MARS: Multi-sample Allocation through Russian roulette and Splitting

JOSHUA MEYER, Saarland University, Germany
ALEXANDER RATH, Saarland University, Germany
ÖMERCAN YAZICI, Saarland University, Germany
PHILIPP SLUSALLEK, German Research Center for Artificial Intelligence, Germany and Saarland University, Germany

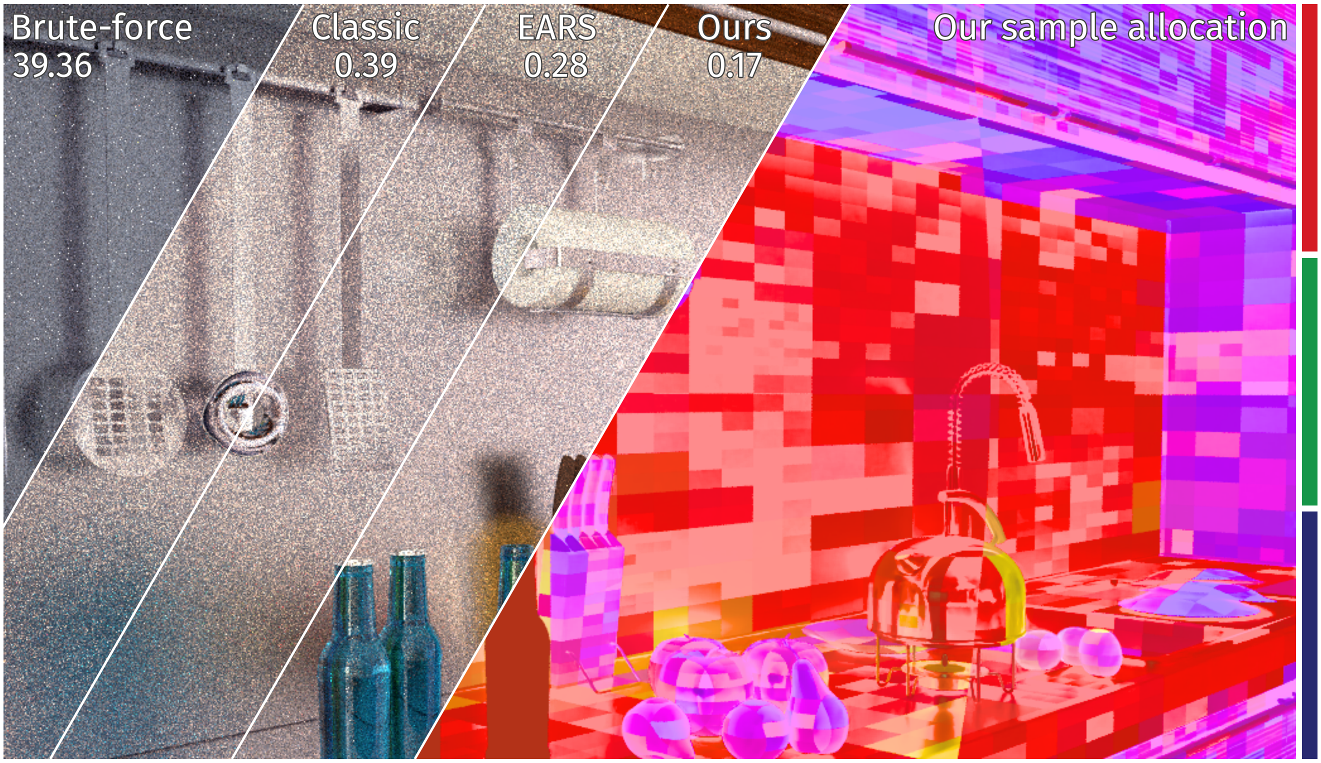


Fig. 1. The performance of multiple importance sampling can be greatly improved by choosing appropriate sample counts for each sampling strategy. Here, we compare the performance of our fixed-point approach applied to bidirectional path tracing (10 minute equal-time renders). Existing methods are either limited to per-pixel decisions ("brute-force" [Grittmann et al. 2022]) or to choosing the same amount of samples for each strategy ("EARS" [Rath et al. 2022]). Our method learns spatially varying sample counts individually for each strategy, visualized on the right, which increases performance by 65% over previous works.

Multiple importance sampling (MIS) is an indispensable tool in rendering that constructs robust sampling strategies by combining the respective strengths of individual distributions.Its efficiency can be greatly improved by carefully selecting the number of samples drawn from each distribution, but automating this process remains a challenging problem.Existing works are mostly limited to mixture sampling, in which only a single sample is drawn in total, and the works that do investigate multi-sample MIS only optimize the sample counts at a per-pixel level, which cannot account for variations beyond the first bounce.Recent work on Russian roulette and splitting has demonstrated how fixed-point schemes can be used to spatially vary sample counts to optimize image efficiency but is limited to choosing the same number of samples across all sampling strategies.Our work proposes a highly flexible sample allocation strategy that bridges the gap between these areas of work.We show how to iteratively optimize the sample counts to maximize the efficiency of the rendered image using a lightweight data structure, which allows us to make local and individual decisions per technique.We demonstrate the benefits of our approach in two applications, path guiding and bidirectional path tracing, in both of which we achieve consistent and substantial speedups over the respective previous state-of-the-art.

CCS Concepts: • **Computing methodologies → Ray tracing**.

Additional Key Words and Phrases: Russian roulette, splitting, multi-sample, multiple importance sampling

ACM Reference Format:
Joshua Meyer, Alexander Rath, Ömercan Yazici, and Philipp Slusallek. 2024. MARS: Multi-sample Allocation through Russian roulette and Splitting. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3680528.3687636

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

1 INTRODUCTION

Synthetically generating realistic images from virtual, mathematical descriptions is an essential aspect in various fields such as multimedia, gaming, and advertisement. This is achieved by reconstructing the image from a sample of admissible light paths, which are commonly obtained through unidirectional or bidirectional path tracing.

Despite decades of research, there is no single sampling distribution that robustly captures all effects. Instead, contemporary works employ a blend of multiple strategies to attain completeness, using a technique known as *multiple importance sampling* (MIS). Many works have explored how to weight strategies in an MIS combination, but surprisingly little attention has been devoted to determining the optimal number of samples to assign to each technique in a (bidirectional) path tracing context. Although attempts have been made to address this question, none have provided entirely satisfactory solutions. In practice, sample counts are often steered by user intervention.

We present a novel theory for MIS sample allocation. By considering the local variances and costs of samples, our approach efficiently distributes samples among various strategies, leading to significantly increased rendering efficiency. Moreover, we validate our theory by applying it to two rendering algorithms, demonstrating its practical viability. In summary, our contributions are:

- We derive a novel theory for multi-sample MIS budget allocation, based on the theory of fixed-point iterations (section 3)
- We show how to apply our theory in the popular context of path guiding, achieving noticeable speedups over previous works (section 4)
- Additionally, we accelerate bidirectional path tracing with our approach and demonstrate equally substantial speedups (section 5)

Our Mitsuba [Jakob 2010] implementation of the two applications is publicly available at https://github.com/woshicado/mars.

2 PREVIOUS WORK

Light transport. Almost 40 years ago, Kajiya [1986] described how light is propagated within a scene , which can succinctly be written as the path integral [Veach 1997]

$$I_{\text{px}} = \int_{\mathcal{P}} f_{\text{px}}(\bar{x})\,\mathrm{d}\bar{x},$$

where a pixel's value I_{px} is given by the integral of the contribution function f_{px} over the space $\mathcal{P}$ of all light transport paths connecting the pixel px to a light source. Due to the high dimensionality, many discontinuities, and overall complexity, obtaining a general analytical solution is impractical. Therefore, Monte Carlo integration is typically used to compute this integral. *Importance sampling* (IS) augments the integration process by sampling from a distribution similar to the integrand in order to reduce variance.

Multiple importance sampling. Often, there is not a single distribution that estimates the target sufficiently well over the whole domain. *Multiple importance sampling* (MIS) [Veach and Guibas 1995b] can be utilized to combine several sampling distributions strategically to leverage their respective strengths. To that end, a *weighting heuristic* approximates which strategy performs well within which subdomain to weight them against each other, which has been explored in great detail by previous works [Kondapaneni et al. 2019; Veach and Guibas 1995b].

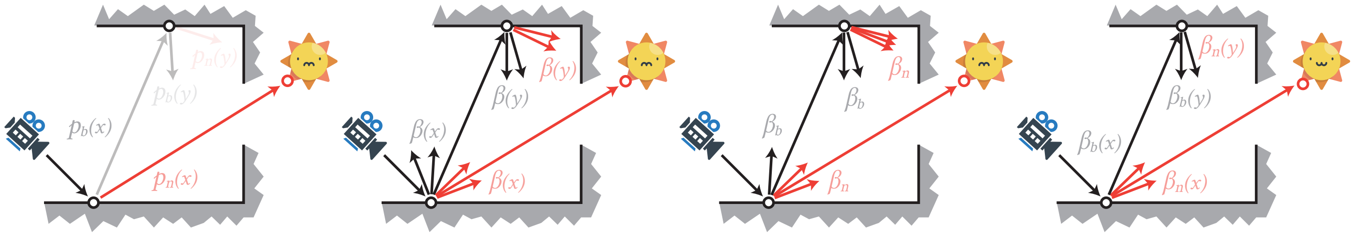
Path tracing. Forward path tracing [Kajiya 1986] is the primary choice for simulating light transport due to its flexibility and extensibility. Starting from the camera, a random walk through the scene is performed to construct a path sample. At each intersection, the reflection properties (*bidirectional scattering distribution function*, BSDF) are sampled to steer the random walk. Additional samples, such as direct connections to light sources (*next event estimation*, NEE), are commonly incorporated through MIS. However, even with sophisticated NEE enhancements, like *manifold next event estimation* [Hanika et al. 2015; Zeltner et al. 2020], forward path tracing alone does not suffice to efficiently render many effects.

Bidirectional methods. In *bidirectional path tracing* (BDPT) [Veach and Guibas 1995a], an additional random walk from light sources is performed. Using MIS, it is possible to connect all pairs of intersections between the camera and light path to obtain many complete path samples [Lafortune and Willems 1993]. It has been shown that connecting each camera path vertex to a fixed number of light subpath vertices can impact the rendering performance positively [Nabata et al. 2020; Popov et al. 2015].

Path guiding. Another popular approach to improve upon classical path tracing is *path guiding* [Jensen 1995; Müller et al. 2017; Vorba et al. 2014]. Using previously acquired samples, it is possible to identify important directions and build adaptive sampling densities accordingly. To increase robustness, guiding is jointly used with standard forward path tracing and NEE in an MIS combination. Subsequent studies explored further improvements to increase robustness for common effects [Rath et al. 2023; Ruppert et al. 2020].

Mixture sampling ratios. The ratios with which techniques are picked have a profound impact on the efficiency of MIS, but existing works on automating those share many limitations. Most works focus on direct illumination only, optimizing the ratio between BSDF and NEE samples on a per-pixel basis, for example through second-order approximations [Lu et al. 2013] or iterative Newton-Raphson root finding [Sbert et al. 2019; Szirmay-Kalos and Sbert 2022]. Murray et al. [2020] introduce a simple heuristic to extend these works to global illumination. A more powerful alternative to learning ratios per pixel is to learn them per region of space. Vorba et al. [2019] use gradient descent to compute sampling ratios between BSDF and path guiding, allowing to optimize for variance or KL divergence. Rath et al. [2020] improved this method by considering image variance instead of local variance alone.

An important limitation shared by these approaches is that they only optimize mixture sampling (also known as *one-sample MIS*), i.e., the random walk in path tracing is ever only continued with a single sample. Consequently, they cannot take advantage of optimizing efficiency at a particular intersection without resampling the whole path prefix. They are also limited to optimizing efficiency locally since the ratios are optimized independently.



(a) One-sample MIS (b) Russian roulette and splitting (c) Per-pixel multi-sample MIS (d) **Our approach**

Fig. 2. Existing sample allocation works share three common limitations. **(a)** One-sample MIS randomly picks a single strategy to continue the path, which can result in poor efficiency since path prefixes are not re-used. **(b)** RRS can re-use path prefixes, but chooses the same sample count for each sampling strategy independent of whether the strategy contributes or not. **(c)** Existing multi-sample MIS approaches are limited to using the same locally-unaware sample accounts along the entire path. **(d)** Our method generalizes all of these methods and chooses spatially varying sample counts for each strategy individually.

Russian roulette and splitting. An integral component of path tracing is Russian roulette and splitting (RRS) [Arvo and Kirk 1990]. RR entails prematurely terminating paths based on their current throughput while splitting increases the number of subpaths at a given intersection. More advanced approaches take into consideration additional metrics. Vorba and Křivánek [2016] propose a method that factors in expectations of future contributions. Further, EARS [Rath et al. 2022] employs a fixed-point iteration scheme to optimize RRS factors for efficiency, based on the theoretical optimal derivations of Bolin and Meyer [1997]. While these methods can optimize for global efficiency, they only determine the total number of samples taken at each intersection and have no control over how they are allocated to individual strategies.

Multi-sample allocation. In *multi-sample MIS*, each distribution gets to draw an arbitrary amount of samples. This combines the strengths of mixture sampling (i.e., determining the ratios across techniques) and RRS (i.e., determining the total number of samples). However, we show that combining existing works from both domains, even if they are individually optimal, is insufficient to achieve optimal performance. Hence, they must be optimized jointly.

