Robust Fitting of Parallax-Aware Mixtures for Path Guiding

LUKAS RUPPERT*, University of Tübingen, Germany
SEBASTIAN HERHOLZ*, University of Tübingen, Germany and Intel Corporation
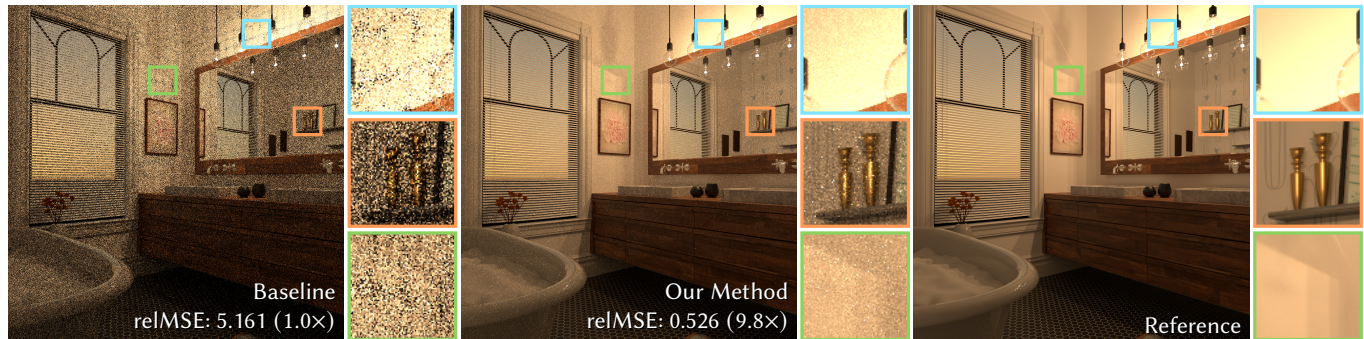HENDRIK P. A. LENSCH, University of Tübingen, Germany

Fig. 1. BATHROOM scene rendered using our guided path tracer implementation with next-event estimation in 10 minutes each. The "baseline" (left) corresponds to a configuration similar to previous guiding approaches. "Our method" (middle) features significantly improved direction sampling in local path guiding by introducing a robust fitting procedure for parallax-aware mixtures, achieving a speedup of 9.8 in this scene.

Effective local light transport guiding demands for high quality guiding information, i.e., a precise representation of the directional incident radiance distribution at every point inside the scene. We introduce a parallax-aware distribution model based on parametric mixtures. By parallax-aware warping of the distribution, the local approximation of the 5D radiance field remains valid and precise across large spatial regions, even for close-by contributors. Our robust optimization scheme fits parametric mixtures to radiance samples collected in previous rendering passes. Robustness is achieved by splitting and merging of components refining the mixture. These splitting and merging decisions minimize and bound the expected variance of the local radiance estimator. In addition, we extend the fitting scheme to a robust, iterative update method, which allows for incremental training of our model using smaller sample batches. This results in more frequent training updates and, at the same time, significantly reduces the required sample memory footprint. The parametric representation of our model allows for the application of advanced importance sampling methods such as radiance-based, cosine-aware, and even product importance sampling. Our method further smoothly integrates next-event estimation (NEE) into path guiding, avoiding importance sampling of contributions better covered by NEE. The proposed robust fitting and update scheme, in combination with the parallax-aware representation, results in faster learning and lower variance compared to state-of-the-art path guiding approaches.

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

Additional Key Words and Phrases: Global Illumination, Ray Tracing, Sampling and Reconstruction, Stochastic Sampling

1 INTRODUCTION

Path-based rendering algorithms have become the industry standard for solving light transport simulations [Fascione et al. 2017, 2018, 2019]. Especially uni-directional path tracing [Kajiya 1986] is now the favorable choice, due to its simplicity to implement and its extensibility to incorporate different sampling strategies or estimators (e.g., next-event estimation). The basic path tracing algorithm achieves a significant gain in quality and efficiency with proper importance sampling, which for global illumination needs to consider both the distribution of the scattering function as well as an approximation of the true radiance distribution. Building on initial work of Jensen [1995] and Lafortune and Willems [1995], recent path guiding approaches, e.g. [Vorba et al. 2014; Müller et al. 2017], use representations of the 5D-radiance distribution that are either learned form a pre-processing step or online during rendering (a.k.a., forward-learning). Approaches based on forward-learning are preferred in practice, as their ability to provide previews more quickly can improve the turnaround times of artists. With local path guiding, even complex light transport effects such as caustics and multiple diffuse bounces can be rendered reliably using simple forward path tracing and, thus, it is now used in production rendering [Vorba et al. 2019].

As an imprecise guiding function generally increases variance rather than reducing it, both the chosen representation and a robust

fitting procedure are of utmost importance. Local deviations in the fitting quality of the estimated radiance distributions can already result in visible artifacts (e.g., Fig. 2). Different representations of the local incident radiance distribution have been proposed, each with its pros and cons. Discretized representations of the spherical incident radiance distribution, e.g., histograms or quadtrees, are usually easy to fit but are not flexible enough to support other sampling techniques (hemispherical importance sampling or product sampling) at reasonable cost. Parametric representations, such as Gaussian or von Mises-Fisher mixture models (GMMs or VMMs), on the other hand support very efficient importance sampling closed-form solutions for a wide variety of tasks, including calculating product mixtures and their integrals. Unfortunately, fitting such a parametric mixture model (PMM) robustly from a set of samples or even online is challenging.

To account for spatial variation, a spatial subdivision scheme (e.g., kd-tree) typically divides the scene into regions, where each region contains a static approximation of the local incident radiance distribution. Lacking any spatial representation in the used models, existing methods average the incident radiance over the spatial domain covered by the region and thus require many small regions with well-fit distributions in areas with large spatial variation (e.g., in the presence of close-by emitters).

In this paper, we make contributions in three main directions:

First, we propose a robust, variance-driven fitting procedure for parametric mixture models, in our case using von Mises-Fisher distributions, given radiance samples that are collected during rendering. We derive efficient criteria for stepwise splitting and merging of mixture components [Wang et al. 2004] that minimize the variance of the final importance-driven estimator. Using these criteria, insufficient fits produced by the traditional EM-algorithm are detected and improved upon, escaping local maxima and automatically choosing a suitable number of components to faithfully represent the directional distribution without having to restart the EM-algorithm with improved initialization (Sec. 4.1).

Second, we extend our fitting procedure towards an incremental learning approach (Sec. 4.2) that reuses, adjusts and updates the representations from previous iterations. Our method, therefore, does not rely on an exponential growth of training samples and can update the guiding representations more frequently, leading to a faster effective learning of the complete radiance field of the scene. This approach generates high-quality sampling distributions in a PMM-based guiding framework using the preferable forward-learning scheme while requiring only modest amounts of memory and computational resources.

Third, we introduce spatial information into the directional distribution model in form of a parallax-compensating representation of the local incident radiance field based on vMFs, that warps the distribution according to the query point and the location of the contributing light source (Sec. 4.3). As the warped distributions stay precise for larger cells, accurate guiding towards spatially varying illumination, including reflected and refracted contributions, is available early on, resulting in superior accuracy during rendering. This further accelerates the training process. Artifacts at the borders of spatial cells caused by deviations from marginalized distributions are avoided entirely (e.g., Fig. 1).

In addition, we present an extension to handle direct light contributions in the presence of a next-event estimator (NEE). The extension reweights the direct light contribution in the gathered sample data, to avoid sampling contributions which are covered well by NEE, and, at the same time, focuses on the direct light contributions which are not covered well by NEE. Our presented framework does not only support accurate incident radiance-based guiding, outperforming state-of-the-art techniques (Sec. 6.3), but is also capable of product importance sampling using approximations of the cosine and BRDF terms.

2 RELATED WORK

2.1 Monte-Carlo-based Light Transport Algorithms

Simulating global light transport involves solving the high-dimensional integral defined by the rendering equation (RE) [Immel et al. 1986; Kajiya 1986], which, in practice, is evaluated by path-based Monte-Carlo algorithms. The simplest of these algorithms is the unidirectional path tracer (PT) [Kajiya 1986], which generates random walks (i.e., paths), that explore the light transport of a scene, starting from the camera. Random walks are constructed iteratively by performing local directional sampling decisions at every path intersection with the scene. In path tracing, this decision most often just considers the BSDF, leading to ineffective sampling when its distribution deviates from the distribution of its product with the incident illumination (e.g., caustics or high-frequency indirect illumination).

A more efficient way to evaluate complex effects like caustics or strong indirect illumination, where light sources are hard to reach, is to reverse the random walk direction using a light tracing (LT) algorithm [Arvo et al. 1986; Dutré et al. 1993]. The advantages of both can be combined into a bidirectional path tracer (BDPT) [Lafortune and Willems 1993; Veach and Guibas 1995a], where the individual sub-paths of PT and LT are connected using multiple importance sampling (MIS) [Veach and Guibas 1995b]. By reformulating the path sampling framework, unified path sampling (UPS) [Hachisuka et al. 2012] and vertex connection and merging (VCM) [Georgiev et al. 2012] seamlessly combine the advantages of BDPT with the ones from particle density based approaches, such as photon mapping [Jensen 1996, 2001]. As shown by Grittmann et al. [2018] and Šik and Křivánek [2019], the computational overhead of bidirectional methods can be decreased by guiding the light path emission, as well as the light path traversal, into important regions (e.g., caustics). Nevertheless, the additional implementational complexity and the strict requirement on the bidirectionality of the modeled light transport, makes bidirectional methods less favorable in practice. Therefore, most production rendering systems nowadays are based on unidirectional path tracing [Fascione et al. 2017, 2018, 2019].

2.2 Path Guiding

The results of recent attempts to perform importance sampling based on previously gathered information about the light transport revived the interest in so called path-guiding methods and made them even applicable in production environments [Vorba et al. 2019]. These methods can typically be categorized into *local* or *global* ones.

