per color channel and updated online as follows:

$$M^y \leftarrow \lambda M^y + (1-\lambda)\, y, \qquad Q^y \leftarrow \lambda Q^y + (1-\lambda)\, y^2,$$
$$M^f \leftarrow \lambda M^f + (1-\lambda)\, f, \qquad Q^f \leftarrow \lambda Q^f + (1-\lambda)\, f^2,$$
$$C^{y,f} \leftarrow \lambda C^{y,f} + (1-\lambda)\, yf.$$

From them, we derive exponentially averaged approximations of the means μ and variances s as

$$\mu^* = M^*, \qquad s^* = \left(Q^* - (M^*)^2\right)_+, \tag{1}$$

and the covariance

$$c^{y,f} = C^{y,f} - M^y M^f. \tag{2}$$

We keep a single set of target moments (M^y, Q^y) shared by both predictors so that they are scored against the same targets, and separate $(M^f, Q^f, C^{y,f})$ per predictor. Markers are disabled during an initial warm-up of K updates per grid (we use $\lambda \approx 0.9$ and $K = 128$ in practice). All statistics are obtained from rays already traced for training, and no additional rays are used.

The reliability grid is not intended to represent radiance itself. It only stores a coarse confidence signal for choosing between two already available predictors. Therefore, a single-resolution grid is sufficient when the main radiometric change is spatially coherent, such as the direct and indirect illumination changes caused by our light rotation. However, this approximation can mix unrelated samples in cells that contain depth discontinuities, thin geometry, strong normal variation, or view-dependent glossy reflections. In such cases, the correlation may average over incompatible transport paths and produce less decisive blend weights.

3.2 Reliability

Our reliability measure is based on the exponentially averaged correlation: it is large when the predictor matches the target, and drops under dynamic scene changes. More precisely, we use

$$\bar{r} = (r_{\text{red}} + r_{\text{green}} + r_{\text{blue}})/3$$

with

$$r = \frac{c^{y,f}}{\sqrt{(s^y + \varepsilon)(s^f + \varepsilon)}}. \tag{3}$$

We compute the blend weight σ based on the two correlations between reference values and fast adapting $\bar{r}_{\text{cur}}$ as well as smoothed $\bar{r}_{\text{EMA}}$ evaluation, respectively:

$$\sigma = \frac{1}{1 + \exp\left(-\kappa(\bar{r}_{\text{cur}} - \bar{r}_{\text{EMA}})\right)},$$

where the gain $\kappa > 0$ controls decisiveness. We use $\kappa = 4$. For a query q in a specific grid cell we evaluate a mixture of predictions

$$\hat{L} = (1-\sigma)\, f_{\text{EMA}}(q) + \sigma\, f_{\text{cur}}(q).$$

4 EVALUATION

We evaluate three methods under identical rendering and training budgets: *EMA-only*, *High-LR only* (adaptive MLP without temporal smoothing), and *Ours* (reliability-weighted blending). All methods use the same scene representation, hash-grid encoding, optimizer, training targets, and samples per pixel.

As a starting point we pre-train the cache MLP over approximately 5000 frames. During our experiment, we study a controlled dynamic change in three scenes by rotating the dominant directional light (Figure 2) at frame $t = 20$.

We measure adaptation speed, temporal stability, and the behavior of the proposed reliability and mixing. Adaptation speed is quantified using FLIP (HDR) error relative to a path-traced reference. Temporal stability is measured using frame-to-frame SMAPE [6, 3] between consecutive frames. We analyze the reliability signal by tracking its spatial distribution (preference maps) and its temporal evolution (mean reliability per frame).