By modeling the expected computational cost of an MIS combination, Grittmann et al. [2022] globally optimize BDPT efficiency using a simple brute-force search through a set of candidate sample counts. Due to the curse of dimensionality, this technique is limited to optimizing global parameters (such as light path counts) or boolean per-pixel decisions.

Our approach (fig. 2) similarly optimizes the efficiency of multi-sample MIS combinations but does so at a much higher granularity of optimizing individual sampling strategies with spatially varying sample counts. This is enabled through a fixed-point scheme similar to EARS [Rath et al. 2022], which we generalize for MIS.

3 THEORY

In the following, we derive an efficiency-aware multi-sample MIS budget allocation strategy. We approach this problem by formulating a general model, analyzing its properties, and deriving a fixed-point scheme. In this section, we outline the key equations and arguments necessary to follow the idea. A full derivation can be found in the supplementary material.

The model. In multi-sample MIS, we are interested in finding an integral value $I = \int_{\mathcal{X}} f(x)\,\mathrm{d}x$ by combining estimates from n_t different sampling strategies with varying sample counts β_t

$$\langle I\rangle = \sum_{t=1}^{n_t} \frac{1}{\beta_t} \sum_{s=1}^{\beta_t} \langle I_t(x_{t,s})\rangle = \sum_{t=1}^{n_t} \langle I_t(x_{t,\cdot}); \beta_t\rangle, \tag{1}$$

where $\langle I; \beta\rangle$ denotes a secondary estimator for I with β samples, and $\langle I_t\rangle$ denotes the primary estimator of technique t.

An important aspect is that estimators often capture different, possibly disjoint parts of the integrand. In path tracing, for example, we commonly have two subintegrands: direct light (estimated by NEE and BSDF sampling), and indirect light (only estimated by BSDF sampling). We address this by expressing the integrand as a sum of subintegrands $f = \sum_{i=1}^{n_i} f_i$ and perform MIS weighting thereof within the primary estimators $\langle I_t\rangle$

$$\langle I_t(x)\rangle = \sum_{i=1}^{n_i} \frac{f_i(x)}{p_t(x)} w_{it}(x),$$

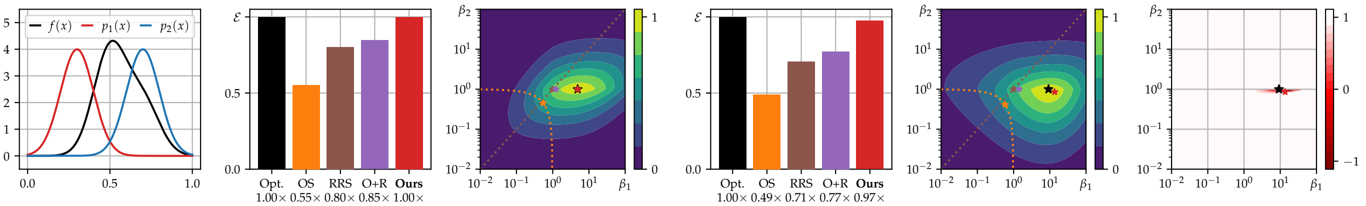
where p_t is the probability density function of technique t and w_{it} is the MIS weight with respect to all other methods that estimate integrand f_i.

Continuous sample counts. Instead of restricting ourselves to integer sample counts $\beta_t \in \mathbb{N}$, we shift the domain to $\mathbb{R}^+$ to make the problem continuous. Since, in reality, we can only perform an integer number of samples, we employ a stochastic rounding function

$$r(\beta) = \begin{cases} \lfloor\beta\rfloor + 1 & \text{with probability } \beta - \lfloor\beta\rfloor, \\ \lfloor\beta\rfloor & \text{otherwise,} \end{cases}$$

where $\lfloor\cdot\rfloor$ is the floor function. The estimator $\langle I\rangle$ still divides by the real-valued β_t, forming an unbiased stochastic sampling scheme. This is analogous to how previous works determine sample counts for RRS [Rath et al. 2022; Vorba and Křivánek 2016], but generalizes the concept to individual techniques instead of using the same splitting factor across all of them.

The objective. Our goal is to assign an optimal number of samples to each strategy, i.e., to find optimal β_t. Previous works have mostly focused on optimizing variance, which works well as long as the computational cost is largely unaffected by the choice of β_t, such as



(a) Integrand

(b) Budget-unaware balance heuristic

(c) Budget-aware balance heuristic

(d) Proxy error

Fig. 3. **(a)** We evaluate the performance of our model on a simple 1D example, in which a single integrand f is estimated by two techniques p_1 and p_2, the latter of which matches slightly better but is 30× as expensive as p_1. **(b)** We compare the efficiency of four approaches: Optimal mixture sampling ("OS"), optimal Russian roulette and splitting ("RRS"), combining the two ("O+R"), and our proxy model ("Ours"). Note that combining optimal mixture sampling and RRS does not yield optimal performance. Our model optimizes ratios and total sample counts jointly, resulting in optimal performance for budget-unaware MIS weights. **(c)** For budget-aware MIS weights, the optimum predicted by our model can deviate from the true efficiency optimum. **(d)** We investigate the error of our proxy by plotting the dot product of the true gradients and our proxy gradients ("1" indicates a perfect match, "-1" indicates opposing directions).

when optimizing one-sample MIS combinations. In multi-sample MIS the cost highly depends on the number of samples, and we must therefore consider the *efficiency* $\mathcal{E}$ [Pharr et al. 2016]

$$\mathcal{E}\left[\langle I\rangle\right] = \frac{1}{\mathbb{V}\left[\langle I\rangle\right]\mathbb{C}\left[\langle I\rangle\right]}.$$

Consequently, to optimize the efficiency, we need to predict the variance and cost for any choice of β_t.

Cost model. The cost of our multi-sample MIS estimator is given by the sum of all techniques' sample counts multiplied by the cost of taking a sample for this technique, i.e., $\mathbb{C}[\langle I\rangle] = \sum_{t=1}^{n_t} \beta_t \mathbb{C}_t + \mathbb{C}_\Delta$, where $\mathbb{C}_\Delta$ is a constant to incorporate overhead costs.

Variance model. We will assume that the techniques are uncorrelated, which allows us to express the variance of the MIS estimator as the sum of the variances of all techniques

$$\mathbb{V}\left[\langle I\rangle\right] = \sum_{t=1}^{n_t} \mathbb{V}\left[\langle I_t;\beta_t\rangle\right].$$

We can express the variance of the secondary estimator in terms of the variance and expected value of the primary estimator:

$$\mathbb{V}\left[\langle I_t;\beta_t\rangle\right] = \frac{1}{\beta_t}\mathbb{V}\left[\langle I_t\rangle\right] + \rho(\beta_t)\,\mathbb{E}^2\left[\langle I_t\rangle\right],$$

where $\rho(\beta) = \frac{(\beta-\lfloor\beta\rfloor)(\lceil\beta\rceil-\beta)}{\beta^2}$ results from stochastic rounding.

Optimization. We want to find the optimal β_t that maximize efficiency. To reduce clutter, our derivation pursues the equivalent goal of minimizing inverse efficiency, i.e., $\mathcal{E}^{-1}\left[\langle I\rangle\right]$. A necessary condition for a minimum with respect to β_t is that the partial derivatives of inverse efficiency equal 0. Since the complex shape of $\rho(\beta_t)$ makes analysis challenging, we resort to the same simplification employed by previous works [Rath et al. 2022]

$$\mathbb{V}\left[\langle I_t;\beta_t\rangle\right] = \begin{cases} \frac{1}{\beta_t}\mathbb{E}\left[\langle I_t\rangle^2\right] - \mathbb{E}^2\left[\langle I_t\rangle\right] & \text{if } \beta_t \le 1, \\ \frac{1}{\beta_t}\mathbb{V}\left[\langle I_t\rangle\right] & \text{otherwise.} \end{cases}$$

This simplification only affects the $\beta > 1$ case, where it drops the noise introduced by stochastic rounding to arrive at a convex expression for variance. An analysis of this simplification, along with alternative options for rounding, can be found in the supplemental.

Proxy model. Even for simple weighting functions such as the balance heuristic, the efficiency is not generally convex in β_t due to the terms the MIS weights introduce in the variance derivatives

$$\frac{\mathrm{d}\mathbb{V}[\langle I\rangle]}{\mathrm{d}\beta_t} = \begin{cases} -\frac{1}{\beta_t^2}\mathbb{E}[\langle I_t\rangle^2] + \sum_{k=1}^{n_t}\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{E}[\langle I_k\rangle^2]}{\mathrm{d}\beta_t} & \text{if } \beta_t \le 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}[\langle I_t\rangle] + \sum_{k=1}^{n_t}\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}[\langle I_k\rangle]}{\mathrm{d}\beta_t} & \text{otherwise.} \end{cases}$$

We instead resort to optimizing a proxy for efficiency, in which the MIS weight dependencies are dropped from the variance derivative

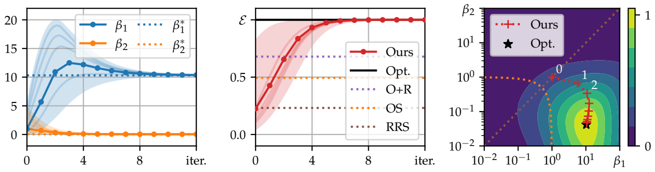
$$\frac{\mathrm{d}\mathbb{V}[\langle I\rangle]}{\mathrm{d}\beta_t} = \begin{cases} -\frac{1}{\beta_t^2}\mathbb{E}[\langle I_t\rangle^2] & \text{if } \beta_t \le 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}[\langle I_t\rangle] & \text{otherwise.} \end{cases}$$

This proxy is optimal for *budget-unaware* MIS weights (i.e., weights independent of β_t), but for *budget-aware* weights (i.e., weights containing the techniques' budgets β_t) its optimum can deviate from the true optimal solution. We perform two evaluations to verify the effectiveness of this proxy: First, we compare its derivatives and optima to the true efficiency for simple one-dimensional functions (see fig. 3 and further examples in the supplemental). Second, in our two rendering applications, we demonstrate that budget-aware weights with this proxy robustly outperform theoretically optimal allocations of budget-unaware weights.

Fixed point. The optimal β_t that minimize inverse efficiency in our proxy model are given by

$$\beta_t = \begin{cases} \beta_t^{\mathrm{RR}} = \sqrt{\frac{\mathbb{C}[\langle I\rangle]}{\mathbb{C}_t}\frac{\mathbb{E}[\langle I_t\rangle^2]}{\mathbb{V}[\langle I\rangle]}} & \text{if } \beta_t^{\mathrm{RR}} < 1, \\ \beta_t^{\mathrm{S}} \;\;= \sqrt{\frac{\mathbb{C}[\langle I\rangle]}{\mathbb{C}_t}\frac{\mathbb{V}[\langle I_t\rangle]}{\mathbb{V}[\langle I\rangle]}} & \text{if } \beta_t^{\mathrm{S}} \;\; > 1, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

An analytical solution to this equation is impractical because β_t appears non-linearly on the right-hand side as part of $\mathbb{C}[\langle I\rangle]$ and $\mathbb{V}\left[\langle I\rangle\right]$. Therefore, to find the root numerically, we employ a fixed-point iteration scheme similar to the work of Rath et al. [2022]. Our equation is of the same form and, thus, we can iteratively approach the root by using eq. (2) as an update function. In practice, this converges quickly after only a few iterations as depicted in fig. 4.



(a) Sample budgets (b) Efficiency (c) Landscape

Fig. 4. **(a)** We visualize the quick convergence of the sample allocations β_1 and β_2 towards their optimal values for a simple 1D example. The shaded area corresponds to trajectories of various initializations. **(b)** After 5 iterations we are close to the optimal efficiency. While training longer brings us closer, it yields diminishing returns. **(c)** We show the path our fixed-point iteration takes along the efficiency landscape.

3.1 Application to Path Tracing

We will now discuss how to apply our theory to path tracing, for which we extend our theory to multiple spatially varying estimators.

Primary estimator. In path tracing, we combine multiple strategies for estimating reflected radiance L_r at each intersection

$$\langle L_{\mathrm{r},t}(x,\omega_\mathrm{o})\rangle = \frac{\langle L_\mathrm{i}(x,\omega_\mathrm{i})\rangle\, B(x,\omega_\mathrm{i},\omega_\mathrm{o})\,|\cos\theta_\mathrm{i}|}{p_t(\omega_\mathrm{i}\mid x,\omega_\mathrm{o})}\, w_t(x,\omega_\mathrm{i},\omega_\mathrm{o}),$$

where technique t samples $\omega_\mathrm{i} \sim p_t$ and B denotes the BSDF.

Secondary estimator. Analogous to eq. (1), we define the multi-sample estimator for L_r

$$\langle L_\mathrm{r}(x,\omega_\mathrm{o})\rangle = \sum_{t=1}^{n_t} \frac{1}{\beta_t(\bar{x})} \sum_{s=1}^{\beta_t(\bar{x})} \langle L_{\mathrm{r},t}(x_s,\omega_\mathrm{o})\rangle = \sum_{t=1}^{n_t} \langle L_{\mathrm{r},t};\beta_t(\bar{x})\rangle,$$

where the sample budgets β_t are optimized for each individual technique t and path prefix $\bar{x}$. Instead of budgets that are constant across the image, we thereby adapt them locally.