Local path guiding methods explore an approximation of the local incident radiance distribution to guide the directional sampling decision at every path intersection point. These approaches mainly differ in the models used to represent the 5-dimensional incident radiance field of a scene and in the way these models are trained. Early approaches [Jensen 1995; Lafortune and Willems 1995] build histogram-based guiding distributions at every path intersection, where the information about the incident radiance either originates from a photon map [Jensen 1995] or a 5-dimensional tree structure [Lafortune and Willems 1995]. Other approaches directly use particle footprints [Hey and Purgathofer 2002], cosine lobes [Bashford-Rogers et al. 2012], B-splines [Pegoraro et al. 2008], Gaussian mixture models [Vorba et al. 2014; Herholz et al. 2016], quad-trees [Bus and Boubekeur 2017; Müller et al. 2017], reinforcement learning [Dahm and Keller 2017], and even neural networks [Müller et al. 2019] to build the guiding distribution. The resulting guiding distribution can either be proportional to the incident radiance or the full integrand of the rendering equation. The latter option however requires an on-the-fly construction of the product guiding distribution at every intersection point. Based on the used representation (e.g., histograms, quad-trees, or cosine lobes) these constructions are often costly when precise or only lead to a rough, inaccurate approximation of the product distribution. Using GMM-based representations of the incident radiance and BRDF lobe distribution, Herholz et al. [2016] efficiently evaluate the exact GMM of the product distribution using a closed-form solution. Local path guiding methods are even capable of optimizing all necessary sampling decisions in the presence of participating media, e.g., scattering, direction, distance, termination, and splitting [Herholz et al. 2019]. Our robust fitting procedure and the parallax-compensation scheme can be smoothly integrated into existing PMM-based guiding methods.

Global path guiding approaches, on the other hand, guide the full path construction process at once. A typical attempt to guide the complete paths is to train guiding distributions in the *primary sample space* (PSS), which can either be modeled using high-dimensional kd-trees [Guo et al. 2018] or by the use of *neural networks* [Zheng and Zwicker 2019; Müller et al. 2019]. Unfortunately, all PSS-based guiding approaches suffer from the curse of dimensionality, which prevents them from learning robust guiding distributions for higher dimensions [Zheng and Zwicker 2019]. Selective path guiding [Reibold et al. 2018] uses a cache of high variance paths, which are collected during rendering to guide new paths based on similar paths from the cache. Since the efficiency of this approach strongly relies on correlations between the selected guiding paths, the method struggles at multiple diffuse or glossy interactions.

2.3 Robust Fitting of Parametric Mixture Models

The EM-algorithm [Dempster et al. 1977; McLachlan and Krishnan 2007] enables the fitting of parametric mixture models (PMMs) using a two-step iterative approach consisting of the *expectation* (E) and the *maximization* (M) step. It has seen widespread adoption in the statistics and machine learning communities and new, specialized PMMs and accompanying variations of the EM-algorithm are being researched to this day [McLachlan et al. 2019]. Using the traditional

approach, the EM algorithm iteratively increases the log-likelihood of a dataset by adjusting the PMM's parameters. However, this maximization of the log-likelihood is not concave in the parameter space and therefore the global optimum can only be reached by chance. Another limitation is that the number of components in the PMM has to be estimated in advance and cannot be changed.

In search of the best possible fit, the EM algorithm can simply be restarted sufficiently many times, while the number of components can be determined using some information criteria (e.g. BIC, AIC [Schwarz 1978; Akaike 1974]) that balance goodness of fit against model complexity. While costly, this approach is common to find in machine-learning literature [Bishop 2006], as it has good chances of determining the best possible fit for a single set of sample data.

3 BACKGROUND ON LOCAL PATH GUIDING

The light transport of a scene is modeled via the rendering equation (RE) as presented by Immel et al. [1986] and Kajiya [1986]:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \underbrace{\int_\Omega f_s(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) |\cos \theta_i| \, d\omega_i}_{L_r(\mathbf{x}, \omega_o)}, \quad (1)$$

which describes the outgoing radiance L_o at point $\mathbf{x}$ in the direction ω_o as the sum of the emitted radiance L_e and the reflected radiance L_r. It is typically solved by implementing a Monte-Carlo estimator for the reflected radiance $\langle L_r \rangle$:

$$\langle L_r(\mathbf{x}, \omega_o) \rangle = \frac{f_s(\mathbf{x}, \omega_o, \omega_i) \langle L_i(\mathbf{x}, \omega_i) \rangle |\cos \theta_i|}{p(\omega_i \mid \mathbf{x}, \omega_o)}, \quad (2)$$

which, in path tracing, is recursively evaluated for a random walk following a random direction ω_i, distributed according to a directional sampling PDF p. The variance, i.e., the expected squared error, of random walk-based estimators, such as PT, recursively depends on the variance of each nested local estimator $\langle L_r \rangle$.

Only considering the BSDF during importance sampling, the variance will usually be rather high. Local path guiding methods reduce variance by incorporating previously observed information about the light transport into the directional sampling strategy used by each $\langle L_r \rangle$. This reduction of the local variances results in a globally improved path distribution. Based on sample data gathered during a pre-processing pass or earlier rendering iterations, local guiding distributions p_g are learned and combined via one-sample MIS [Veach and Guibas 1995b] with a defensive sampling strategy, such as BSDF importance sampling p_{f_s}:

$$p(\omega_i \mid \mathbf{x}, \omega_o) = (1 - \alpha) p_g(\omega_i \mid \mathbf{x}, \omega_o) + \alpha p_{f_s}(\omega_i \mid \mathbf{x}, \omega_o), \quad (3)$$

where α represents the probability of sampling according to p_{f_s}. Most commonly, α is set to 0.5 [Hesterberg 1995], preventing a strong variance increase when the guiding distribution is not yet reliable enough.

The ideal guiding distribution would be proportional to the L_r integrand (Eq. 1), considering all its factors (f_s, L_i, and $\cos \theta_i$). Unfortunately, realizing this is generally infeasible, since it requires learning and representing a 7-dimensional distribution. It is, therefore, common practice to learn an approximation of the local L_i-distribution from path traced samples. To cope with spatial variation, often a

static, spatially averaged approximation of the L_i-distribution is used for each region defined by a spatial subdivision structure.

The simplest and most popular guiding distribution p_g directly uses the learned L_i-approximation for sampling:

$$p_g(\omega_i \mid \mathbf{x}) \propto L_i(\mathbf{x}, \omega_i) \quad \text{or} \quad p_g(\omega_i \mid \mathbf{x}) \propto L_i(\mathbf{x}, \omega_i) |\cos\theta_i|. \quad (4)$$

Accounting for all factors in the L_r integrand requires building the product guiding distribution p_g on-the-fly at every path intersection. One efficient method [Herholz et al. 2016, 2019] represents approximations of the individual distributions of the incident radiance L_i and the BSDF lobe f_s by parametric mixture models (PMMs). It provides an exact and efficient closed-form solution for the conceptually optimal guiding distribution:

$$p_g(\omega_i | \mathbf{x}, \omega_o) \propto f_s(\mathbf{x}, \omega_o, \omega_i) L_i(\mathbf{x}, \omega_i) |\cos\theta_i|, \quad (5)$$

where the cosine term is included either in the L_i- or f_s-distribution.

However, a robust and efficient estimation of PMM parameters representing an incident radiance distribution from a set of photons or radiance samples is not trivial. We present a novel solution to robustly fit PMM parameters to approximate the L_i-distribution in Sec. 4.1. As demonstrated in Sec. 6.2, combined with even rough approximations of the BSDF, our approach already improves the sampling quality effectively.

3.1 Von Mises-Fisher Mixture Models – VMM

In our work, we make use of a specific PMM, the von Mises-Fisher mixture model (VMM), which is based on the *von Mises-Fisher* (vMF) distribution function [Fisher et al. 1987]. The vMF distribution is an isotropic probability distribution defined on $(d-1)$-dimensional spheres in the $\mathbb{R}^d$ space. In 3D space, Jakob [2012] defined the following numerically stable formulation of the vMF distribution:

$$v(\omega \mid \mu, \kappa) = \frac{\kappa}{2\pi(1 - \exp(-2\kappa))} \exp\left(\kappa\left(\mu^T\omega - 1\right)\right), \quad (6)$$

where μ describes the mean direction and κ the concentration of the isotropic probability density function. Since a single vMF lobe is clearly insufficient, the VMM uses a weighted sum of K individual vMF distributions to model complex directional distributions:

$$\mathcal{V}(\omega \mid \Theta) = \sum_{k=1}^{K} \pi_k \, v(\omega \mid \Theta_k). \quad (7)$$

As is common practice in PMM literature [Bishop 2006], we shorten the VMM's parameter set to $\Theta = \{\pi_1, \Theta_1, \ldots, \pi_K, \Theta_K\}$, which represents the individual mixture weights π_k and the associated vMF component's parameters $\Theta_k = \{\kappa_k, \mu_k\}$ in a concise manner. In order to model a valid PDF, the component weights π_k need to be positive and sum up to one.

A useful feature of PMMs is that one can derive closed-form solutions for sampling, the evaluation of the component-wise product distributions, as well as convolutions and their corresponding integrals [Vorba et al. 2014; Herholz et al. 2016, 2019]. We make use of the vMF's properties to develop efficient criteria for splitting and merging (Sec. 4.1), a Bayesian incremental update method (Sec. 4.2), and a parallax-aware representation of incident radiance (Sec. 4.3). While demonstrated on VMMs, the concepts can be generalized to other PMMs (Sec. 7).