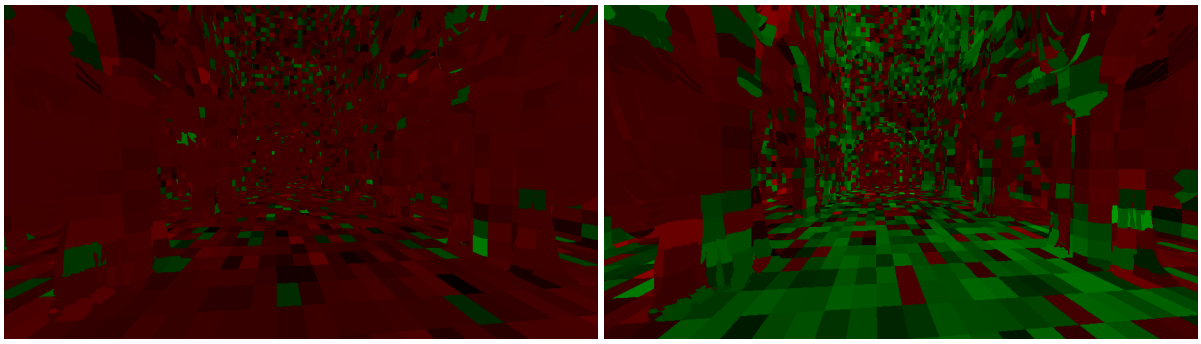
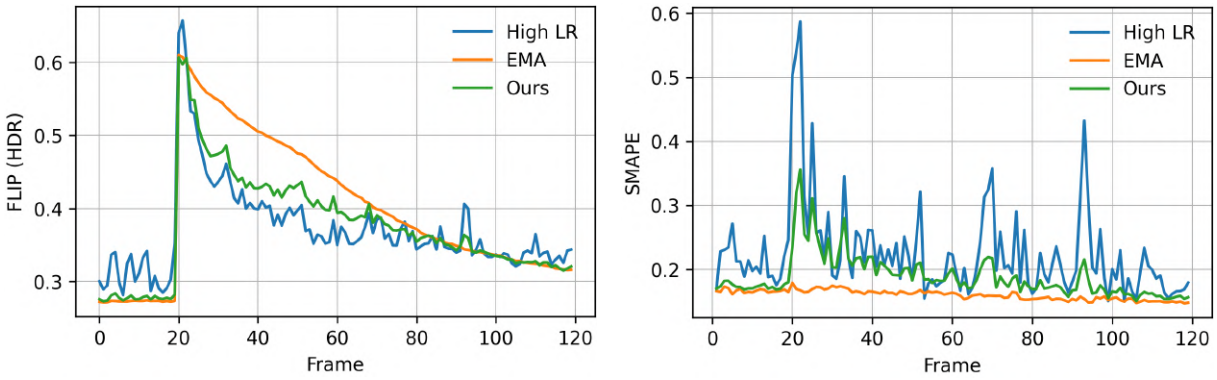


Figure 3: **Green:** prefer high-LR (adaptive), **Red:** prefer EMA (stable). Left/right images correspond to pre/post light rotation in Fig. 2.



FLIP (HDR)

Frame-to-frame SMAPE [6, 3] (LDR, t vs. $t-1$)

Figure 4: Quality vs. frames **Left:** FLIP, perceptual error to reference (lower is better). **Right:** SMAPE, frame-to-frame change, i.e. flicker (lower is better).

4.1 Reliability over time

Across scenes, reliability maps favor EMA in steady periods and switch to the High-LR in regions undergoing the largest radiance change (Figure 3).

Top-right graph in Fig. 1 plots mean reliability over time. Before the change, EMA has higher mean reliability and therefore dominates the final output. Immediately after the change, EMA reliability drops more sharply than the reliability of the High-LR predictor. Our method shifts toward the latter and then gradually returns as the scene restabilizes, matching the intended behavior: quick handover during change, and a return to stable EMA afterward.

4.2 Quantitative results

We evaluate adaptation speed with FLIP (HDR) and temporal stability with a frame-to-frame SMAPE-based flicker metric. For consecutive frames I_t and I_{t-1}, we compute

$$\mathrm{SMAPE}(t,t-1) = \frac{1}{|\Omega|}\sum_{p\in\Omega}\frac{2\,|I_t(p)-I_{t-1}(p)|}{|I_t(p)|+|I_{t-1}(p)|+\varepsilon}.$$

Lower SMAPE indicates smaller temporal change and hence less visible flicker.

All methods share the same ray budget, encoding, and optimizer. Our method adds one extra MLP evaluation (EMA) per query only. In Figure 4 we can see that, after the light rotation, *High-LR* and *Ours* reduce FLIP [1] faster than *EMA*. *High-LR* exhibits frequent spikes. Post-change, all methods spike in frame-to-frame SMAPE [6, 3]; *Ours* returns to low SMAPE faster than *High-LR* and remains near *EMA-only*.