Random walks. Our random walk for a pixel px begins with a ray sampled by the camera and then recursively applies the multi-sample estimator of L_r at each intersection

$$\langle I_\mathrm{px}\rangle = \frac{W_\mathrm{px}(x_0,\omega_{\mathrm{i},0})}{p(x_0,\omega_{\mathrm{i},0})} \langle L_\mathrm{r}(x_1,\omega_{\mathrm{o},1})\rangle.$$

Multiple estimators. Optimizing each pixel individually is insufficient to achieve a global optimum as this will oversample bright regions. Hence, we maximize the efficiency built from the average variance $\mathbb{V}_\mathrm{I}$ and average cost $\mathbb{C}_\mathrm{I}$ over the entire image instead [Rath et al. 2022]. To not assign too much weight to bright pixels, we similarly use the *relative* variance of each pixel, i.e., we divide by the pixel's ground truth I_px

$$\mathcal{E}^{-1} = \underbrace{\left(\frac{1}{N_\mathrm{px}} \sum_\mathrm{px}^{N_\mathrm{px}} \frac{\mathbb{V}\left[\langle I_\mathrm{px}\rangle\right]}{I_\mathrm{px}^2}\right)}_{\mathbb{V}_\mathrm{I}} \underbrace{\left(\frac{1}{N_\mathrm{px}} \sum_\mathrm{px}^{N_\mathrm{px}} \mathbb{C}[\langle I_\mathrm{px}\rangle]\right)}_{\mathbb{C}_\mathrm{I}}.$$

Since, in practice, I_px is not readily available, we discuss an approximation thereof in in section 4.1.

Optimization. Extended to multiple spatially varying estimators, our update function becomes

$$\beta_t(\bar{x}) = \begin{cases} \beta_t^\mathrm{RR} = \dfrac{T(\bar{x})}{I_\mathrm{px}} \sqrt{\dfrac{\mathbb{C}_\mathrm{I}}{\mathbb{C}_t} \dfrac{\mathbb{E}[\langle L_{\mathrm{r},t}\rangle^2]}{\mathbb{V}_\mathrm{I}}} & \text{if } \beta_t^\mathrm{RR} < 1, \\ \beta_t^\mathrm{S} \;= \dfrac{T(\bar{x})}{I_\mathrm{px}} \sqrt{\dfrac{\mathbb{C}_\mathrm{I}}{\mathbb{C}_t} \dfrac{\mathbb{V}[\langle L_{\mathrm{r},t}\rangle]}{\mathbb{V}_\mathrm{I}}} & \text{if } \beta_t^\mathrm{S} \;> 1, \\ 1 & \text{otherwise.} \end{cases}$$

Here, $T(\bar{x}_k)$ denotes the throughput weight of the path prefix

$$T(\bar{x}_k) = \frac{W_\mathrm{px}(x_0,\omega_{\mathrm{i},0})}{p(x_0,\omega_{\mathrm{i},0})} \prod_{j=1}^{k-1} \frac{B(x_j,\omega_{\mathrm{i},j},\omega_{\mathrm{o},j})\,|\cos\theta_{\mathrm{i},j}|}{p_{t_j}(\omega_{\mathrm{i},j}\mid x_j,\omega_{\mathrm{o},j})} \frac{w_{\mathrm{ind},t_j}(\bar{x}_j)}{\beta_{t_j}(\bar{x}_j)},$$

where t_j denotes the technique that was used at the j-th bounce, and w_ind is the MIS weight over all techniques that participate in estimating indirect light (i.e., lead to a secondary estimator being performed at x_j).

4 APPLICATION I: PATH GUIDING

We begin by demonstrating how our technique can be used in unidirectional rendering to combine three common strategies: BSDF sampling, next event estimation (NEE) and guided sampling.

4.1 Implementation

We test our method in a guided path tracer on top of the implementation of Müller et al. [2017] in Mitsuba [Jakob 2010]. Each completed training iteration of the guiding iteration marks one step of our fixed-point scheme. To compute the necessary variance and cost estimates, we build Monte Carlo estimates from the samples obtained throughout the iteration.

Local estimates. We share the spatio-directional tree used by guiding to store the local estimates needed by our method, such as cost, variances, and second moments for each region and each technique. For every cache c and technique t, the first and second moments are estimated from the iteration's training samples that landed in the cache and were produced by the technique itself

$$\mathbb{E}\left[\langle L_{\mathrm{r},t}\rangle\right] \approx \frac{1}{n_{t,c}} \sum_{s=1}^{n_{t,c}} \langle L_{\mathrm{r},t_s}(x_s)\rangle;\; \mathbb{E}\left[\langle L_{\mathrm{r},t}\rangle^2\right] \approx \frac{1}{n_{t,c}} \sum_{s=1}^{n_{t,c}} \langle L_{\mathrm{r},t_s}(x_s)\rangle^2,$$

from which we can compute the estimate for the variance.

To estimate the cost of each technique, we employ the well-established heuristic of using the average path length generated by the technique as a proxy for cost [Rath et al. 2022; Veach 1997].

Image estimates. In lieu of the real ground truth, we apply a denoiser [Áfra 2019] to determine the pixel estimates I_px. The pixel estimates are also used to estimate the image variance

$$\mathbb{V}\left[\langle I\rangle\right] \approx \frac{1}{N_\mathrm{px}} \sum_\mathrm{px} \frac{1}{N_\mathrm{spp}} \sum_{s=1}^{N_\mathrm{spp}} \left(\frac{\langle I_\mathrm{px}\rangle_s - \hat{I}_\mathrm{px}}{\hat{I}_\mathrm{px}}\right)^2.$$

The cost of the entire image is similarly estimated by counting the total number of rays that were traced.

Fixed-point iteration. Implementing our fixed-point iteration algorithm requires an iterative rendering process, in which each iteration performs some number of samples to estimate the variances and costs, thus, allowing us to perform steps toward the fixed-point. Path guiding already uses an iterative scheme to learn its guided sampling distribution in a training phase that we can integrate the fixed-point iteration into. We use ordinary throughput-based Russian roulette without splitting for the first three training iterations to give the data structure time to adapt and let the estimates become less noisy before we switch to our sample allocation.

Outlier clamping. Outliers can drastically reduce the quality of the variance estimates. To combat this, we clamp the contribution of paths to a fixed multiple of the pixel estimate (50). Note that this is only used for our local- and image variance estimates, and is not used for the render itself. Similar to previous works, we also clamp the maximal number of samples per technique (to $[0.05, 20]$).

Handling colors. To extend our theory to color, we perform computations channel-wise and use the L_2 norm to compute the final sample counts analogous to previous works [Rath et al. 2022].

4.2 Evaluation

We compare our approach against the state-of-the-art in path guiding, based on mixture optimization using gradient descent [Vorba et al. 2019] in conjunction with EARS [Rath et al. 2022] to steer sample counts ("EARS"). We use the variant that minimizes KL divergence of image contributions [Rath et al. 2020], which works best across all our scenes. As a baseline, we include path guiding with throughput-based RR ("classic RR") starting at the 5th bounce, while all other methods start performing RR with the first bounce.

We provide results of our strategy for the budget-unaware and the budget-aware MIS weight scenarios. As discussed, the budget-aware configuration uses the the budget-unaware derivations as an approximation. An ablation ("Ours + Grad. Descent") combines gradient descent (mixing BSDF and guiding) and our approach (sample counts of NEE and the mixture) to demonstrate that our approach chooses better mixtures than the previous state-of-the-art optimizer.

Setup. All images are rendered for 5 minutes (2.5min training, 2.5min rendering) on an AMD Ryzen Threadripper 2950X. The maximum path length is 40 and we perform 9 training iterations whose time increases exponentially [Müller et al. 2017]. We limit the data structure's footprint to a maximum of 72MB. The noise is quantified using relative mean squared error (relMSE), with 0.01% of the brightest outliers discarded to arrive at a robust error estimate.

Results. We tested our method on a set of 10 common benchmark scenes, where it achieves an average speedup of 1.32× over the previous state-of-the-art, even though it traces 17% fewer rays. We find that the budget-aware variant almost always yields superior performance. A selection of interesting scenes can be found in fig. 6.

In Living Room, our method automatically detects that BSDF sampling performs best to estimate the incident light on the floor. On diffuse surfaces, guiding is used instead. NEE is left mostly disabled, saving computational cost and yielding a 39% speedup.

The Glossy Kitchen contains various glossy surfaces, which make guiding and NEE poor choices. By disabling NEE almost entirely and using guiding only on the counter in the lower part where it is beneficial, we save precious time that can be spent elsewhere.

Corona Benchmark features strong indirect illumination from a directly illuminated spot on the sofa. Our method correctly identifies that any path arriving at the sofa should attempt a high number of NEE samples. Note that this would not be possible with per-pixel-based approaches, which are limited to using the same sampling ratios independent of where the path lands at later bounces.

The Bedroom is a scene where all three available techniques are useful in different areas. The left wall benefits greatly from performing NEE and every path encountering it can utilize this.

As an outdoor scene, we include the Pool. In this scene, containing many caustics, our budget-aware method achieves a speedup of 43%. Our method correctly identifies that guiding is necessary to estimate the caustic cast by the windows on the right of the pool.

Convergence in practice. We verified that our method converges in practice by ensuring repeated rendering yields the same results, and that the curves of average budget allocation per technique as well as overall efficiency flatten over time. We provide these convergence plots in the supplementary material.

5 APPLICATION II: BIDIRECTIONAL PATH TRACING

In our bidirectional application, the techniques to combine are path tracing, connections with light paths, and NEE. While NEE is a special case of light paths, its different variance and cost properties make it beneficial to assign it its own budget. In theory, performance could be improved further by assigning separate budgets of light paths for any length, but we leave exploring this as future work.

While it might make sense to perform splitting for both the camera- and light paths, doing so comes with its own set of challenges. Most importantly, splitting paths on both ends and combining all acquired intersections leads to an exponential increase in connections between the paths which is computationally expensive. For this reason, we resort to throughput-based RR for the light path.

We leave exploring more sophisticated variants of bidirectional rendering, such as vertex merging, light path caches, or stochastic connections as future work.

5.1 Implementation

The implementation of our second application is analogous to the first. Local estimates are recorded in the octree data structure proposed by EARS [Rath et al. 2022]. To execute our fixed-point scheme, we rewrite Mitsuba's BDPT implementation to become progressive: Rendering occurs in short iterations that grow in duration over time, with the final image being the inverse-variance weighted combination of all iterations [Vorba et al. 2019]. We apply a similar outlier rejection to EARS [Rath et al. 2022], which eliminates the 10 brightest pixels in each pass for the image variance estimate. Similar to guiding, we additionally tried applying an outlier rejection to the local samples recorded in the octree but found that this results on average in a 6% performance decrease on the evaluated scenes: Since in this application sampling is not learned adaptively, it seems beneficial to perform excessive splitting in the presence of outliers.

5.2 Evaluation

We compare against EARS applied to camera paths ("EARS") and provide the results of a vanilla BDPT baseline ("classic RR"). Sample allocation/RRS is only performed on the camera path side, and throughput-based RR is used to construct light paths. Additionally, we implemented a variation of efficiency-aware MIS [Grittmann et al. 2022] ("brute-force"), which is applied to pure bidirectional path tracing without merging and makes decisions on a per-pixel basis instead of a single global one to allocate the budget.

All images are rendered for 5 minutes on an AMD Ryzen Threadripper 2950X with 16 cores with a maximum path length of 40. The first 3 iterations use classic RR also for the camera path to let estimates converge. The data structure's memory size is limited to at most 72MB. As in guiding, we evaluate a version with budget-unaware MIS weights which performs 13% worse on average. Averaged, we achieve a speedup of 1.6× over EARS applied to BDPT, while tracing 13% fewer rays, and a speedup of 2.6× over BDPT.

The teaser (fig. 1) shows a modified version of the KITCHENETTE including a small tealight. Plain RRS cannot separate the contributions of the different techniques, wasting compute time on negligible contributions, while our method can locally identify regions where each particular sampling technique performs well.

Figure 7 showcases a variety of interesting scenes, e.g., the GLOSSY KITCHEN that shows a similar pattern. The glossy features neither benefit greatly from light connections nor NEE. Being able to automatically detect and tune down these techniques locally, still using connections on diffuse surfaces, yields a speedup of 74%.

The GLOSSY BATHROOM similarly avoids NEE, but for a different reason: No light is directly visible in this scene. Instead, our method relies on BSDF sampling for glossy surfaces and light paths for rougher surfaces. Similar to EARS, our method uses less splitting overall (indicated by the darker colors). EARS exhibits the same issue as in its original paper due to issues with variance estimates on glossy surfaces. While our method fundamentally suffers from the same issue, this is offset by an overall better sample allocation resulting in a 79% speedup over EARS.

In contrast to the previous scenes, the LIVING ROOM is a scene where NEE is invaluable, given that guiding is unavailable. The sofa on the right is directly hit by the light sources and well estimated by NEE. Random BSDF bounces from diffuse elements in the scene ending at this location can exploit this effective technique. Light paths are avoided entirely, as they struggle to penetrate the room.

The BEDROOM and the BOOKSHELF both feature difficult indirect illumination where light is only induced from small areas: from behind the curtains and from small light sources, respectively. For BSDF samples alone, it is hard to find a contributing path without the assistance of NEE or light path connections. Therefore, at intersections where there are no occluders and a direct light connection is possible, our method automatically increases the NEE sampling budget to reduce the variance at this intersection. At other intersections, with an occluder present, light connections can provide help avoiding it. Making these decisions locally in space allows for superior estimations of indirect illumination while previous work is limited to taking a single decision for the whole path.