4 METHODOLOGY

We present a robust framework for local, PMM-based path guiding, at the example of vMF mixtures (VMMs). The foundation of our guiding framework is an accurate VMM-based approximation $\mathcal{V}$ of the local incident radiance distribution at each point $\mathbf{x}$ in the scene:

$$L_i(\mathbf{x}, \omega_i) \propto \mathcal{V}(\omega_i \mid \Theta(\mathbf{x})), \quad (8)$$

where Θ refers to the VMM-parameters stored in the local region of a spatial subdivision structure (i.e., kd-tree) containing $\mathbf{x}$.

Our VMM-based representation supports a set of different path guiding strategies, which either only consider the radiance (Eq. 4, left), integrate the cosine term (Eq. 4, right), or even approximate the full product (Eq. 5). Each supported strategy increases the path sampling quality further, outperforming current state-of-the-art algorithms based on discretized representations (Fig. 8, Table 3).

For training, we adopt the forward-learning scheme presented by Müller et al. [2017], which focuses exploration entirely on camera paths and also allows for an immediate preview of the rendered image. In each training iteration, we render the scene to collect sample data. At each intersection of a random walk transport path, a sample $s_n \in S, \|S\| = N$, is generated, containing the position $\mathbf{x}_n$, the sampling direction ω_n, the corresponding local sampling PDF $p(\omega_n \mid \mathbf{x})$, and an MC estimate of the incident radiance $\tilde{L}_i(\mathbf{x}, \omega_n)$:

$$S = \{s_1, \ldots, s_N\}, \text{where } s_n = \{\mathbf{x}_n, \omega_n, p(\omega_n \mid \mathbf{x}_n), \tilde{L}_i(\mathbf{x}_n, \omega_n)\}. \quad (9)$$

We then partition the gathered samples into the regions defined by the kd-tree, subdividing it at the same time (Sec. 5). Finally, we fit the VMMs as described in the following subsections, drop the sample data, and iterate.

We first present an adaptive variant of the weighted EM (wEM) [Vorba et al. 2014] in a batch-EM setting, suitable for an exponential training approach, where the number of samples is doubled each iteration and distributions are fitted from scratch (Sec. 4.1). We measure the approximation quality of the VMM to the true L_i-distribution based on the normalized expected variance (NEV) of the single-sample MC estimate of fluence $\langle\Phi(\mathbf{x})\rangle$ within the region:

$$\mathbb{V}\left[\frac{\langle\Phi(\mathbf{x})\rangle}{\Phi(\mathbf{x})}\right] = \frac{1}{\Phi^2(\mathbf{x})}\left(\mathbb{E}\left[\langle\Phi(\mathbf{x})\rangle^2\right] - \mathbb{E}\left[\langle\Phi(\mathbf{x})\rangle\right]^2\right), \quad (10)$$

$$\text{where} \quad \langle\Phi(\mathbf{x})\rangle = \frac{L_i(\mathbf{x}, \omega_i)}{\mathcal{V}(\omega_i \mid \Theta(\mathbf{x}))} \approx \int_\Omega L_i(\mathbf{x}, \omega_i) \, d\omega_i, \quad (11)$$

using a random direction vector ω_i, distributed according to $\mathcal{V}(\Theta(\mathbf{x}))$. Based on the NEV, we develop efficient criteria for stepwise splitting and merging [Wang et al. 2004], which adapts the local distribution of mixture components after the initial fit, such that it accurately and efficiently represents the observed sample data.

We extend the adaptive fitting approach by a Bayesian incremental update scheme, which allows for training from small batches of sample data (Sec. 4.2). This approach significantly reduces the turnaround times between gathering sample data and using their information for guided sampling, speeding up the training process. Also, the amount of memory required to store each training iteration's samples (e.g., from 4spp) is limited and does not increase.

Finally, we extend the VMM's parameters to allow for a parallax-aware representation of incident radiance that remains valid and precise across larger spatial regions even in the presence of close-by contributors, including reflected and refracted contributions (Sec. 4.3). This further reduces the variance in each region and allows our method to maintain larger regions with more samples available to each fit.

In the following, we only consider the VMM for a single spatial region and therfore omit the depenecy of Θ on $\mathbf{x}$ from the notation.

4.1 Variance-Based Adaptive Weighted Batch-EM Fitting

The accuracy of local incident radiance approximations is an important factor in the sampling quality of any guiding approach (see Fig. 2). In practice, these approximations are the result of a fitting process using the often scarce and noisy sample data generated by a previous rendering iteration, gathered in small spatial regions of the scene. In the following, we first review the weighted EM (wEM) algorithm [Vorba et al. 2014], and then present our extension, that achieves robust parameter estimation by adapting stepwise splitting and merging [Wang et al. 2004], introducing our analytic, variance-based split and merge criteria.

Fitting by Weighted Expectation-Maximization. Traditional applications of fitting the parameter set Θ of a VMM $\mathcal{V}$ to a batch of N samples $s_n \in S$ with directions ω_n use the EM algorithm [Dempster et al. 1977; McLachlan and Krishnan 2007] to estimate the unknown density of the sample data. The weighted EM approach [Vorba et al. 2014] adds a weight w_n to each sample s_n, to adapt the actual distribution of the samples. After random parameter initialization, the algorithm iterates over the complete batch of weighted samples S multiple times, alternating between the *expectation* (E) and *maximization* (M) steps until convergence. Each iteration results in a greater or equal log-likelihood $\mathcal{L}$:

$$\hat{\Theta} = \arg\max_{\Theta} \; \mathcal{L}(S) = \sum_{n=1}^{N} w_n \log \mathcal{V}(\omega_n \mid \Theta). \quad (12)$$

During the *expectation*-step, each sample is partially assigned to each component using the following *soft-assignment* function γ_k:

$$\gamma_k(\omega_n) = \frac{\pi_k \, v(\omega_n \mid \Theta_k)}{\mathcal{V}(\omega_n \mid \Theta)}, \quad (13)$$

which represents the probability of the sample direction ω_n being generated by the k-th mixture component.

In the following *maximization*-step, the parameters of each component are re-estimated using their corresponding weighted maximum likelihood estimators (wMLEs). For the k-th vMF distribution, the wMLE of the updated parameters $\hat{\Theta}_k = \{\hat{\mu}_k, \hat{\kappa}_k\}$ is based on the *sufficient statistics* vector r_k and the mean cosine $\bar{r}_k$:

$$r_k = \sum_{n=1}^{N} w_n \gamma_k(\omega_n)\omega_n, \quad \bar{r}_k = \frac{\|r_k\|}{\sum_{n=1}^{N} w_n \gamma_k(\omega_n)}. \quad (14)$$

Using these two parameters, $\hat{\mu}_k$ and $\hat{\kappa}_k$ of the distribution are evaluated as follows:

$$\hat{\mu}_k = \frac{r_k}{\|r_k\|} \quad \text{and} \quad \hat{\kappa}_k \approx \frac{3\bar{r}_k - \bar{r}_k^3}{1 - \bar{r}_k^2}, \quad (15)$$

where $\hat{\mu}_k$ is the normalized version of r_k and $\hat{\kappa}_k$ is approximated according to Banerjee et al. [2005]. The new estimate for the k-th component weight $\hat{\pi}_k$ resolves to:

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N} w_n \gamma_k(\omega_n)}{\sum_{j=1}^{K} \sum_{n=1}^{N} w_n \gamma_j(\omega_n)}, \quad \text{with} \quad w_n = \frac{\tilde{L}_i(\mathbf{x}_n, \omega_n)}{\tilde{\Phi}(\mathbf{x}) p(\omega_n \mid \mathbf{x}_n)} \quad (16)$$

$$\text{and} \quad \tilde{\Phi}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} \frac{\tilde{L}_i(\mathbf{x}_n, \omega_n)}{p(\omega_n \mid \mathbf{x}_n)}. \quad (17)$$

Our objective is to fit the VMM $\mathcal{V} \propto L_i$ from a batch of samples where the PDF of each sample may be arbitrary but is known to be the previously used sampling strategy (e.g., BSDF importance sampling, guiding, or their MIS combination (Eq. 3)). To adjust the density of the samples towards a distribution proportional to the radiance distribution, the per-sample weight w_n is set as the ratio between the normalized incident radiance estimate and the sampling probability density function. With these weights, the VMM becomes approximately proportional to the incident radiance:

$$\mathcal{V}(\omega_i \mid \Theta) \approx w \cdot p(\omega_i \mid \mathbf{x}) \propto \tilde{L}_i(\mathbf{x}, \omega_i). \quad (18)$$

However, the outcome of the wEM heavily depends on the parameter initialization and the chosen number of modes in the VMM. Underfitting a multimodal radiance distribution by a single vMF lobe (mode collapse) will underestimate the true peaks during importance sampling and might therefore increase the variance, which can lead to high-intensity outliers (a.k.a. fireflies). It is common practice in ML, to overcome this problem of the EM algorithm by restarting it multiple times, picking the fit with the highest log-likelihood, and then to repeat this whole process to choose an adequate number of mixture components. Unfortunately, we cannot afford to perform such a costly procedure in our rendering framework. Instead, we address these issues by applying stepwise splitting and merging (SSMEM) [Wang et al. 2004], for which we derive efficient criteria that minimize the variance of the resulting estimator.

Variance-based split and merge. Splitting and merging of components as part of the EM-fit allows us to escape the local optima and simultaneously select a suitable number of components to accurately represent sample data without overfitting. The introduced merge-criterion C_{merge} measures the loss in approximation quality incurred by merging two components. The split-criterion C_{split} measures how well the component's distribution model matches the observed data, to detect cases of underfitting (i.e., collapsed modes). As illustrated by Fig. 2, we start with a regular wEM fit, followed by a merging step and a splitting step which includes a partial wEM fit [Ueda et al. 2000a,b] of the newly created components.