4.3 Crop analysis across three scenes

In Fig. 5, 6, 7, for each scene we select two crops: **orange** denotes a mostly steady region (small radiometric change), and **blue** denotes a region with a strong post-change contrast shift. On more complex surfaces, such as regions with strong view dependence or high-gloss transport chains, the effect is mixed. In these cases, the reliability signal is weaker or noisier.

Crops were selected to be geometrically simple to produce a clearer and more interpretable reliability signal. In more complex regions (e.g., with strong view dependence or glossy reflections), the signal becomes noisier due to weaker or more variable supervision.

4.4 Runtime

We report FPS, averaged over the evaluation window (frames $t = 0 \ldots T$) and measured at 1280×720 on

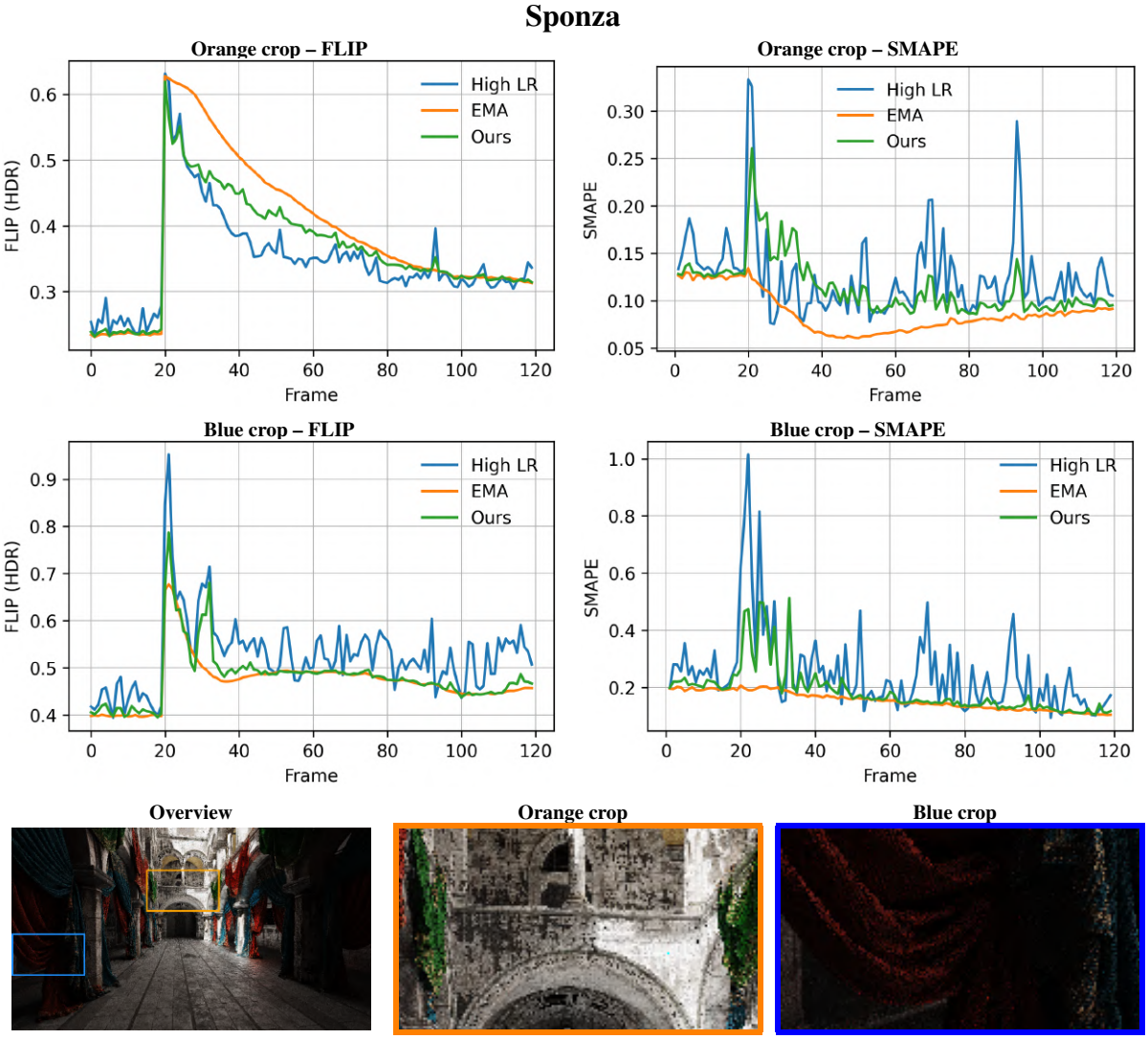


Figure 5: Error-map comparison for Sponza. FLIP and SMAPE are shown for the orange and blue crop regions. The bottom row shows the scene overview and the two corresponding crop insets.

NVIDIA RTX 3080. Table 1 summarizes the results across the three scenes. As expected from one additional MLP evaluation per query, *Ours* lies close to *EMA-only* by a nearly constant offset.

	High-LR	EMA	Ours
Sponza (ms / FPS)	13.52 / 74.0	15.37 / 65.1	15.83 / 63.1
Bistro (ms / FPS)	14.12 / 70.8	15.81 / 63.3	16.24 / 61.6
San Miguel (ms / FPS)	14.92 / 67.0	16.89 / 59.2	17.26 / 57.9

Table 1: Runtime at 1280×720 on NVIDIA RTX 3080.

5 CONCLUSION

We presented a reliability-weighted scheme for real-time neural (incident) radiance caching that combines a rapidly adapting predictor with its EMA-stabilized counterpart. A per-region, exponentially weighted correlation between Monte Carlo targets and predictions provides the reliability estimate. This reliability controls a per-query mixture of the two predictors. The method adds one extra MLP evaluation but does not require extra rays.

On lighting changes, the method reduces visual error (FLIP [1]) as quickly as a high-LR model immediately after change and returns to EMA-level flicker (SMAPE) once the scene restabilizes. The design is simple to integrate and reuses the NRC/NIRC stack (hash grid, optimizer, termination).

Limitations remain in regions where the reliability estimate receives weak or ambiguous supervision. In very dark or low-signal areas, the Monte Carlo target has high relative variance, so the correlation between target and prediction can be noisy despite exponential averaging and variance clamping. Long specular transport chains are also difficult because small changes in direction or visibility can produce large radiance changes that are only sparsely sampled. Finally, the single-resolution voxel grid can mix unrelated samples