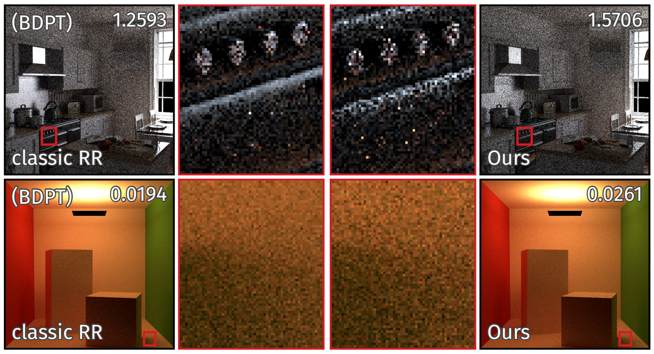


Fig. 5. We show two failure cases. The COUNTRY KITCHEN is a noisy scene in which our method requires time to converge and convert the overhead into an advantage. The 30 seconds it was rendered for do not suffice, but by rendering only slightly longer we surpass BDPT's performance. In the second case, we present a CORNELL BOX. In trivial scenes, the overhead we introduce dominates the advantage. We give the relMSE in the upper right.

Convergence in practice. As in path guiding, we verified that repeated runs result in the same performance. We similarly provide plots for the convergence of average sample allocations across techniques and overall efficiency in the supplementary material.

6 LIMITATIONS AND FUTURE WORK

Our method efficiently allocates sampling budgets while remaining flexible and extensible, but relies heavily on the accuracy and robustness of local estimates for variance.

Overhead. Our method introduces the overhead of collecting local estimates which does not always pay off. In the presence of very short rendering times, the collected estimates have not converged enough yet when the rendering finishes. In fig. 5 we present the COUNTRY KITCHEN where a rendering time of 30 seconds does not suffice, even though we outperform BDPT quickly afterward.

Local estimates. Outliers in the local estimates can reduce performance due to excessive splitting. We used a simple outlier rejection scheme in the guiding application, but future work could look into applying more sophisticated techniques [Zirr et al. 2018].

Discontinuity artefacts. Noise in the spatial caches can lead to noticeable discontinuities [Rath et al. 2022], which is especially pronounced with shorter render times. Possible mitigations could be to apply spatial denoising to the local estimates or interpolate them to arrive at smoothly varying sample counts.

Proxy accuracy. Our proxy is only optimal for budget-unaware MIS weights. While it performs robustly for budget-aware weights across all our tested scenes, a more sophisticated proxy can likely increase efficiency even further. Our supplemental includes additional 1D examples that investigate the error of the proxy further.

Dynamic scenes. We only considered static scenes. Future work could extend our method to dynamic scenes, such as real-time rendering, in which local estimates or sample allocations from earlier frames could be re-used to kickstart the fixed-point scheme.

MIS weights. We limited our analysis to the most common MIS weighting function, the balance heuristic [Veach and Guibas 1995b]. Future work could investigate the impact of more sophisticated budget-aware weights on our proxy model, such as the optimal weights proposed by Kondapaneni et al. [2019].

Correlated Techniques. Our decomposition of the variance assumes that techniques are uncorrelated. To perform optimally for correlated techniques (such as photon mapping), one also needs to track correlation statistics, which we leave as future work.

Applications outside of rendering. We presented our approach in the context of rendering, but it could be applied to any Monte Carlo integration problem that relies on multi-sample MIS estimation.

7 CONCLUSION

We present a novel theory on multi-sample MIS allocation strategy based on fixed-point schemes. For budget-unaware weights, our method is proven to yield optimal performance as long as variances and costs are known accurately. We demonstrate that our approach achieves consistent speedups over previous approaches, even with noisy estimates and budget-aware weights, in both path guiding and bidirectional path tracing. Our theory can be applied to any Monte Carlo integration problem, even outside of rendering, in which multiple sampling techniques are combined through MIS.

ACKNOWLEDGMENTS

This project has received funding from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through GRK 2853/1 "Neuroexplicit Models of Language, Vision, and Action" (project number 471607914), as well as from Velux Stiftung project 1350. We thank the anonymous reviewers for their insightful remarks, as well as the artists of our test scenes: Benedikt Bitterli [2016], Miika Aittala, Samuli Laine, and Jaakko Lehtinen (Veach Door), Evermotion and Tiziano Portenier (Bookshelf, Glossy Bathroom, Glossy Kitchen), Ondřej Karlík (Pool), Wig42 (Modern Living Room), Jay-Artist (Country Kitchen), and Ludvík Koutný (Living Room).

REFERENCES

Attila T. Áfra. 2019. Intel® Open Image Denoise. https://www.openimagedenoise.org/

James Arvo and David Kirk. 1990. Particle Transport and Image Synthesis. *ACM SIGGRAPH Computer Graphics* 24, 4 (Sept. 1990), 63–66. https://doi.org/10.1145/97880.97886

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/

Mark R. Bolin and Gary W. Meyer. 1997. An Error Metric for Monte Carlo Ray Tracing. In *Rendering Techniques '97 (Eurographics)*, Julie Dorsey and Philipp Slusallek (Eds.). Springer, Vienna, 57–68. https://doi.org/10.1007/978-3-7091-6858-5_6

Pascal Grittmann, Ömercan Yazici, Iliyan Georgiev, and Philipp Slusallek. 2022. Efficiency-Aware Multiple Importance Sampling for Bidirectional Rendering Algorithms. *ACM Transactions on Graphics* 41, 4 (July 2022), 80:1–80:12. https://doi.org/10.1145/3528223.3530126

Johannes Hanika, Marc Droske, and Luca Fascione. 2015. Manifold Next Event Estimation. *Computer Graphics Forum* 34, 4 (July 2015), 87–97. https://doi.org/10.1111/cgf.12681

Wenzel Jakob. 2010. Mitsuba Renderer. https://mitsuba-renderer.org/

Henrik Wann Jensen. 1995. Importance Driven Path Tracing Using the Photon Map. In *Rendering Techniques '95*, Patrick M. Hanrahan and Werner Purgathofer (Eds.). Springer, Vienna, 326–335. https://doi.org/10.1007/978-3-7091-9430-0_31

James T. Kajiya. 1986. The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 143–150. https://doi.org/10.1145/15922.15902

Ivo Kondapaneni, Petr Vevoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Křivánek. 2019. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 4 (July 2019), 37:1–37:14. https://doi.org/10.1145/3306346.3323009

Eric P. Lafortune and Y. Willems. 1993. Bi-Directional Path Tracing.

H. Lu, R. Pacanowski, and X. Granier. 2013. Second-Order Approximation for Variance Reduction in Multiple Importance Sampling. In *Computer Graphics Forum*, Vol. 32. 131–136. https://doi.org/10.1111/cgf.12220

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum* 36, 4 (2017), 91–100. https://doi.org/10.1111/cgf.13227

David Murray, Sofiane Benzait, Romain Pacanowski, and Xavier Granier. 2020. On Learning the Best Local Balancing Strategy. In *Eurographics 2020 - Annual Conference of the European Association for Computer Graphics*, Vol. 20. 1–4. https://doi.org/10.2312/egs.20201009

Kosuke Nabata, Kei Iwasaki, and Yoshinori Dobashi. 2020. Resampling-Aware Weighting Functions for Bidirectional Path Tracing Using Multiple Light Sub-Paths. *ACM Transactions on Graphics* 39, 2 (March 2020), 15:1–15:11. https://doi.org/10.1145/3338994

Matt Pharr, Wenzel Jakob, and Greg Humphreys (Eds.). 2016. *Physically Based Rendering: From Theory to Implementation (3rd Ed.).* Morgan Kaufmann Publishers Inc.

Stefan Popov, Ravi Ramamoorthi, Fredo Durand, and George Drettakis. 2015. Probabilistic Connections for Bidirectional Path Tracing. *Computer Graphics Forum* 34, 4 (2015), 75–86. https://doi.org/10.1111/cgf.12680

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Vévoda, Philipp Slusallek, and Jaroslav Křivánek. 2020. Variance-Aware Path Guiding. *ACM Transactions on Graphics (TOG)* 39, 4 (Aug. 2020), 151:1–151:12. https://doi.org/10.1145/3386569.3392441

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Philippe Weier, and Philipp Slusallek. 2022. EARS: Efficiency-Aware Russian Roulette and Splitting. *ACM Transactions on Graphics* 41, 4 (July 2022), 81:1–81:14. https://doi.org/10.1145/3528223.3530168

Alexander Rath, Ömercan Yazici, and Philipp Slusallek. 2023. Focal Path Guiding for Light Transport Simulation. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3588432.3591543

Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust Fitting of Parallax-Aware Mixtures for Path Guiding. *ACM Transactions on Graphics* 39, 4 (Aug. 2020), 147:1–147:15. https://doi.org/10.1145/3386569.3392421

Mateu Sbert, Vlastimil Havran, and László Szirmay-Kalos. 2019. Optimal Deterministic Mixture Sampling. *Eurographics 2019 - Short Papers* 73 (2019), 4 pages. https://doi.org/10.2312/EGS.20191018

Laszlo Szirmay-Kalos and Mateu Sbert. 2022. Robust Sample Budget Allocation for MIS. In *Eurographics 2022 - Short Papers*. The Eurographics Association, 17–20. https://doi.org/10.2312/EGS.20221022

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation.* Ph. D. Dissertation. Stanford University, CA, USA. Advisor(s) Guibas, Leonidas J.

Eric Veach and Leonidas Guibas. 1995a. Bidirectional Estimators for Light Transport. In *Photorealistic Rendering Techniques (Focus on Computer Graphics)*, Georgios Sakas, Stefan Müller, and Peter Shirley (Eds.). Springer, Berlin, Heidelberg, 145–167. https://doi.org/10.1007/978-3-642-87825-1_11

Eric Veach and Leonidas J. Guibas. 1995b. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 419–428. https://doi.org/10.1145/218380.218498

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path Guiding in Production. In *ACM SIGGRAPH 2019 Courses (SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, 1–77. https://doi.org/10.1145/3305366.3328091

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-Line Learning of Parametric Mixture Models for Light Transport Simulation. *ACM Transactions on Graphics* 33, 4 (July 2014), 101:1–101:11. https://doi.org/10.1145/2601097.2601203

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation. *ACM Transactions on Graphics* 35, 4 (July 2016), 42:1–42:11. https://doi.org/10.1145/2897824.2925912

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints. *ACM Transactions on Graphics* 39, 4 (Aug. 2020), 149:1–149:15. https://doi.org/10.1145/3386569.3392408

Tobias Zirr, Johannes Hanika, and Carsten Dachsbacher. 2018. Re-Weighting Firefly Samples for Improved Finite-Sample Monte Carlo Estimates. In *Computer Graphics Forum*, Vol. 37. 410–421. https://doi.org/10.1111/cgf.13335