Our criteria for splitting and merging are based around the Pearson χ^2-divergence [Neyman and Pearson 1933]. As previously shown [Jona-Lasinio et al. 1999; Müller et al. 2019], it corresponds to the normalized estimator variance (NEV) that results from using the PDF q for sampling instead of the ideal PDF p:

$$D_{\chi^2}\left(p_{L_i} \,\|\, q_{\mathcal{V}}\right) = \int_{\Omega} \frac{\left(p_{L_i}(\omega) - q_{\mathcal{V}}(\omega)\right)^2}{q_{\mathcal{V}}(\omega)} \, d\omega = \mathbb{V}\left[\frac{\langle \Phi(\mathbf{x})\rangle}{\Phi(\mathbf{x})}\right],$$
$$\text{where} \quad p_{L_i}(\omega) = \frac{L_i(\mathbf{x}, \omega)}{\Phi(\mathbf{x})} \quad \text{and} \quad q_{\mathcal{V}}(\omega) = \mathcal{V}(\omega \mid \Theta). \quad (19)$$
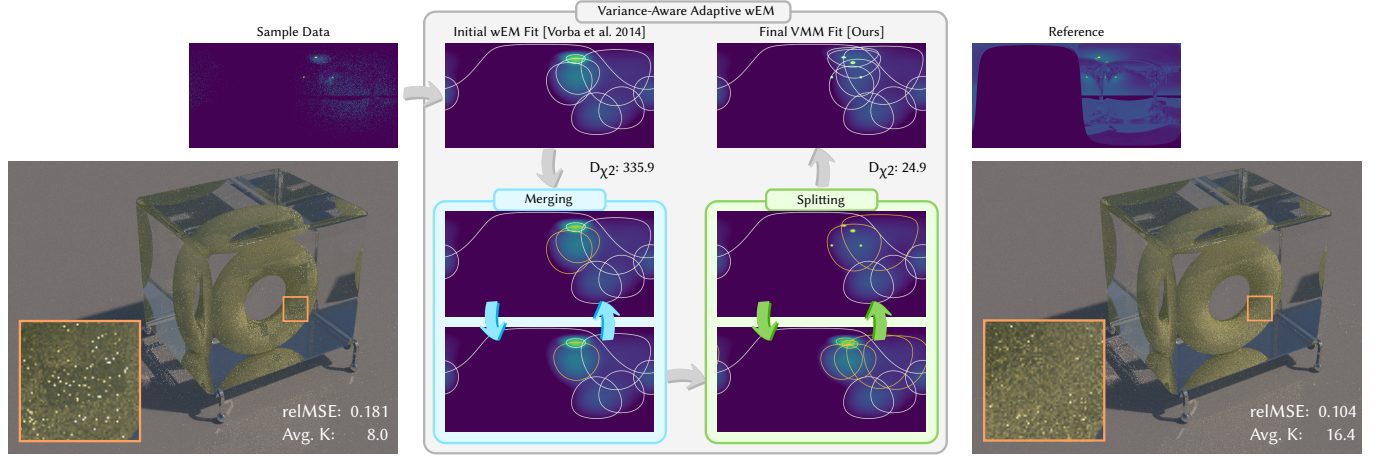
Fig. 2. Illustration of our variance-based adaptive fitting procedure at the example of the Torus scene. The sample data contains multiple small reflections of the sun. In a weighted EM fit, mode collapses can occur and a blurry fit with high divergence (335.9) results. By iterative merging, followed by iterative splitting with a partial EM fit performed on split components, these issues can be detected and resolved. The result is a high-quality fit with much lower divergence (24.9). The image on the left has been rendered without splitting and merging, while the image on the right has been rendered with splitting and merging.

Splitting. The splitting operation improves the approximation quality of the mixture locally by adding another component in regions where a fitted mixture component does not represent the observed sample data with sufficient accuracy (e.g., mode collapse). We identify which of the existing components to split by evaluating its normalized estimator variance (NEV) (i.e., Pearson χ^2-divergence), scaled by the component weight π_k, as the splitting criterion C_{split}. It compares the partial distribution $p_{\text{L}_i,k}$ of p_{L_i} that should be represented by the k-th VMM component to the actually fitted vMF distribution $q_{v,k}(\omega) = v(\omega \mid \Theta_k)$:

$$C_{\text{split}}(k \mid \Theta) = \pi_k D_{\chi^2}\left(p_{\text{L}_i,k} \,\middle\|\, q_{v,k}\right), \quad (20)$$

where $p_{\text{L}_i,k}$ is obtained from p_{L_i} by multiplying by the fraction of the soft-assignment term γ_k (Eq. 13) and the component weight π_k:

$$p_{\text{L}_i,k}(\omega) = \frac{\gamma_k(\omega)L_i(\mathbf{x}, \omega)}{\pi_k \Phi(\mathbf{x})} = \frac{v(\omega \mid \Theta_k)L_i(\mathbf{x}, \omega)}{\mathcal{V}(\omega \mid \Theta)\Phi(\mathbf{x})}. \quad (21)$$

C_{split} is evaluated at little additional cost by reusing the sample data S from the wEM fitting step. Since each sample s_n contains the direction ω_n, the estimate $\tilde{L}_i$ of incident radiance and the PDF $p(\omega_n)$ of generating ω_n, we can build a Monte-Carlo estimator $\langle C_{\text{split}} \rangle$ for the split criterion, which can be simplified to the following equation:

$$\langle C_{\text{split}}(k \mid \Theta) \rangle = \pi_k \left(\frac{1}{N\tilde{\Phi}^2(\mathbf{x})} \sum_{n=1}^{N} \frac{v(\omega_n \mid \Theta_k)\tilde{L}_i^2(\mathbf{x}_n, \omega_n)}{\mathcal{V}^2(\omega_n \mid \Theta)p(\omega_n \mid \mathbf{x}_n)} - 1 \right). \quad (22)$$

To bound the amount of normalized variance added by each component, we split if $\langle C_{\text{split}}(k \mid \Theta) \rangle$ exceeds our splitting threshold t_{split}. The split operation replaces the original component by two new components, where we use principal component analysis to favorably initialize their respective initial mean directions (App. A.4). These new components undergo a partial wEM step [Ueda et al. 2000a,b], to fit the sample data previously assigned to the original

component. The other components are kept fixed during this step. We subsequently re-evaluate the splitting criterion to check if any component needs to be split again.

Merging. The objective of merging is to first avoid overfitting, and second, to increase the computational efficiency of the model. If too many components are available, EM-based algorithms tend to either overfit the observed data or to generate multiple components with the same or similar shape only differing in weights. Striving for an efficient MC estimator, components should only be merged when the variance increase is insignificant compared to the increase in evaluation efficiency. Similar to previous work, a recursive component reduction strategy identifies the cost of merging for all possible pairs of components and then merges the combination with the lowest cost. We define the cost of merging the pair of components (i, j) into one component k by measuring the increase in normalized variance using the Pearson χ^2-divergence between the original mixture $\mathcal{V}(\Theta)$ and the merged mixture $\mathcal{V}(\Theta')$:

$$D_{\chi^2}\left(\mathcal{V}(\Theta) \,\middle\|\, \mathcal{V}(\Theta')\right)$$
$$= \int_{S^2} \frac{(\pi_i\, v(\omega|\Theta_i) + \pi_j\, v(\omega|\Theta_j) - \pi_k'\, v(\omega|\Theta_k'))^2}{\mathcal{V}(\omega|\Theta')} \, d\omega, \quad (23)$$

where $\pi_k' = \pi_i + \pi_j$. The parameters Θ_k' of the merged component are computed in closed form (App. A.3). By concentrating just on the involved components, C_{merge} is formulated as the upper bound of the divergence:

$$D_{\chi^2}\left(\mathcal{V}(\Theta) \,\middle\|\, \mathcal{V}(\Theta')\right) \leq C_{\text{merge}}(i, j \mid \Theta)$$
$$= \int_{S^2} \frac{(\pi_i\, v(\omega|\Theta_i) + \pi_j\, v(\omega|\Theta_j) - \pi_k'\, v(\omega|\Theta_k'))^2}{\pi_k'\, v(\omega|\Theta_k')} \, d\omega. \quad (24)$$

As evaluating C_{merge} by a Monte-Carlo estimator involving the samples S for all $\frac{K^2-K}{2}$ pairs (i, j) would be too costly, we derive an efficient and exact closed-form solution for Eq. 24 that only involves

the parameters of the pair (i, j). Our solution makes use of two features of the vMF distribution: First, the closed-form solution for evaluating the integral of the product of two vMF lobes (App. A.1):

$$v(\Theta_i) \otimes v(\Theta_j) = \int_{S^2} v(\omega|\Theta_i) \, v(\omega|\Theta_j) \, d\omega. \tag{25}$$

Second, the fact that the distribution of the multiplicative inverse of a vMF lobe results in another, scaled vMF lobe (App. A.2):

$$v^{-1}(\omega|\mu, \kappa) = \frac{1}{v(\omega|\mu, \kappa)} = \frac{4\pi^2(1 - \exp(-2\kappa))^2}{\kappa^2 \exp(-2\kappa)} \, v(\omega|-\mu, \kappa). \tag{26}$$

Using these two features, the closed-form solution for Eq. 24 is obtained as the sum of several integrated vMF products:

$$C_{\text{merge}}(i, j \mid \Theta) = \frac{\pi_i^2 s_{ii}}{\pi_k'} \, v(\Theta_{ii}) \otimes v^{-1}(\Theta_k')$$

$$+ \frac{2\pi_i \pi_j s_{ij}}{\pi_k'} \, v(\Theta_{ij}) \otimes v^{-1}(\Theta_k') + \frac{\pi_j^2 s_{jj}}{\pi_k'} \, v(\Theta_{jj}) \otimes v^{-1}(\Theta_k') - \pi_k', \tag{27}$$

where, for example, Θ_{ij} represents the product distribution between the ith and jth component and s_{ij} the corresponding scaling factor. Each component pair (i, j), for which C_{merge} evaluates below our defined threshold t_{merge}, is replaced by a merged component k, whose parameters are again computed in closed form (App. A.3).