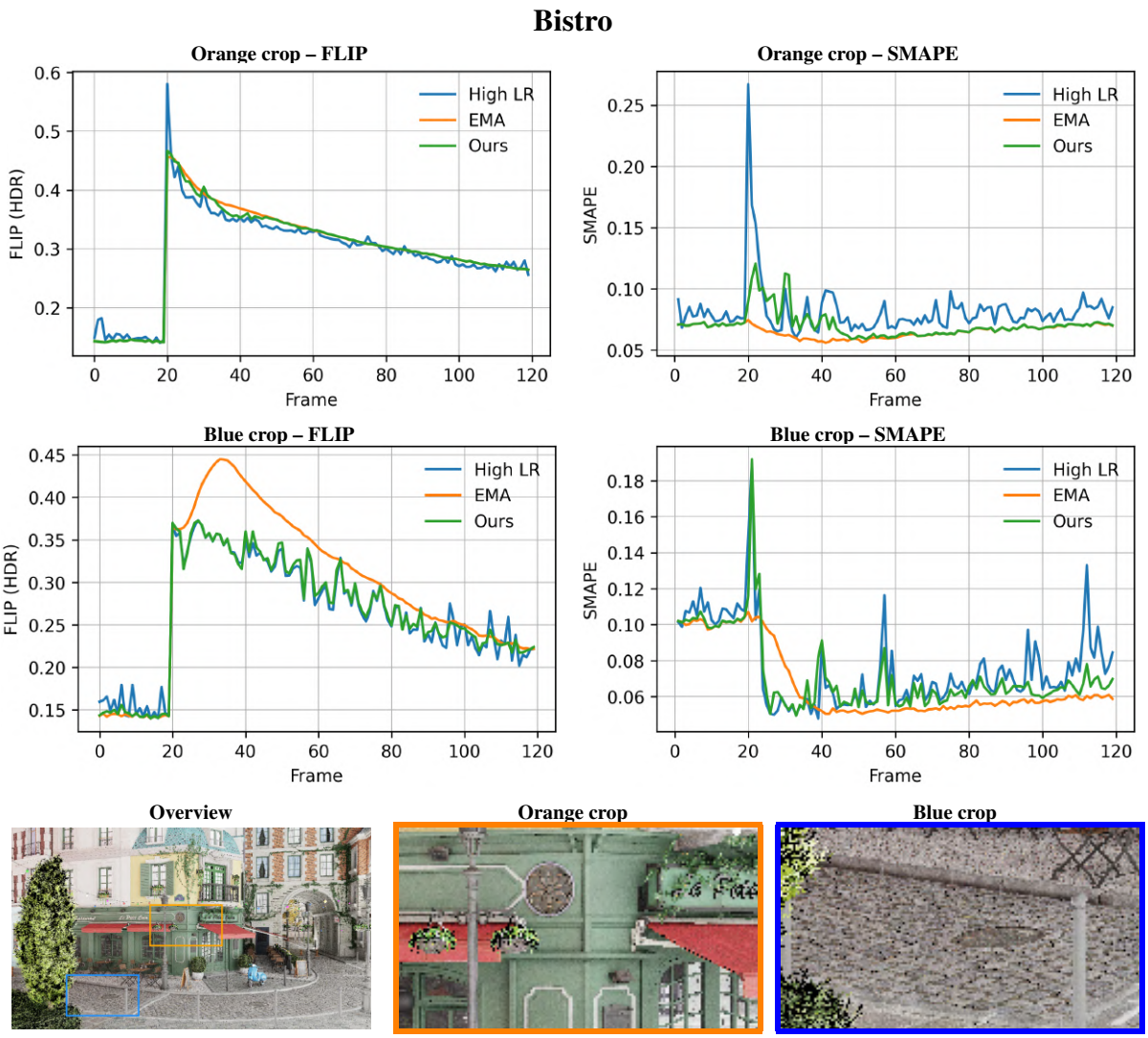


Figure 6: Error-map comparison for Bistro. FLIP and SMAPE are shown for the orange and blue crop regions. The bottom row shows the scene overview and the two corresponding crop insets.

near depth discontinuities, thin geometry, high normal variation, or glossy view-dependent surfaces. These cases may lead to less decisive or suboptimal blending weights.

Future work could make the reliability estimate more robust. Possible directions include variance-aware confidence bounds, robust correlation estimators, and adaptive sample allocation in low-reliability regions. Another direction is to use geometry-aware cells, for example by splitting statistics by normal, material roughness, depth, or visibility. Reliability could also drive decisions beyond blending, such as local sample allocation, optimizer scheduling, or selecting between different cache types.

6 ACKNOWLEDGEMENTS

The authors would like to thank the Intel Corporation for their support of this work.

7 REFERENCES

[1] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. FLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2):15:1–15:23, 2020.

[2] Mikhail Dereviannykh, Dmitrii Klepikov, Johannes Hanika, and Carsten Dachsbacher. Neural two-level monte carlo real-time rendering. *Computer Graphics Forum*, 44(2):e70050, 2025.

[3] Paul Goodwin and Richard Lawton. On the asymmetry of the symmetric mape. *International Journal of Forecasting*, 15(4):405–408, 1999.