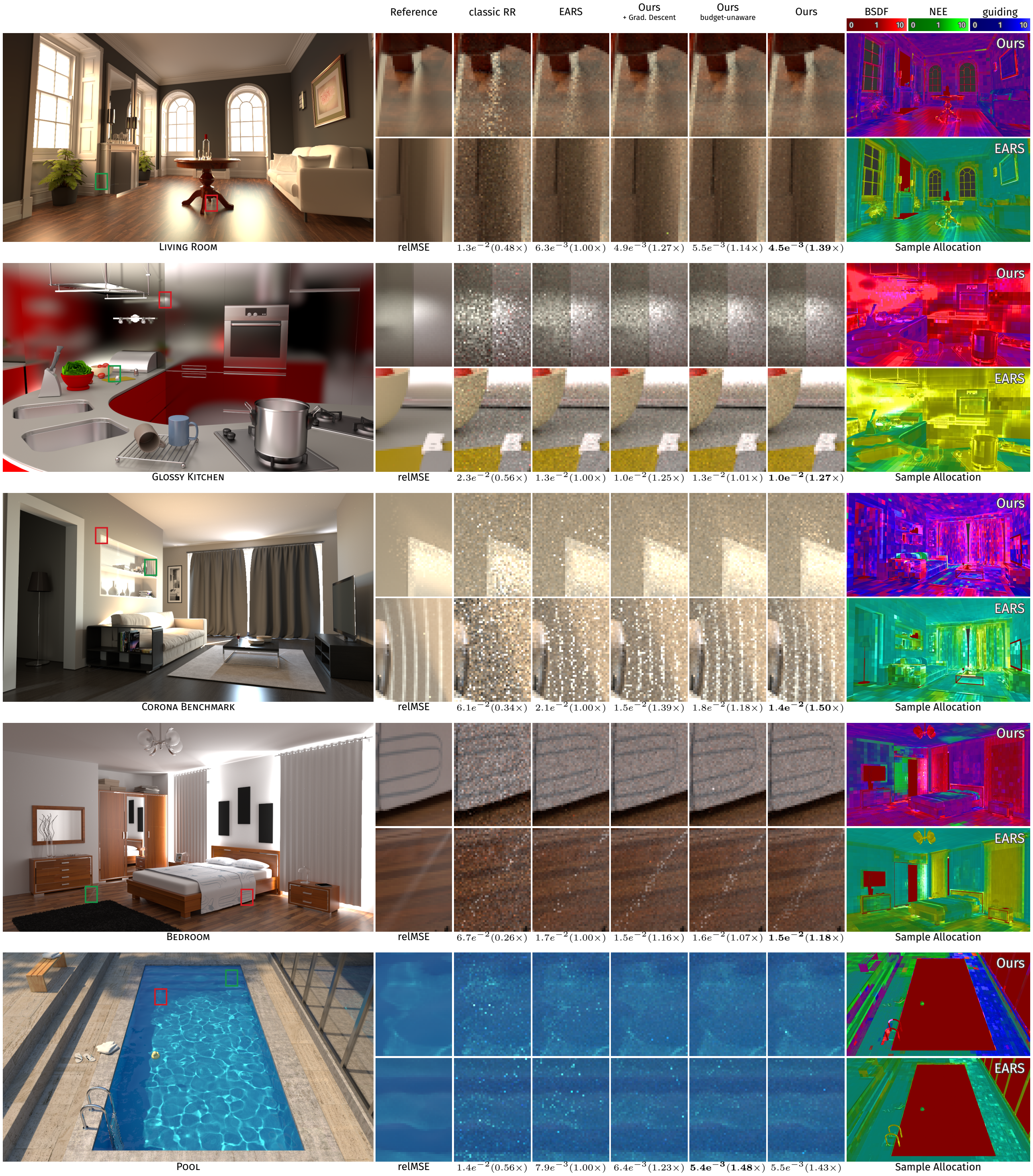

Reference
classic RR
EARS
Ours + Grad. Descent
Ours budget-unaware
Ours
BSDF 0 1 10
NEE 0 1 10
guiding 0 1 10
Ours
EARS
LIVING ROOM
relMSE
1.3e−2(0.48×) 6.3e−3(1.00×) 4.9e−3(1.27×) 5.5e−3(1.14×) 4.5e−3(1.39×)
Sample Allocation
GLOSSY KITCHEN
relMSE
2.3e−2(0.56×) 1.3e−2(1.00×) 1.0e−2(1.25×) 1.3e−2(1.01×) 1.0e−2(1.27×)
Sample Allocation
CORONA BENCHMARK
relMSE
6.1e−2(0.34×) 2.1e−2(1.00×) 1.5e−2(1.39×) 1.8e−2(1.18×) 1.4e−2(1.50×)
Sample Allocation
BEDROOM
relMSE
6.7e−2(0.26×) 1.7e−2(1.00×) 1.5e−2(1.16×) 1.6e−2(1.07×) 1.5e−2(1.18×)
Sample Allocation
POOL
relMSE
1.4e−2(0.56×) 7.9e−3(1.00×) 6.4e−3(1.23×) 5.4e−3(1.48×) 5.5e−3(1.43×)
Sample Allocation

Fig. 6. We compare our method against state-of-the-art EARS applied to guiding in 5 scenes in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error over the whole image (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and guiding samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in the supplementary material. Our "+ Grad. Descent" test uses gradient descent for the BSDF/guiding mixture and our method to combine it with NEE using budget-aware weights.

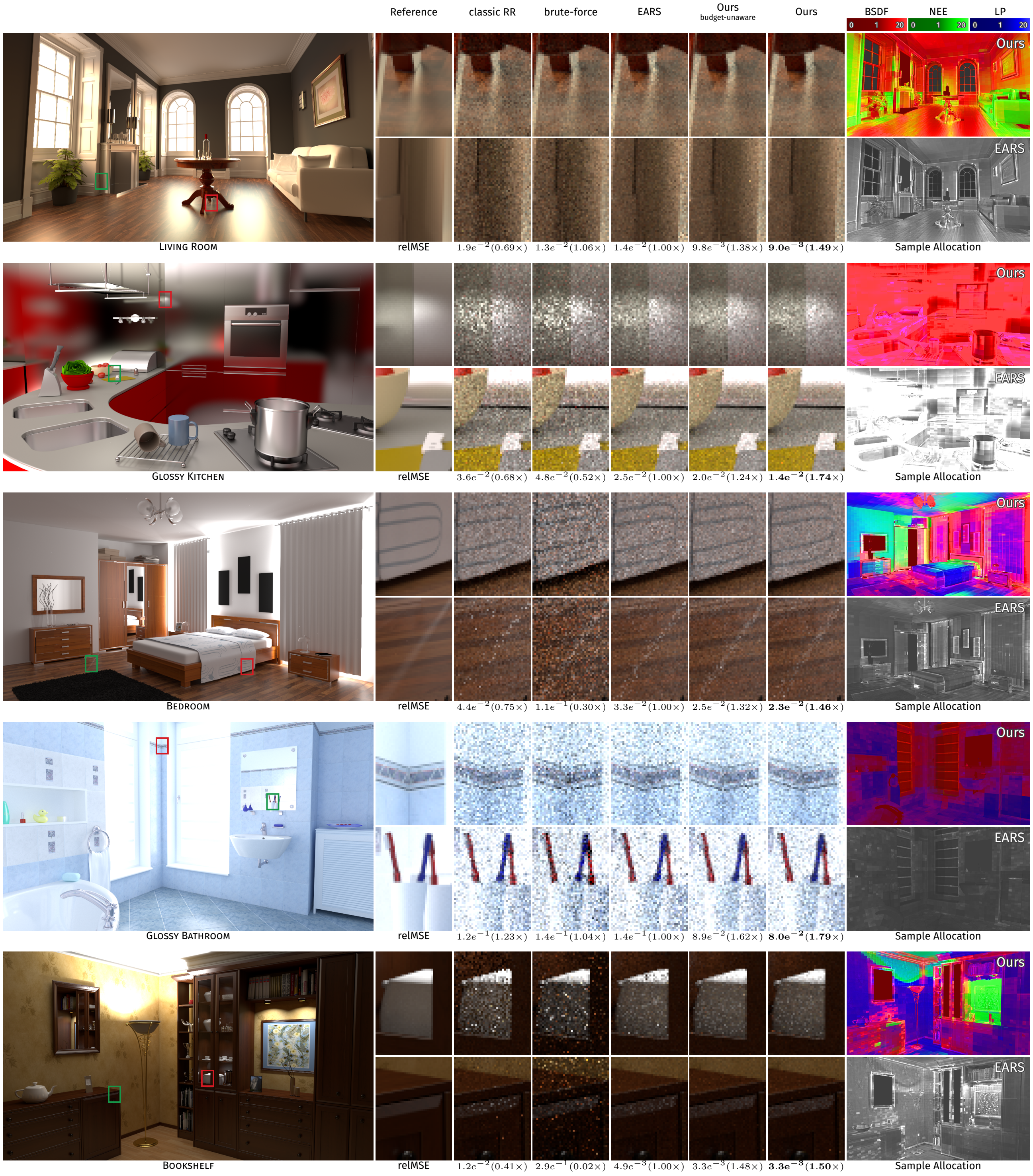


Fig. 7. We compare our method against state-of-the-art EARS applied to BDPT in 5 scenes in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error over the whole image (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and light path samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in the supplementary material.

A DERIVATION

In this section, we provide the formal derivation of the fixed-point iteration scheme that is described in the paper. We start by introducing the model and follow up with the derivation of the formulation for variance. With these foundations, we can give an expression for efficiency to take the derivative of and compute its root.

A.1 Base Model

We are interested in finding the value of $I = \int_{\mathcal{X}} f(x)\,\mathrm{d}x$ by combining estimates from n_t different strategies with varying sample counts β_t, leading us to a multi-sample MIS formulation

$$\langle I\rangle = \sum_{t=1}^{n_t} \frac{1}{\beta_t} \sum_{s=1}^{\beta_t} \langle I_t(x_{t,s})\rangle = \sum_{t=1}^{n_t} \langle I_t(x_{t,\cdot}); \beta_t\rangle,$$

where $\langle I; \beta\rangle$ is a secondary estimator with sample count β. Further, we express the integrand f as a sum of subintegrands $f = \sum_{i=1}^{n_i} f_i$, allowing us to consider and combine techniques that estimate different, possibly disjoint, parts of the main integrand. This gives each primary estimator the form

$$\langle I_t(x)\rangle = \sum_{i=1}^{n_i} \frac{f_i(x)}{p_t(x)} w_{it}(x),$$

where p_t is the probability density function of technique t and $w_{it}(x)$ is the MIS weight with respect to all other methods that estimate integrand f_i. Its dependency on i is due to the characteristic function $\mathbb{1}_{it}$ in the MIS weights, which is 1 whenever technique t estimates integrand f_i and 0 otherwise. This ensures correctness and $\sum_{t=1}^{n_t} w_{it}(x) = 1$ for all i. For example, for the budget-unaware balance heuristic we obtain

$$w_{it}(x) = \frac{\mathbb{1}_{it} p_t(x)}{\sum_{k=1}^{n_t} \mathbb{1}_{ik} p_k(x)}.$$

For brevity, we will drop the argument of I_t where it is not the subject of interest. Note, that we can also drop the argument with explicit subscript s for the s-th sample since the samples are pairwise independent.

Continuity. Sample allocation is a discrete problem since we can only take a natural number of samples. Similar to previous work [Vorba and Křivánek 2016], we employ the stochastic rounding function

$$r(\beta) = \begin{cases} \lfloor\beta\rfloor + 1 & \text{with probability } \beta - \lfloor\beta\rfloor \\ \lfloor\beta\rfloor & \text{otherwise,} \end{cases}$$

to allow for real-valued β and make the optimization simpler.

Unbiasedness. Having adapted the standard multi-sample MIS model and introduced stochastic rounding, we have to verify that our model still yields an unbiased estimate of I. This is a straightforward computation in which we use that the sum of the MIS weights over all techniques is 1:

$$\begin{aligned}
\mathbb{E}\left[\langle I\rangle\right] &= \frac{1}{\beta_t} \sum_{t=1}^{n_t} \mathbb{E}\left[\sum_{s=1}^{r(\beta_t)} \langle I_t\rangle\right] \\
&= \left(\frac{(\beta_t - \lfloor\beta_t\rfloor)(\lfloor\beta_t\rfloor + 1)}{\beta_t} + \frac{(1 - \beta_t + \lfloor\beta_t\rfloor)\lfloor\beta_t\rfloor}{\beta_t}\right) \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t\rangle\right] \\
&= \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t\rangle\right] = \sum_{t=1}^{n_t} \sum_{i=1}^{n_i} \mathbb{E}\left[\frac{f_i(x)}{p_t(x)} w_{it}(x)\right] \\
&= \int_{\mathcal{X}} \sum_{i=1}^{n_i} f_i(x) \underbrace{\left(\sum_{t=1}^{n_t} w_{it}(x)\right)}_{=1} \mathrm{d}x = I.
\end{aligned}$$

Variance. The efficiency $\mathcal{E}$ [Pharr et al. 2016] of an estimator $\langle I\rangle$ is given by

$$\mathcal{E}[\langle I\rangle] = \frac{1}{\mathbb{V}\left[\langle I\rangle\right] \mathbb{C}\left[\langle I\rangle\right]}.$$

Consequently, to allocate sampling budgets maximizing efficiency, having formulations for variance and cost is essential. We start with variance. It is generally given by

$$\mathbb{V}\left[\langle I\rangle\right] = \mathbb{E}\left[\langle I\rangle^2\right] - \mathbb{E}^2\left[\langle I\rangle\right].$$

We already know that $\mathbb{E}^2\left[\langle I\rangle\right] = I^2$ since our estimator is unbiased. Thus, we only need to compute the second moment of our estimator

$$\begin{aligned}
\mathbb{E}\left[\langle I\rangle^2\right] &= \mathbb{E}\left[\left(\sum_{t=1}^{n_t} \langle I_t; \beta_t\rangle\right)^2\right] \\
&= \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t; \beta_t\rangle^2\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k; \beta_k\rangle\langle I_l; \beta_l\rangle\right] \\
&= \sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t; \beta_t\rangle^2\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k\rangle\right] \mathbb{E}\left[\langle I_l\rangle\right],
\end{aligned}$$

where for the last equality we assume independence of the techniques, which usually is a reasonable assumption, and the fact that $\mathbb{E}\left[\langle I_t; \beta_t\rangle\right] = \mathbb{E}\left[\langle I_t\rangle\right]$, as can easily be seen in the previous unbiasedness computation.

Assuming independence of the acquired samples within a technique, and $\beta_t \in \mathbb{N}_{>0}$, take note of the identity

$$\mathbb{E}\left[\left(\sum_{s=1}^{\beta_t} \langle I_t\rangle\right)^2\right] = \beta_t \mathbb{E}\left[\langle I_t\rangle^2\right] + \beta_t(\beta_t - 1)\mathbb{E}^2\left[\langle I_t\rangle\right].$$

Using it, we can further simplify the second moment by getting rid of its inherent dependency on β_t. For any fixed t, the secondary estimator's second moment is of the form

$$\begin{aligned}
\mathbb{E}\left[\langle I_t; \beta_t\rangle^2\right] &= \frac{1}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{r(\beta_t)} \langle I_t\rangle\right)^2\right] \\
&= \frac{\beta_t - \lfloor\beta_t\rfloor}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{\lfloor\beta_t\rfloor+1} \langle I_t\rangle\right)^2\right] \\
&\quad + \frac{1 - \beta_t + \lfloor\beta_t\rfloor}{\beta_t^2} \mathbb{E}\left[\left(\sum_{s=1}^{\lfloor\beta_t\rfloor} \langle I_t\rangle\right)^2\right] \\
&= \frac{1}{\beta_t} \mathbb{E}\left[\langle I_t\rangle^2\right] + \frac{2\beta_t\lfloor\beta_t\rfloor - \lfloor\beta_t\rfloor^2 - \lfloor\beta_t\rfloor}{\beta_t^2} \mathbb{E}^2\left[\langle I_t\rangle\right].
\end{aligned}$$

One last trick is to express the first moment squared, I^2, in terms as a sum of the technique estimators as follows

$$I^2 = \left(\sum_{t=1}^{n_t} \mathbb{E}\left[\langle I_t \rangle\right]\right)^2 = \sum_{t=1}^{n_t} \mathbb{E}^2\left[\langle I_t \rangle\right] + \sum_{k \neq l}^{n_t} \mathbb{E}\left[\langle I_k \rangle\right] \mathbb{E}\left[\langle I_l \rangle\right].$$

Putting these formulations for the first and second moment together, the formula for the variance of our estimator simply becomes the sum of the variance of all techniques in use

$$\mathbb{V}\left[I\right] = \sum_{t=1}^{n_t} \mathbb{V}\left[\langle I_t; \beta_t \rangle\right] = \sum_{t=1}^{n_t} \left(\frac{1}{\beta_t}\mathbb{V}\left[\langle I_t \rangle\right] + \rho(\beta_t)\mathbb{E}^2\left[\langle I_t \rangle\right]\right),$$

where $\rho(\beta) = \frac{(\beta - \lfloor \beta \rfloor)(\lceil \beta \rceil - \beta)}{\beta^2}$ arises due to stochastic rounding.