Since both C_{split} and C_{merge} are based on the same per-component χ^2-divergence, either evaluated from sample data or in closed form, using similar threshold values for t_{split} and t_{merge} could lead to cyclical splitting and merging of the same components across iterations. To discourage this behavior, we chose $t_{\text{split}} = 0.5$ over a magnitude larger than $t_{\text{merge}} = 0.025$, accounting for potential variance in the MC estimate of C_{split}.

Advantages of the χ^2 Split and Merge Metric. In the context of importance sampling, using the χ^2-divergence as the criterion for split and merge has the advantage that the fitting process focuses directly on reducing the variance of the resulting MC estimator. Other commonly used criteria, such as the Kullback Leibler (KL) divergence, or the integrated squared difference (ISD), on the other hand, concentrate on fitting the distribution closely in high-valued regions in the observed sample data and neglect regions with low values. This neglect can lead to an MC estimator which undersamples these regions, resulting in high-variance samples (i.e., fireflies).

4.2 Incremental Fitting/Update Scheme

After fitting VMMs to an initial set of samples S, the next task is to derive an iterative update scheme. Sampling from the current, potentially coarse, approximations of L_i will already improve the quality in the next batch of samples S'. These higher-quality samples need to be integrated into the following training iteration, to fit more precise approximations of L_i, which ultimately improve the sampling quality in the final rendering process.

In a guiding framework like ours, the shape of the L_i-distribution approximated by a region can change with every additional training iteration due to the refinement of the spatial subdivision structure. Therefore, straight-forward accumulation of the information gathered from S and S' may be counterproductive, as they are not guaranteed to describe the same L_i-distribution (e.g., due to occlusion).

Müller et al. [2017] avoid this issue by re-fitting all distributions using statistics solely collected form S'. To ensure that each region aggregates enough samples after spatial subdivision, the number of samples is doubled in each training iteration, also doubling the time until updated distributions can be used for guiding. In our wEM-based adaptive VMM fitting approach (Sec. 4.1), this would require storing the exponentially growing amount of sample data collected during each training iteration, which is infeasible in practice.

To make the best use of the laboriously gathered sample data, we instead propose an incremental approach that robustly updates the directional distributions even from small batches of samples. Our training process benefits from immediate updates, converging faster to the true distribution of L_i (Fig. 3).

We model the relationship between the updated distribution and the distribution that resulted from the previous training iteration by applying a Bayesian VMM fitting approach [Bagchi and Guttman 1988; Bangert et al. 2010]. The previous VMM parameters Θ are used as a *prior* to update the novel parameter set Θ' in a *maximum a posteriori* (MAP) estimation. Specifically, we optimize the posterior log likelihood $\mathcal{L}(S')$, which combines the log likelihood of the novel data of N' samples with the weighted log likelihood of the prior distribution $\rho(\Theta' \mid \Theta)$ derived from N previously observed samples:

$$\hat{\Theta}' = \arg\max_{\Theta'} \mathcal{L}(S') = N \log \rho(\Theta' \mid \Theta) + \sum_{n=1}^{N'} w_n' \log \mathcal{V}(\omega_n' \mid \Theta')$$

$$= N \sum_{k=1}^{K} \log \underbrace{\rho(\pi_k', \mu_k', \kappa_k' \mid \pi_k, \mu_k, \kappa_k)}_{\text{per component priors}} + \sum_{n=1}^{N'} w_n' \log \mathcal{V}(\omega_n' \mid \Theta'). \tag{28}$$

This scheme will produce robust updates even for small numbers of novel samples. Indeed, even individual novel samples are robustly incorporated.

When using a distribution from the exponential family, such as the vMF or Gaussian distribution, this MAP formulation corresponds directly to the incremental EM formulation by Neal and Hinton [1998] resulting in the following, rather simple MAP parameter estimation for the sufficient statistics:

$$r_k'' = r_k + r_k', \quad N'' = N + N'. \tag{29}$$

The MAP estimates for the sufficient statistics (r_k'' and N'') are then used to estimate Θ' using the MLEs for the different VMM parameters (π_k, μ_k, and κ_k) as described in Sec. 4.1 (Eq. 15 and 16). After an update step, directional splitting and merging is performed to quickly adapt to entirely new information and to adjust the number of VMM modes. If the overall sample batch size is significantly bigger than the number of samples used by each incremental update, Neal and Hinton [1998] have shown that the incremental EM approach can achieve a similar fitting quality as batch EM but at a significantly lower computational cost.

Incremental Update after Spatial Split. The MAP formulation of the incremental fitting process in Eq. 28 and 29 assumes that all, previous and newly observed, samples originate from the same target distribution. In our application, spatial subdivision can violate this assumption. E.g., the radiance described by samples that contributed to the fitted parameters of the parent region might not be visible in
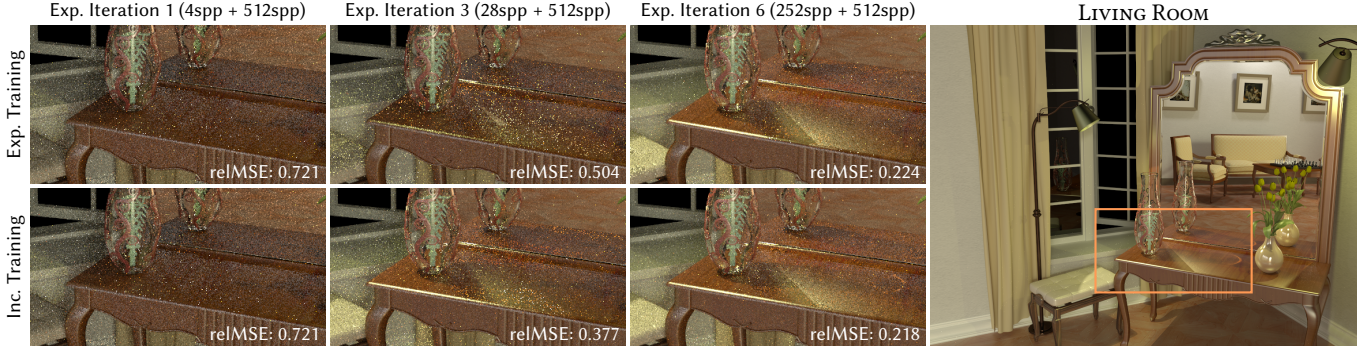
Fig. 3. Comparing the effectiveness of our proposed incremental fitting approach (bottom), using 4spp per training iteration, to traditional, batch-based fitting (top), which doubles its training samples in each successive iteration. The quality of each method is evaluated by rendering 512spp after every other exponential training iteration has finished (e.g., 1st, 3rd, and 6th). While starting with the same initial fit (left), our incremental approach quickly benefits from the more frequent updates of the guiding distributions, leading to a faster learning rate: exp. 3rd vs. inc. 7th (middle) and exp. 6th vs. inc. 63rd iteration (right).

all of its children. Also, the new sample data is likely to contain less variance due to improved sampling and should thus contribute more to the combined estimate. To adjust for these changes, we decay the strength of the sufficient statistics describing the prior observations upon spatial subdivision, when a VMM distribution and its gathered statistics are passed from a parent node to its children:

$$r_k := \alpha r_k, \quad N := \alpha N, \quad \text{where} \quad \alpha = 0.25. \tag{30}$$

Newly observed samples, gain, thereby, more importance, such that new information can be learned quickly. Due to the globally decreasing variance in the sample data and the continued application of splitting and merging, the approach remains robust.

4.3 Parallax-Aware Incident Radiance Representation

A major shortcoming of most recent local path guiding approaches is the fact that they use a static approximation of the incident radiance distribution within each spatial region. This used approximation is a marginalization (i.e., average) of the incident radiance distribution over the area covered by the region (Fig. 4 top). Especially in early training iterations, when these regions are still large, such a representation leads to high-variance artifacts towards the edges of such regions, where the actual incident radiance distribution deviates most from the marginalized one (Fig. 1 left and Fig. 5 left). The primary causes of these deviations are close-by contributions that experience parallax. These artifacts diminish as the sizes of the spatial subdivision regions are getting smaller but may require long training times until a sufficient subdivision is reached. Moreover, the efficiency of the intermediate training iterations suffers from the high-variance sample data gathered early on.

(Ir)radiance caching methods successfully deal with spatially varying illumination by gradient-aided interpolation [Ward and Heckbert 1992; Křivánek et al. 2005a,b; Marco et al. 2018]. We address this issue by introducing a parallax-aware VMM representation (Fig. 4 bottom), which encodes the perceived origin $\mathbf{o}_k$ of the incident radiance covered by each vMF-component indirectly via a reference location $\mathbf{p}$ (e.g., the mean sample position) and a per-component distance d_k between $\mathbf{o}_k$ and $\mathbf{p}$. Using these quantities, we define the parallax-aware vMF-distribution $v_{\mathbf{p}}$, which resolves to a regular vMF-distribution for any sampling location $\mathbf{x}$:

$$v_{\mathbf{p}}(\mathbf{x}, \omega \mid \mathbf{p}, \mu_k, d_k, \kappa_k) = v(\omega \mid \mu_{\mathbf{x},k}, \kappa_k). \tag{31}$$

Where the $\mathbf{x}$-position dependent mean direction $\mu_{\mathbf{x},k}$ is defined as:

$$\mu_{\mathbf{x},k} = \frac{\mathbf{o}_k - \mathbf{x}}{\|\mathbf{o}_k - \mathbf{x}\|} \quad \text{for} \quad \mathbf{o}_k = \mathbf{p} + \mu_k d_k. \tag{32}$$

The values of μ_k and κ_k are determined by the regular fitting procedure. However, first, each sample s_n is reprojected to the reference location $\mathbf{p}$, adapting the direction ω_n and distance d_n accordingly:

$$\omega'_n = \frac{\mathbf{o}_n - \mathbf{p}}{d'_n} \text{ and } d'_n = \|\mathbf{o}_n - \mathbf{p}\|, \text{ where } \mathbf{o}_n = \mathbf{x}_n + \omega_n d_n. \tag{33}$$

Through this reprojection, the sample data represents a precise directional distribution at the reference location $\mathbf{p}$ during the fitting process (Fig. 4 middle). The resulting distribution $v_{\mathbf{p}}$ can then be warped to any sample location $\mathbf{x}$, where it also closely matches the incident radiance up to some remaining error, e.g., due to occlusions.