[4] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

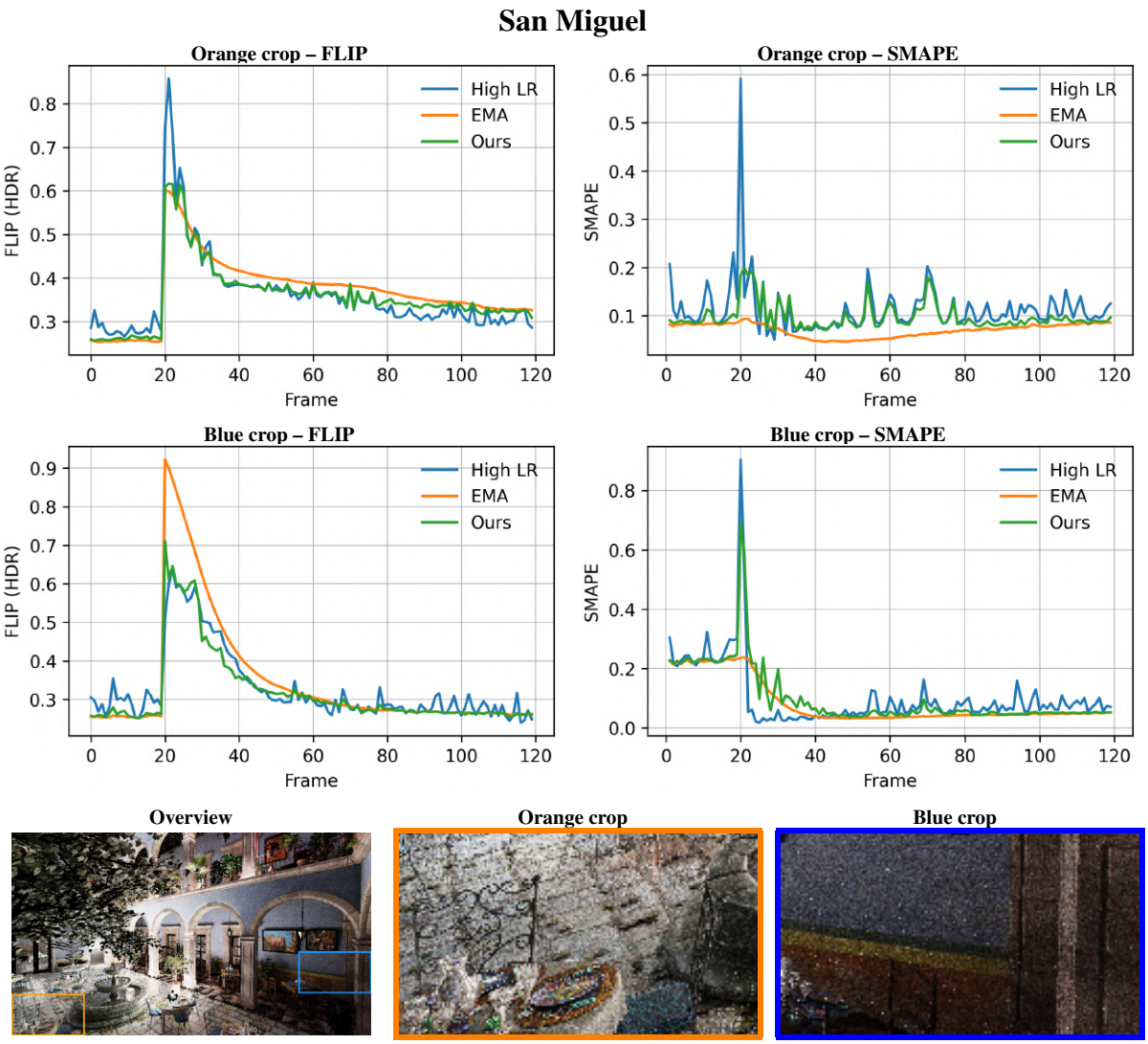


Figure 7: Error-map comparison for San Miguel. FLIP and SMAPE are shown for the orange and blue crop regions. The bottom row shows the scene overview and the two corresponding crop insets.

[5] Jaroslav Křivánek, Kadi Bouatouch, Sumanta N Pattanaik, and Jiri Zara. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. *Rendering Techniques*, 2006:127–138, 2006.

[6] Spyros Makridakis. Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, 9(4):527–529, 1993.

[7] Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*, 2024.

[8] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[9] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.*, 40(4):36:1–36:16, August 2021.

[10] Gregory J. Ward and Paul Heckbert. Irradiance gradients. In *Proceedings of the 3rd Eurographics Workshop on Rendering*, pages 85–98, Bristol, UK, 1992. Springer-Verlag. (PDF available from UCL Computer Science).

[11] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pages 85–92, New York, NY, USA, 1988. Association for Computing Machinery.