Cost. Modeling the cost of our estimator is done in a similar way as in previous work [Rath et al. 2022]. We assume the cost $\mathbb{C}_t$ of taking a single sample for technique t to be constant. To obtain the cost of the overall estimator, we multiply each technique's cost by the *real-valued* β_t and sum them up. We also include a constant cost term $\mathbb{C}_\Delta$ to incorporate all overhead costs. Formally, this yields $\mathbb{C}\left[\langle I \rangle\right] = \sum_{t=1}^{n_t} \beta_t \mathbb{C}_t + \mathbb{C}_\Delta$.

A.2 Optimizing for Efficiency

Our goal is to obtain a sample allocation that maximizes efficiency. Instead of doing this directly, we consider the equivalent problem of minimizing *inverse efficiency* $\mathcal{E}^{-1}$ to improve the readability and reduce clutter of the following equations. At a minimum, the derivative necessarily has a root. Therefore, we first compute the derivative w.r.t. all β_t of inverse efficiency in this section. In the next sections, we then find an expression for its minimum.

Efficiency derivative. We can use the product formula to get

$$\frac{\mathrm{d}\mathcal{E}^{-1}[\langle I \rangle]}{\mathrm{d}\beta_t} = \frac{\mathrm{d}\mathbb{V}\left[\langle I \rangle\right]}{\mathrm{d}\beta_t}\mathbb{C}\left[\langle I \rangle\right] + \mathbb{V}\left[\langle I \rangle\right]\frac{\mathrm{d}\mathbb{C}\left[\langle I \rangle\right]}{\mathrm{d}\beta_t},$$

revealing that we need derivatives of the cost and variance expressions computed previously.

Cost derivative. The derivative of cost is easy as only one of the summands depends on β_t. Therefore: $\frac{\mathrm{d}\mathbb{C}[\langle I \rangle]}{\mathrm{d}\beta_t} = \mathbb{C}_t$.

Variance derivative. The derivative of variance is given by

$$\begin{aligned}\frac{\mathrm{d}\mathbb{V}\left[\langle I \rangle\right]}{\mathrm{d}\beta_t} &= \sum_{k=1}^{n_t}\left(\frac{\mathrm{d}}{\mathrm{d}\beta_t}\frac{1}{\beta_k}\mathbb{V}\left[\langle I_k \rangle\right] + \frac{\mathrm{d}}{\mathrm{d}\beta_t}\rho(\beta_k)\mathbb{E}^2\left[\langle I_k \rangle\right]\right) \\ &= -\frac{1}{\beta_t^2}\mathbb{V}\left[\langle I_t \rangle\right] + \frac{\mathrm{d}\rho(\beta_t)}{\mathrm{d}\beta_t}\mathbb{E}^2\left[\langle I_t \rangle\right] \\ &\quad + \underbrace{\sum_{k=1}^{n_t}\left(\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}\left[\langle I_k \rangle\right]}{\mathrm{d}\beta_t} + \rho(\beta_k)\frac{\mathrm{d}\mathbb{E}^2\left[\langle I_k \rangle\right]}{\mathrm{d}\beta_t}\right)}_{=0\text{, for budget-unaware MIS weights}}\end{aligned}$$

Note that, at this point, the sum over all n_t techniques is 0 if the MIS weights are budget-unaware, i.e., do not contain β. Moreover, the derivative of ρ in this equation, which arises due to stochastic rounding, forces us to make a simplification that has been done by previous work in similar situations as well [Rath et al. 2022], i.e., we assume $\rho(\beta) = \max(1, \frac{1}{\beta}) - 1$. It is easy to verify that this simplification only affects the $\beta > 1$ case, giving us

$$\mathbb{V}\left[\langle I_t; \beta_t \rangle\right] = \begin{cases}\frac{1}{\beta_t}\mathbb{E}\left[\langle I_t \rangle^2\right] - \mathbb{E}^2\left[\langle I_t \rangle\right] & \text{if } \beta_t < 1 \\ \frac{1}{\beta_t}\mathbb{V}\left[\langle I_t \rangle\right] & \text{otherwise}\end{cases} \tag{3}$$

as a formulation for variance, and

$$\frac{\mathrm{d}\mathbb{V}\left[\langle I \rangle\right]}{\mathrm{d}\beta_t} = \begin{cases}-\frac{1}{\beta_t^2}\mathbb{E}\left[\langle I_t \rangle^2\right] & \text{if } \beta_t < 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}\left[\langle I_t \rangle\right] & \text{otherwise}\end{cases}$$

for the derivative in the *budget-unaware* scenario.

In the budget-aware scenario, the exact expression strongly depends on the MIS weight heuristic, but is not generally convex anymore even for simple heuristics like the balance heuristic. Hence, we will focus on the budget-unaware case and treat it as a simplification for the budget-aware case. With this simplification, we lose optimality, but we show empirically that it performs well both in simple 1D examples (see fig. 10) and complex rendering applications (see paper).

Putting the equations together, we obtain

$$\frac{\mathrm{d}\mathcal{E}^{-1}[\langle I \rangle]}{\mathrm{d}\beta_t} = \begin{cases}-\frac{1}{\beta_t^2}\mathbb{E}\left[\langle I_t \rangle^2\right]\mathbb{C}\left[\langle I \rangle\right] + \mathbb{V}\left[\langle I \rangle\right]\mathbb{C}_t & \text{if } \beta_t \leq 1, \\ -\frac{1}{\beta_t^2}\mathbb{V}\left[\langle I_t \rangle\right]\mathbb{C}\left[\langle I \rangle\right] + \mathbb{V}\left[\langle I \rangle\right]\mathbb{C}_t & \text{otherw.}\end{cases} \tag{4}$$

Budget-aware derivatives. The variance's derivative includes some extra terms that are generated by the derivative of the MIS weights. For any fixed k, these extra terms take the form

$$\begin{aligned}&\frac{1}{\beta_k}\frac{\mathrm{d}\mathbb{V}\left[\langle I_k \rangle\right]}{\mathrm{d}\beta_t} + \rho(\beta_k)\frac{\mathrm{d}\mathbb{E}^2\left[\langle I_k \rangle\right]}{\mathrm{d}\beta_t} \\ &= \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k \rangle\frac{\mathrm{d}\langle I_k \rangle}{\mathrm{d}\beta_t}\right] + \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k \rangle\right]\mathbb{E}^2\left[\frac{\mathrm{d}\langle I_k \rangle}{\mathrm{d}\beta_t}\right] \\ &\quad + 2\rho(\beta_k)\mathbb{E}\left[\langle I_k \rangle\right]\mathbb{E}^2\left[\frac{\mathrm{d}\langle I_k \rangle}{\mathrm{d}\beta_t}\right] \\ &= \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k \rangle\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right] \\ &\quad + \frac{2}{\beta_k}\mathbb{E}\left[\langle I_k \rangle\right]\mathbb{E}^2\left[\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right] \\ &\quad + 2\rho(\beta_k)\mathbb{E}\left[\langle I_k \rangle\right]\mathbb{E}^2\left[\sum_{i=1}^{n_i}\frac{f_i(x)}{p_k(x)}\frac{\mathrm{d}w_{ik}(x)}{\mathrm{d}\beta_t}\right].\end{aligned}$$

A.3 Fixed-Point Iteration

To devise a fixed-point scheme, we need to find the inverse efficiency's derivative's roots, which we achieve by rearranging eq. (4). For brevity, let $M_t = \frac{\mathrm{d}\mathbb{V}[\langle I \rangle]}{\mathrm{d}\beta_t}$. Then

$$\begin{aligned}&\frac{\mathrm{d}\mathcal{E}^{-1}[\langle I \rangle]}{\mathrm{d}\beta_t} \overset{!}{=} 0 \\ \iff & -\frac{1}{\beta_t^2}M_t\mathbb{C}\left[\langle I \rangle\right] + \mathbb{V}\left[\langle I \rangle\right]\mathbb{C}_t = 0 \\ \iff & \beta_t = \sqrt{\frac{M_t}{\mathbb{V}\left[\langle I \rangle\right]}\frac{\mathbb{C}\left[\langle I \rangle\right]}{\mathbb{C}_t}},\end{aligned}$$

where the last equivalence holds because $\beta_t > 0$.

With this equation, there is to note that β_t appears on both sides. And non-linearly so on the right-hand side as a part of $\mathbb{V}\left[\langle I\rangle\right]$. Hence, an analytical solution becomes impractical and we have to resort to numerical methods. The equation looks similar to what previous work [Rath et al. 2022] has encountered before and solved with a fixed-point iteration scheme. Following their example, we can employ a fixed-point iteration by using the equation above as an update function and compute the new sample allocations based on the variances and costs of using the previous iterations' allocations, as given by

$$\beta_{t,i+1} = \begin{cases} \beta^{\mathrm{RR}}_{t,i} = \sqrt{\frac{\mathbb{E}[\langle I_t;\beta_{t,i}\rangle^2]}{\mathbb{V}[\langle I;\boldsymbol{\beta}_i\rangle]}\frac{\mathbb{C}[\langle I;\boldsymbol{\beta}_i\rangle]}{\mathbb{C}_t}} & \text{if } \beta^{\mathrm{RR}}_{t,i} < 1, \\ \beta^{\mathrm{S}}_{t,i} = \sqrt{\frac{\mathbb{V}[\langle I_t;\beta_{t,i}\rangle]}{\mathbb{V}[\langle I;\boldsymbol{\beta}_i\rangle]}\frac{\mathbb{C}[\langle I;\boldsymbol{\beta}_i\rangle]}{\mathbb{C}_t}} & \text{if } \beta^{\mathrm{S}}_{t,i} > 1, \\ 1 & \text{otherwise,} \end{cases}$$

where $\boldsymbol{\beta}_i$ is the vector of all $\beta_{t,i}$ of the i-th iteration. The third case arises due to a technicality. With our proxy, the problem is split into two cases, one where $\beta_t \leq 1$ and one where $\beta_t > 1$. In some scenarios, an update step using the first case would lead to $\beta_{t,i+1} \geq 1$, whereas an update step in the second case would lead to $\beta_{t,i+1} \leq 1$. The two cases are in disagreement. Therefore, in such scenarios, we resort to following neither decision by allocating a single sample.

A.4 Proof of Optimality for Budget-Unaware Weights

We know that the efficiency is of the form

$$\mathcal{E}^{-1}(\beta) = \underbrace{\left(\sum_t^{n_t}\frac{1}{\beta_t}\mathbb{V}[\langle I_t\rangle] + \mathbb{V}_\Delta\right)}_{\mathbb{V}_\Sigma}\underbrace{\left(\sum_t^{n_t}\beta_t\mathbb{C}[\langle I_t\rangle] + \mathbb{C}_\Delta\right)}_{\mathbb{C}_\Sigma},$$

and hence optimal β^*_t must satisfy

$$\beta^*_t = \sqrt{\frac{\mathbb{V}[\langle I_t\rangle]}{\mathbb{C}[\langle I_t\rangle]}\frac{\mathbb{C}_\Sigma}{\mathbb{V}_\Sigma}} = \lambda\sqrt{\frac{\mathbb{V}[\langle I_t\rangle]}{\mathbb{C}[\langle I_t\rangle]}}.$$

For budget-unaware weights, the variance $\mathbb{V}[\langle I_t\rangle]$ is independent of β_t, and hence all that remains to be found is λ. Rewriting the inverse efficiency in terms of λ yields

$$\mathcal{E}^{-1}(\lambda) = \left(\frac{1}{\lambda}M + \mathbb{V}_\Delta\right)\left(\lambda M + \mathbb{C}_\Delta\right),$$

where $M = n_t\sqrt{\mathbb{V}[\langle I_t\rangle]\mathbb{C}[\langle I_t\rangle]}$. Since our update formula enforces that we move on the line defined by λ, our fixed-point scheme degenerates to the EARS fixed-point scheme after a single step, which has been proven to be optimal [Rath et al. 2022].

B ADDITIONAL DISCUSSION

B.1 Rounding

Rounding variants. Rounding fractional sample counts to integers, which is required since we can only take an integer amount of samples, always introduces additional noise in the secondary estimator. The shape and magnitude of noise depend on how the secondary estimator is normalized after rounding. We evaluate two unbiased options: **(1)** dividing by the real-valued sample count as done by previous works [Rath et al. 2022; Vorba and Křivánek 2016], and **(2)** dividing by the rounded sample count in the case of splitting and real-valued count for RR. The theoretical difference in noise is demonstrated in fig. 8. In practice, we have found that both techniques perform equally well, and hence for consistency with previous works, we normalize by real-valued sample counts in all of our evaluations.