Similar to previous work in (ir)radiance caching [Ward et al. 1988], we use the harmonic mean to estimate the per-component distance d_k, where we weight each sample's distance by the product of its weight, soft-assignment, and the component's PDF to put more weight on samples close to the center of the vMF lobe:

$$d_k = \left(\frac{\sum_{n=1}^{N} w_n \gamma_k(\omega_n) \, v(\omega_n \mid \Theta_k) d'^{-1}_n}{\sum_{n=1}^{N} w_n \gamma_k(\omega_n) \, v(\omega_n \mid \Theta_k)} \right)^{-1}. \tag{34}$$

The sample distances d_n are defined as the distance to the next diffuse surface interaction encountered along the remaining path (including glossy surfaces with a roughness of at least 0.3). On specular reflections or refractions, distances add up. Additionally, at refractions, we scale the remaining distance by the following correction factor to convert the true distance to the apparent distance:

$$\left| \frac{\cos \theta_{\text{int}}}{\cos \theta_{\text{ext}}} \right| \cdot \frac{\eta_{\text{ext}}}{\eta_{\text{int}}}, \tag{35}$$

where η_{ext} is the optical thickness of the medium in which the remaining path continues in direction ω_{ext} and vice versa.

In summary, our parallax-compensation approach significantly reduces approximation errors that are caused by marginalization
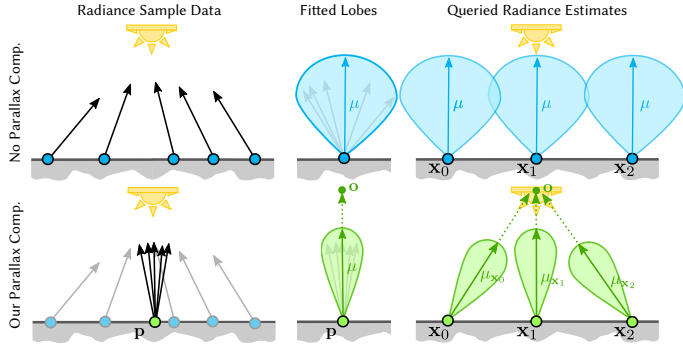
Fig. 4. Comparing different vMF lobe representations: without parallax compensation (top) and with our parallax compensation (bottom), applied to the sampled data before fitting (middle). When queried for a specific position $\mathbf{x}_n$, our parallax compensation re-adjusts the vMF-lobe such that it always sharply points towards the origin $\mathbf{o}$, i.e., the light source (right).
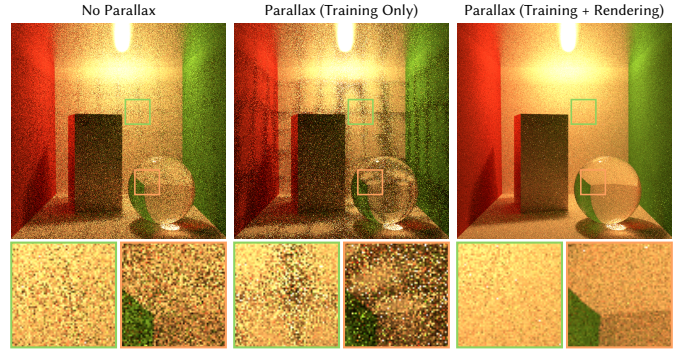

Fig. 5. The commonly used marginalization of the directional distribution (left) increases the sample variance at the borders of the spatial subdivision. When only reprojecting samples to a reference location $\mathbf{p}$ before fitting (Eq. 33), this effect is increased (middle) but can be removed by warping (Eq. 32) the resulting distribution to each sample position $\mathbf{x}$ (right).

and therefore increases training efficiency and image quality, as shown for the GLOSSY CBOX in Fig. 5.

5 IMPLEMENTATION

We implemented our robust fitting and guiding framework as a plugin for the Mitsuba renderer [Jakob 2010]. We make use of the SSE instruction set to vectorize sampling, evaluation and fitting of the VMMs.

Parallax-Aware Merging of Components. Since our merge criterion in Sec. 4.1 is derived for the standard vMF distribution, it does not consider the fact that, in our parallax-aware VMM representation, the directions of the lobes change depending on the actual query position $\mathbf{x}$. Evaluating C_{merge} only at the reference point $\mathbf{p}$ may merge components that are only similar for $\mathbf{x} = \mathbf{p}$ but differ at other positions. To prevent such merges, we do not only evaluate C_{merge} for $\mathbf{x} = \mathbf{p}$ but also for three different random locations inside the spatial region. Two components are only merged if all four C_{merge} evaluations lie below our merge threshold t_{merge}.

Materials, Mirrors and Glass. For product guiding, similar to Herholz et al. [2016], we fit the BSDF using PMMs. However, we only use one lobe for diffuse and one for glossy components, for simplicity. Especially in the presence of anisotropic BSDFs, a more detailed representation should provide more efficient sampling distributions. For materials with discrete components (e.g., mirrors or glass), we first decide whether to sample that component based on the BSDF model's internal selection weight. For the discrete directions, guiding has zero probability and we therefore only sample the BSDF.

BSDF Selection Weight. During training and rendering, we sample new directions using a one-sample MIS combination (Eq. 3) of our guiding distribution p_g and BSDF importance sampling p_{f_s} with a selection weight for sampling the BSDF of $\alpha = 0.5$. Only when guiding according to the complete reflected radiance integrand (Eq. 5), we found it safe to lower the selection weight to $\alpha = 0.25$, which leads to an additional performance increase. Using the same value when just guiding based on the incident radiance distribution, or

the cosine product, leads to a substantial variance increase in most scenes during training and rendering.

Integrating Next-Event Estimation in Path Guiding. In modern rendering systems, it is common practice to increase the efficiency of estimating the direct contribution L_{dir} by combining a path-based estimator $\langle L_{\text{dir}}^{\text{path}} \rangle$ with a next-event estimator $\langle L_{\text{dir}}^{\text{NEE}} \rangle$ (NEE), using multiple importance sampling (MIS) [Veach and Guibas 1995b]:

$$\langle L_{\text{dir}}^{\text{MIS}} \rangle = w_{\text{path}} \cdot \langle L_{\text{dir}}^{\text{path}} \rangle + w_{\text{NEE}} \cdot \langle L_{\text{dir}}^{\text{NEE}} \rangle. \qquad (36)$$

The MIS weights w_{path} and w_{NEE} try to combine the outcome of the individual estimators in such a way, that the resulting variance of $\langle L_{\text{dir}}^{\text{MIS}} \rangle$ is lower than the variance of the individual estimators. Previous work did usually neglect the MIS combination and either fully include or exclude the contribution from $\langle L_{\text{dir}}^{\text{MIS}} \rangle$ during training, when next-event estimation is enabled. While the first completely ignores the potential advantages of NEE, the latter assumes that the NEE is optimal and has *zero variance*. Ideally, a guiding distribution should only sample those direct light contributions which cannot be handled well by NEE. We encourage the fitting to consider the MIS combination (Eq. 36) by multiplying the MIS weight w_{path} of the guided estimator to gathered direct light estimates in the sample data. This weight will be high in cases where the probability of sampling this contribution using NEE is low, e.g., due to many other emitters that could be chosen instead. On the other hand, this weight will be low for contributions that can be reliably sampled by NEE. Initially, almost all light sources are handled better by NEE than by the untrained guided estimator (i.e., only using BSDF importance sampling), leading to low values for w_{path}. So, applying the weight immediately would remove most direct light contributions from the sample data, leaving very little chance for later recovery. This is why we only apply the weight after a small burn-in phase, e.g., the first 12spp of training. At this point, it is worth noting that, in our framework, sample data is only generated for the primary light transport paths and not for directions sampled by NEE.

EM Initialization and MAP Priors. We Initialize our VMMs with 16 components distributed by the spherical Fibonacci point set [Hannay

and Nye 2004] to uniformly cover the sphere. Similar to Vorba et al. [2014] and Herholz et al. [2019], we avoid overfitting by replacing the *weighted maximum likelihood estimators* (wMLEs) for π_k and κ_k, which are used in the M-step of wEM fitting, by their *weighted maximum a posteriori* (wMAP) counterparts:

$$\pi_k' = \frac{\pi_k + \pi_{\text{prior}}}{1 + K\pi_{\text{prior}}}, \quad \text{and} \quad \bar{r}_k' = \frac{\bar{r}_k \pi_k' N + \bar{r}_{\text{prior}} N_{\text{prior}}}{\pi_k' N + N_{\text{prior}}}. \tag{37}$$

In particular, we use a weight prior value of $\pi_{\text{prior}} = 0.01$ to ensure that each component represents at least $1/K$ percent of the incident radiance. The concentration prior $\bar{r}_{\text{prior}} = 0$ applies the prior assumption that the data represented by a component is distributed uniformly. This assumption is only applied to $N_{\text{prior}} = 0.2$ virtual samples. These priors are also applied during our incremental fitting approach described in Sec. 4.2, where the total number of samples N is not subject to the decay introduced in Eq. 30 to allow this prior to vanish with increasing number of samples.