Low discrepancy rounding. In practice, we apply a small trick to reduce the variance introduced by stochastic rounding by roughly 2%. With naïve stochastic rounding, every technique draws an independent random number to decide whether to round the number of samples up or down. Looking at the sampling process as a whole, the decision is not whether 1 sample is taken or not, but rather whether n_t samples are taken, each individually. With an unlucky draw none of the allocations are rounded up, whereas with a lucky draw, all of them are rounded up. We instead only draw a *single* random number to decide the rounding of *all* techniques (see alg. 1). After a technique was rounded, the probability of rounding up the next sample increases or decreases with whether the previous ones were rounded down or up, respectively. Overall, this leads to the random process only deciding between taking 1 sample more or less, instead of n_t samples, which reduces variance slightly.

ALGORITHM 1: Low Discrepancy Stochastic Rounding

```
/* Given stochastic number of samples $\beta_t \in \mathbb{R}^+_0$            */
$r \leftarrow random()$ ;        /* Uniform random number in $[0,1]$ */
for $t = 1$ to $n_t$ do     /* Round all $n_t$ sample allocations */
    $\gamma_t \leftarrow \lfloor\beta_t + r\rfloor$ ; /* $\beta_t \in \mathbb{R}^+_0$; stochast. rounded $\gamma_t \in \mathbb{N}$ */
    $r \leftarrow r + \beta_t - \gamma_t$ ; /* Remaining fractional random part */
end
/* Use each $\gamma_t$ as number of samples for technique $t$.  */
```

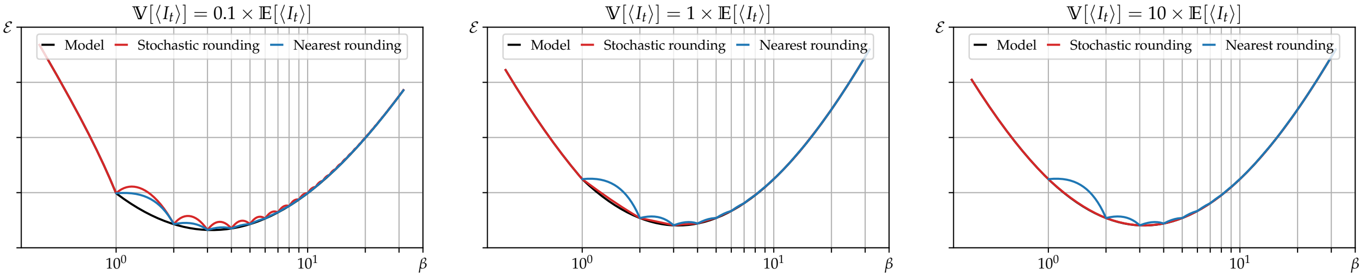
B.2 Convergence

In addition to the rendered images that show an improvement over state-of-the-art methods and the visualizations of where which technique is used, we provide efficiency and sample allocation plots over rendering time. In fig. 9, we present efficiency and sample allocation plots of a small selection of scenes in path guiding as well as in bidirectional path tracing. We can observe the curves flattening, or being near constant at the end of the rendering process, indicating a sufficiently converged sample-allocation fixed point in our proxy model.

C ADDITIONAL RESULTS

C.1 1D Examples

Apart from verifying the adequacy of our proxy model in complex rendering applications, we also investigate its performance and deviation from the true efficiency in a number of 1D examples visualized in fig. 10. While our proxy model matches exactly for the budget-unaware case, its optimum can deviate from the true optimal solution for budget-aware weights. This is most pronounced when the optimal solution is to discard all but one distribution, either because one strategy samples perfectly (second last row) or the problem becomes non-convex as individual strategies perform better than the MIS combination (last row).

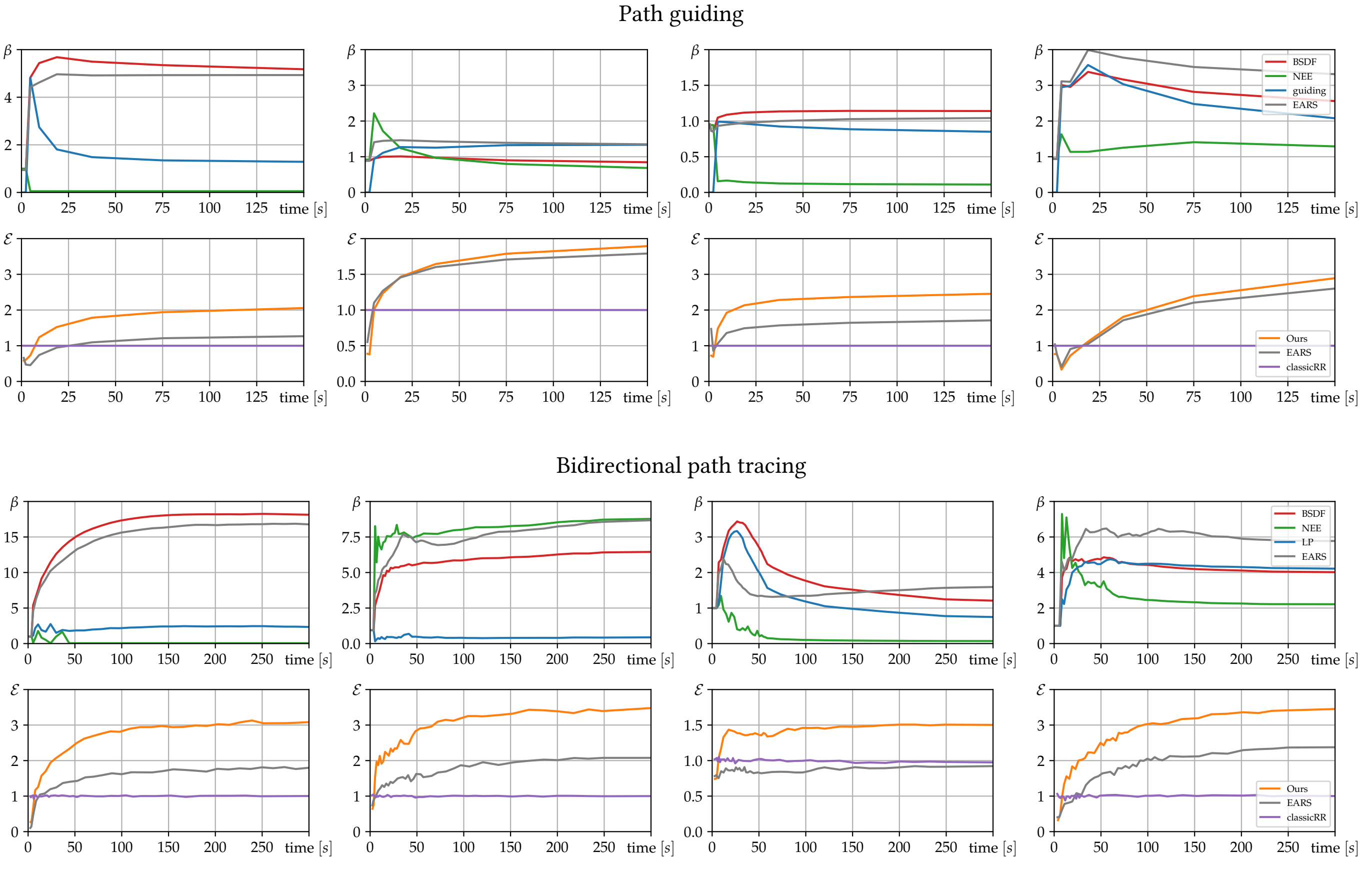


(a) Low variance

(b) Medium variance

(c) High variance

Fig. 8. We compare the efficiency of stochastic rounding, rounding to nearest integers, and the simplified model for the secondary estimator variance our work has in common with previous works. The performance of stochastic rounding depends on the ratio of estimator variance to expected value. All models perform the same for RR ($\beta < 1$) and at integer splitting factors. **(a)** For low variance estimators, stochastic rounding can introduce more noise than nearest rounding. **(b)** When the estimator variance is on the same order of magnitude as its expected value, stochastic rounding becomes the best option. **(c)** Stochastic rounding and the simplified model begin to agree perfectly when the variance is high.



(a) Glossy Kitchen

(b) Dining Room

(c) Glossy Bathroom

(d) Bookshelf

Fig. 9. We show convergence of the average primary hit point sample allocation (first row) and overall efficiency (second row, higher is better) in four selected scenes for path guiding and BDPT. The evaluation setup is the same as in the main text. Our fixed-point scheme quickly converges towards solutions that are better than the baseline and previous state-of-the-art.

C.2 Mixture Optimization in Path Guiding

In path guiding, we optimize the BSDF/guiding ratio of previous work using the KL divergence extended to image contributions [Rath et al. 2020]. This cannot directly maximize efficiency because **(1)** it does not contain information about cost, **(2)** only optimizes decisions locally rather than balancing variance across pixels and regions, and **(3)** optimizes KL divergence instead of variance. While **(1)** and **(2)** are fundamental limitations of the approach, **(3)** is in theory addressed by the variance optimizing variant [Vorba et al. 2019], but performed poorly across our tests scenes–likely due to the Adam optimizer drastically reducing its step-size in the presence of strong outliers. Hence, as suggested by the original authors, we use the KL divergence instead. A brief selection of scenes rendered for 5 minutes showcasing minimizing for **(a)** KL divergence, **(b)** KL divergence with image contributions, **(c)** variance, and **(d)** variance with image contributions can be found in fig. 11.

C.3 Additional Renders

For brevity, we have only included the most interesting scenes in our main text. We show results for all other scenes from our test set in fig. 12 (for path guiding) and fig. 13 (for BDPT). As with the scenes shown in the paper, our method consistently outperforms the respective previous state-of-the-art.