Spatial Subdivision Structure. We use a kd-tree as a spatial data structure for guiding. Each node contains a VMM representing the local distribution of incident radiance. We start with a single region covering the whole scene space and subdivide regions that gathered at least 32k non-zero samples. By counting only the non-zero valued samples, we ensure that sufficiently many valid samples are available to fit the VMMs. We collect statistics about the sample positions within each region, including mean, variance and their bounding box. The subdivision process splits the region at the mean position in the dimension of highest variance. This results in an approximately even division of samples.

Spatial Sample Splatting. In some cases, the important incident radiance contributions are sampled so scarcely, that they are only ever sampled within a small subset of the regions where they could be observed. In those regions, the directional distribution model will learn about this contribution and improve its sampling strategy accordingly. However, its neighboring regions may lack this information and still need to find out about it by chance. To facilitate information sharing across regions, we spatially splat samples, i.e., randomly reposition them within a small region around their true location. To limit the blurring of directional information across the spatial domain we splat within an ellipsoid whose form factor and size is derived from the bounding box of the samples previously observed in the local region.

6 EVALUATION

In this section, we evaluate the proposed techniques individually on a number of challenging scenes. We evaluate our approach regarding three aspects: First, we evaluate the fitting procedure (Sec. 6.1), then we compare various guiding approaches (Sec. 6.2) and finally, we compare our approach against the work of Müller et al. [2017] (Sec. 6.3). Quality is measured by the relative mean squared error (relMSE). To exclude outliers, we discard the 0.1% highest error values before computing the average. A small epsilon of $\varepsilon = 0.001$ is added to the denominator to avoid divisions by zero.

Except for the equal-time comparisons, or unless explicitly stated otherwise, we used 1020spp total for training and 1024spp for the final rendering of our images. Our incremental training approach

(Sec. 4.2) uses the sample data gathered from using 4 samples per pixel (SPP) in each training iteration. All our evaluations are performed on a machine with 512GB of RAM and two Intel(R) Xeon(R) Gold 5115 CPUs @ 2.40GHz, using all 40 logical cores.

6.1 Fitting Procedure

In the following, we analyze the performance of the proposed fitting method (Sec. 4.1 and Sec. 4.2), the appropriateness of the splitting and merging thresholds, as well as the distribution and progression of the number of components.

Variance-Based Adaptive Fitting. As shown in Fig. 2, the traditional EM-based algorithm results in unpredictable fit quality, producing high-variance artifacts in the rendered image. Our adaptive fitting method refines the VMMs by splitting and merging, generates more reliable guiding distributions and thus reduces the error significantly, e.g., in Fig. 8 (+adaptive) and Fig. 9 (ours).
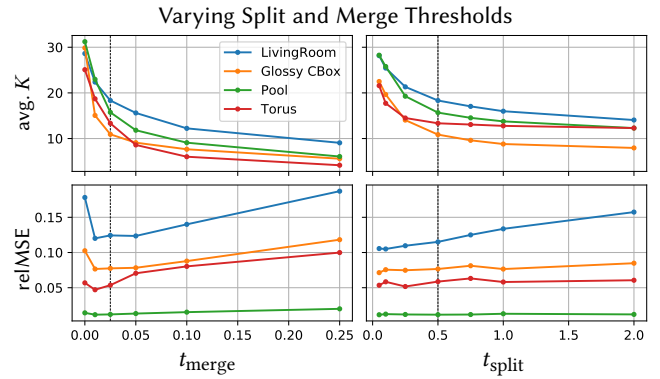

Fig. 6. Average number of components K and relMSE for varying thresholds of the split and merge criteria, evaluated at equal samples. Raising the thresholds results in mixtures with less components and thus higher error. Changing the thresholds in the variance-based split and merge criteria correlates with a change in relMSE.

Split and Merge Thresholds. The thresholds set for the split and merge criteria determine the balance between approximation and estimation error, i.e., the balance between a fast-to-evaluate and an accurate model. Fig. 6 shows how the number of components and the image quality varies based on these thresholds. Setting the merge criterion's threshold to zero corresponds to a special case, which only allows for merging of exactly overlapping components and, therefore, often results in higher error when mixtures reach the implementation-defined maximum number of components (32), and splitting is no longer possible. We chose low thresholds for high quality, which are barely above the range where further increases in quality would start to require exponentially more components.

Mixture Component Usage. For different scenes, the average numbers of mixture components needs to adapt to the complexity of the local subsets of the incident radiance field. The data shown in Fig. 7 left shows that, in the beginning, a higher number of components is used, as the initially large regions contain more diverse information about the radiance. With increased spatial subdivision, the average
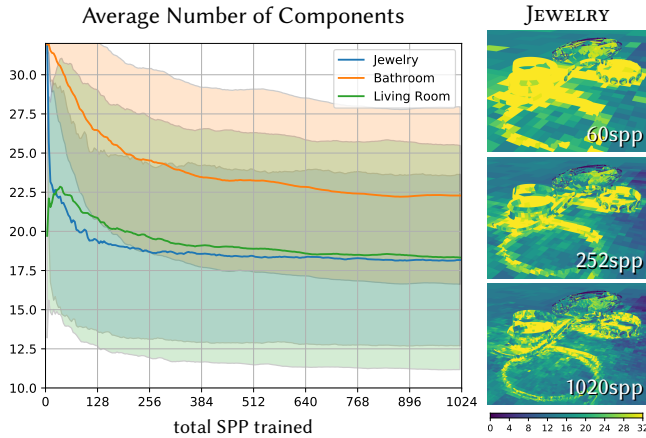
Fig. 7. Our splitting and merging criteria result in a stable, scene-dependent number of components that our VMMs converge towards during training. The JEWELRY scene shows that more components are required e.g. in areas with varying surface normals, where a wider range of incident radiance is visible. Areas with high variance utilize mixtures with the maximum amount of components to reduce their variance as much as possible, while areas with simpler illumination use smaller, fast-to-evaluate mixtures.

Table 1. Comparing the exponential and incremental training approaches. Exploring complex light transport earlier, combined with better retention of learned information and a low update cost, allows the incremental approach to outperform the exponential approach in most scenes. Less time and vastly less memory is required when compared to the last exponential training iteration, which had to gather the samples from 512spp rather than 4spp.

Scene	Exponential			Incremental		
	time	relMSE	mem. (MB)	time	relMSE	mem. (MB)
BATHROOM	59.6m	0.308	28 823	**46.7m**	**0.143**	**330**
CLOCKS	5.1m	0.373	1 759	**5.0m**	**0.197**	**19**
C.-KITCHEN (DAY)	12.5m	0.056	17 014	**10.9m**	**0.043**	**137**
C.-KITCHEN (NIGHT)	11.5m	0.096	16 093	**9.3m**	**0.087**	**131**
GLOSSY CBOX	6.8m	0.112	10 934	**6.2m**	**0.078**	**100**
JEWELRY	8.4m	0.371	4 222	**6.1m**	**0.353**	**34**
KITCHEN	24.6m	0.140	16 062	**12.1m**	**0.123**	**127**
KITCHENETTE	28.6m	0.120	42 895	**25.5m**	**0.106**	**343**
LIVING ROOM	15.7m	0.213	15 389	**12.1m**	**0.120**	**127**
POOL	8.3m	0.016	16 129	**7.1m**	**0.012**	**126**
TORUS	6.3m	**0.054**	15 472	**5.1m**	0.055	**121**

Table 2. By computing the cosine product, vMF-components on the lower hemisphere are mostly eliminated and the ratio of invalid samples (for which the BSDF evaluates to zero) can be greatly reduced, resulting in higher-quality light-transport path samples and lower relMSE in equal time (10m). In the absence of Russian roulette, the improved sampling quality increases the average path length (avg. p. l.) and thereby reduces the number of samples per pixel (SPP) that can be evaluated.

Scene	% invalid		relMSE		SPP		avg. p. l.	
	L_i	$\cos\theta_i$	L_i	$\cos\theta_i$	L_i	$\cos\theta_i$	L_i	$\cos\theta_i$
average (all scenes)	32.21	19.74	0.114	0.106	3 255	2 577	5.90	6.93
BATHROOM	36.12	13.41	0.470	0.395	440	316	10.54	12.89
C.-KITCHEN (NIGHT)	40.22	20.19	0.053	0.051	2 888	2 032	7.49	9.09
GLOSSY CBOX	42.86	33.51	0.038	0.035	5 780	4 296	7.24	8.27
LIVING ROOM	24.77	11.24	0.094	0.092	2 324	1 732	3.85	4.47

number of components decreases to a point, where it remains stable even after prolonged training. The directional distributions typically become simpler with decreasing region size, which can then be represented using fewer components, as seen in the JEWELRY scene (Fig. 7 right). Only in areas with complex incident radiance, a high number of components is required.

Incremental Fitting. The incremental fitting approach presented in Sec. 4.2 reduces the turnaround time after which the knowledge contained in the gathered sampled data can be used to make more informed sampling decisions. As an example, Fig. 3 shows how the incremental approach can speed up the learning of complex light transport paths compared to an exponential approach. In Table 1, we compare the incremental and exponential approaches in a wide range of scenes. It shows that the incremental approach is generally faster, as distributions do not need to be fitted from scratch, but can instead be updated quickly. The amount of memory required to store the sample data is orders of magnitude lower, as the exponential approach requires twice the amount of memory each iteration. Superior results can be achieved, processing the same number of samples, since features, once found, can be explored quickly and then be shared with neighboring regions each iteration via spatial splatting (Sec. 5). This especially benefits scenes with small, hard to reach light sources such as BATHROOM, CLOCKS, and LIVING ROOM.

6.2 Guiding Methods

In Fig. 8, we incrementally improve upon the results of a guided path tracer with parallax-compensation, adaptive fitting, the cosine- and BSDF-products, as well as next-event estimation (NEE). We have already shown in Fig. 5 that the parallax-compensation can greatly improve image quality in the presence of close-by contributors. We

also examined the adaptive fitting procedure in the previous section. In the following, we examine the remaining product sampling approaches and the NEE-weights.