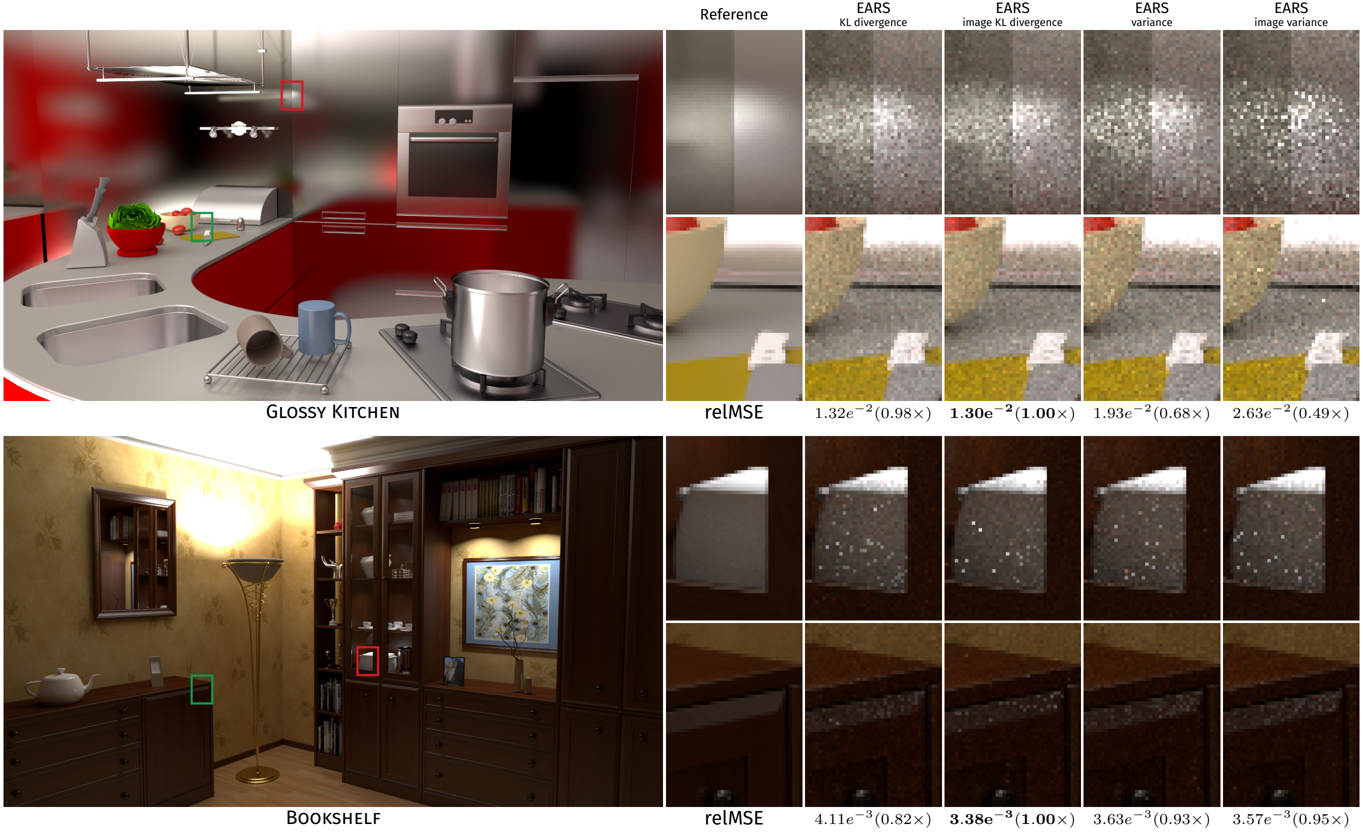
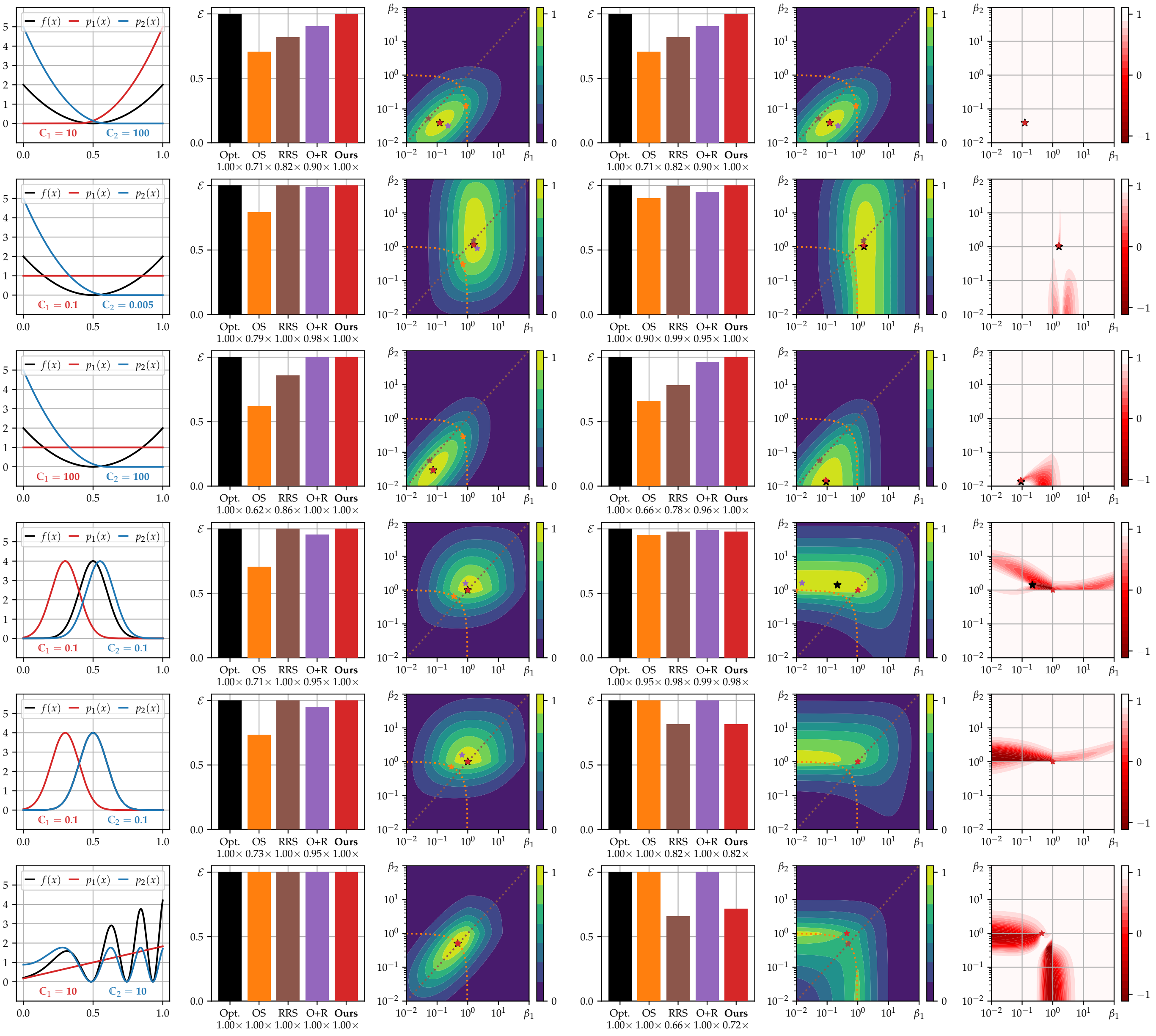


Fig. 11. We compare variants of gradient descent for the BSDF and guiding sample mixture on top of EARS in guiding: **(a)** KL divergence, **(b)** KL divergence of image contributions, **(c)** variance, and **(d)** variance of image contributions. Across our set of test scenes, KL divergence with image contributions performs best.



(a) Integrand (b) Budget-unaware balance heuristic (c) Budget-aware balance heuristic (d) Proxy error

Fig. 10. **(a)** We evaluate the performance of our model on simple 1D examples, in which a single integrand f is estimated by two techniques p_1 and p_2, the costs of which are indicated by $\mathbb{C}_1$ and $\mathbb{C}_2$ ($\mathbb{C}_\Delta = 1$ and $\mathbb{V}_\Delta = 1$ across all examples). **(b)** We compare the efficiency of four approaches: Optimal mixture sampling (“OS”), optimal Russian roulette and splitting (“RRS”), combining the two (“O+R”), and our proxy model (“Ours”). Note that combining optimal mixture sampling and RRS does not yield optimal performance. Our model optimizes ratios and total sample counts jointly, resulting in optimal performance for budget-unaware MIS weights. **(c)** For budget-aware MIS weights, the optimum predicted by our model can deviate from the true efficiency optimum. **(d)** We investigate the error of our proxy by plotting the dot product of the true gradients and our proxy gradients (“1” indicates a perfect match, and “-1” indicates opposing directions).

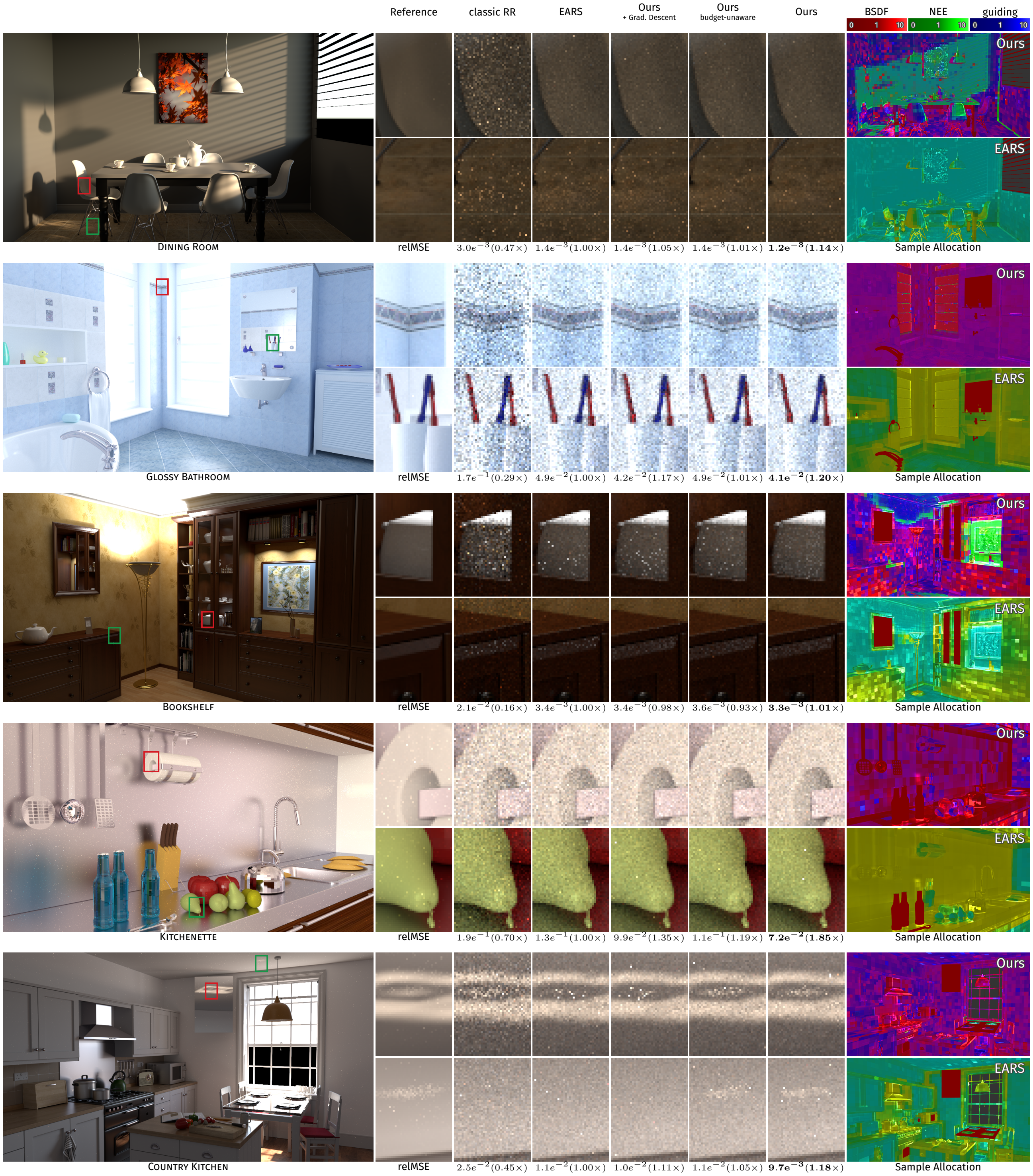


Fig. 12. We compare our method against state-of-the-art EARS applied to guiding in the 5 scenes not provided in the paper. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and guiding samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in the main text. Our "+ Grad. Descent" test uses gradient descent for the BSDF/guiding mixture and our method to combine it with NEE using budget-aware weights.

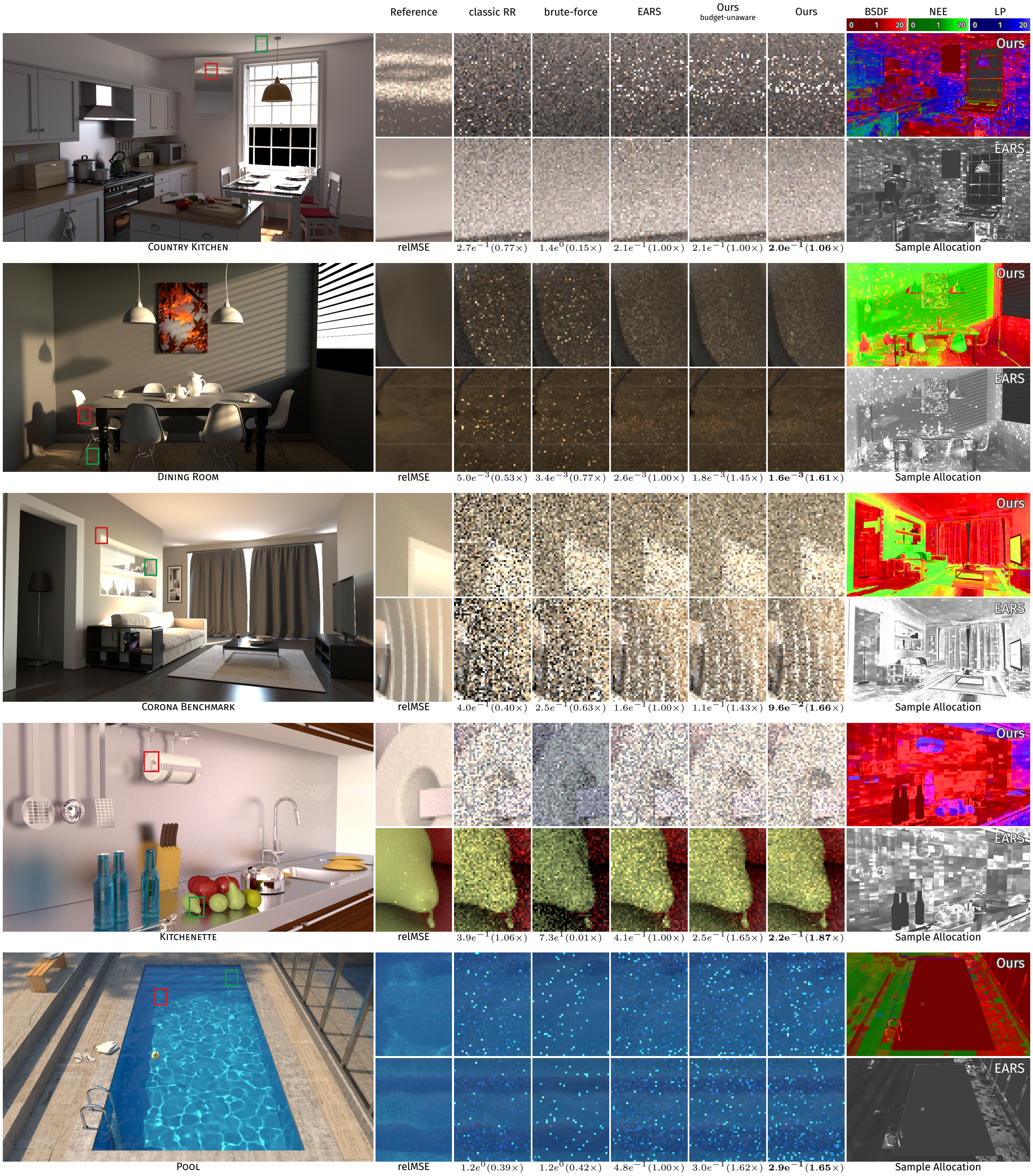


Fig. 13. We compare our method against state-of-the-art EARS applied to BDPT in the 5 scenes not provided in the paper in a 5 minute equal-time comparison. The numbers below the crops correspond to the relative mean squared error (relMSE, lower is better), with the speedup compared against EARS in parentheses (higher is better). To the right, we visualize the allocation decisions EARS and our method take. The color channels red, green and blue correspond to BSDF, NEE, and light path samples, respectively, with brighter colors indicating higher sample counts. The other 5 scenes we tested are provided in the main text.