Hemispherical Guiding with Spherical Distributions. Using spherical distributions for guiding in the presence of varying surface normals, such as on corners or edges, can lead to high amounts of invalid samples. By proposing directions on the lower hemisphere, the random walk is terminated immediately. The probability of such samples can be significantly reduced by computing the product with the surface-normal oriented cosine-lobe just before sampling. The increased sampling quality allows long light transport paths to be established more reliably, increasing average path length and image quality, as shown in Table 2.

Not all long light transport paths contribute to the image by an amount that warrants the required computation, though. To improve efficiency, stochastic path termination, e.g., [Vorba and Křivánek 2016], could be used to terminate less important paths early in a controlled fashion.
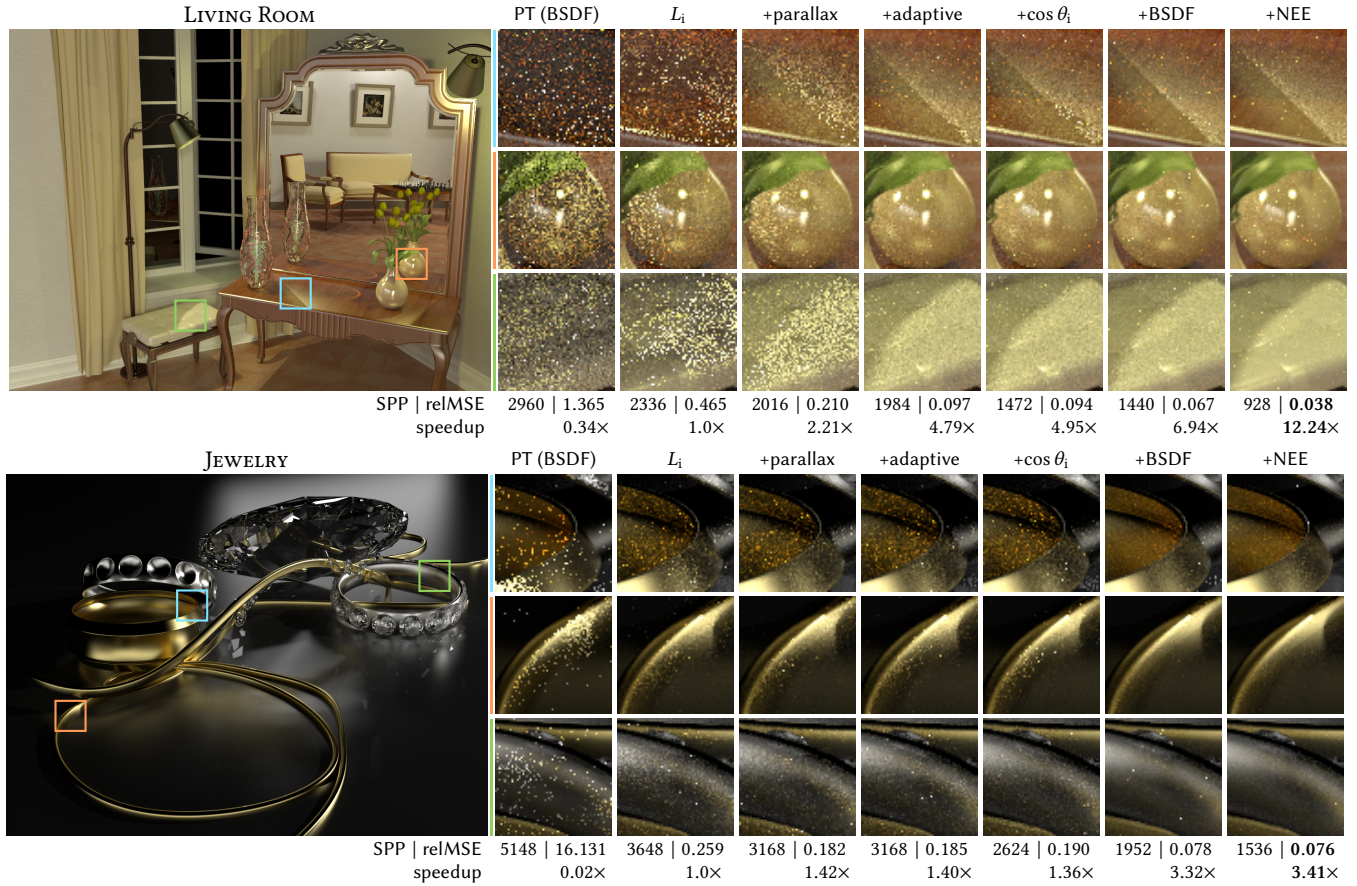
Fig. 8. Equal time comparison (10 min.) successively including more components of our framework. Each method was trained for 3 minutes and rendered for another 7 minutes. The unguided path tracer used all 10 minutes for rendering.

Product Importance Sampling & NEE. With the addition of product importance sampling [Herholz et al. 2016], an even better sampling distribution can be achieved on glossy surfaces which only reflect within narrow lobes. With our NEE-weights, next event estimation can be safely incorporated into the guided path tracer to further improve efficiency. Fig. 8 shows these improvements for a small set of scenes.

6.3 State-of-the-Art Comparison

We compare our approach to the exponential forward-learning approach by Müller et al. [2017]. In their approach, we enabled the spatio-directional splatting from their improved version presented in [Vorba et al. 2019], along with their proposed smaller spatial tree threshold (4k instead of 12k), which consistently surpassed the results of their traditional approach in our testing. We did not apply any of their further presented improvements, which are mostly orthogonal to the guiding approach. For a fair comparison, we did also not enable the BSDF-product or NEE in our method. We also first executed the method of Müller et al. [2017] with a 10 minute budget and used the same training and rendering times as the ones automatically chosen by their method. Table 3 shows the detailed results, most importantly the relative mean squared error (relMSE).

In most scenes, our variance-aware incremental learning scheme and the parallax-aware distributions explore the path space more efficiently, leading to a significant error reduction. Müller et al. [2017] can process more samples in the same time and achieve slightly better results in the TORUS and KITCHEN scenes, which are mainly illuminated by the sun. Partially, this is due to the shorter average path length, which, to some extent, is the result of unintended path termination when sampling the lower hemisphere in the absence of a cosine-product distribution (see Sec. 6.2).

In Fig. 9, a visual comparison is shown for two scenes. There, we additionally compare our BSDF-product which provides additional significant improvements in quality on glossy materials.

7 DISCUSSION AND LIMITATIONS

Inadequate Spatial Subdivision. The used spatial subdivision strategy is rather naïve (i.e., only based on the number of samples). It completely neglects characteristic light transport effects such as local changes in irradiance (e.g., caustics, or occlusions) or scene geometry and how well these can be represented by a parallax-aware VMM-distribution. As a consequence, regions with spatial variation beyond parallax may not be resolved fine enough, while others, that could be represented by a single distribution may be

Table 3. Equal time comparison. Samples per pixel (SPP) for training (T) and rendering (R), average path length (avg. p. l.), average number of VMM components (avg. K), number of regions, memory consumption (mem.) in MB by the guiding (G) and sample (S) data, and relative mean squared error (relMSE) of the final image after 10 minutes total. The method of Müller et al. [2017] automatically chooses its training time. For a fair comparison, we used the same training and rendering times in our method. Our adaptive training method, parallax-compensation and cosine-product significantly reduce the error in most scenes.

| | time | | Our Method | | | | | | | Müller et al. [2017] | | | | |
| | | | SPP | | avg. | avg. | #reg- | mem. | | relMSE | SPP | | avg. | #reg- | mem. | relMSE |
Scene	T	R	T	R	p. l.	K	ions	G	S	(speedup)	T	R	p. l.	ions	G	(baseline)
Bathroom	244s	356s	140	316	12.90	26.2	6 547	14.7	337.2	**0.395** (14.46×)	252	348	11.93	46 830	219.7	5.713 (1.0×)
Clocks	35s	565s	244	4 460	4.09	17.7	324	0.7	14.8	**0.126** (73.98×)	508	6 484	4.00	4 537	20.7	9.321 (1.0×)
C.-Kitchen (Day)	75s	525s	208	2 468	9.06	20.9	3 830	8.6	136.1	**0.024** (1.25×)	508	4 360	7.15	13 482	58.3	0.030 (1.0×)
C.-Kitchen (Night)	168s	432s	528	2 032	9.09	18.8	10 080	22.7	140.9	**0.051** (1.69×)	1 020	2 908	8.35	20 021	87.4	0.086 (1.0×)
Glossy CBox	46s	554s	212	4 296	8.27	12.3	2 888	6.4	106.1	**0.035** (25.89×)	508	6 332	7.30	10 340	47.8	0.906 (1.0×)
Jewelry	66s	534s	272	3 496	3.52	19.1	1 475	3.3	38.3	**0.168** (1.98×)	508	4 732	3.33	9 456	46.5	0.332 (1.0×)
Kitchen	102s	498s	228	1 908	5.20	23.0	4 292	9.7	130.2	0.088 (0.94×)	508	2 980	4.65	15 260	71.3	**0.083** (1.0×)
Kitchenette	218s	382s	248	708	10.48	20.7	12 996	29.2	367.0	**0.155** (1.06×)	508	896	8.95	33 596	139.5	0.164 (1.0×)
Living Room	108s	492s	248	1 732	4.47	19.8	4 644	10.4	132.2	**0.092** (2.34×)	508	2 608	4.02	17 215	79.0	0.215 (1.0×)
Pool	156s	444s	604	2 684	4.91	16.8	12 405	27.8	139.1	**0.007** (2.14×)	1 020	3 348	4.89	12 650	55.5	0.015 (1.0×)
Torus	116s	484s	620	4 244	3.83	13.5	12 516	28.1	134.3	0.024 (0.96×)	1 020	5 280	3.76	11 009	48.4	**0.023** (1.0×)
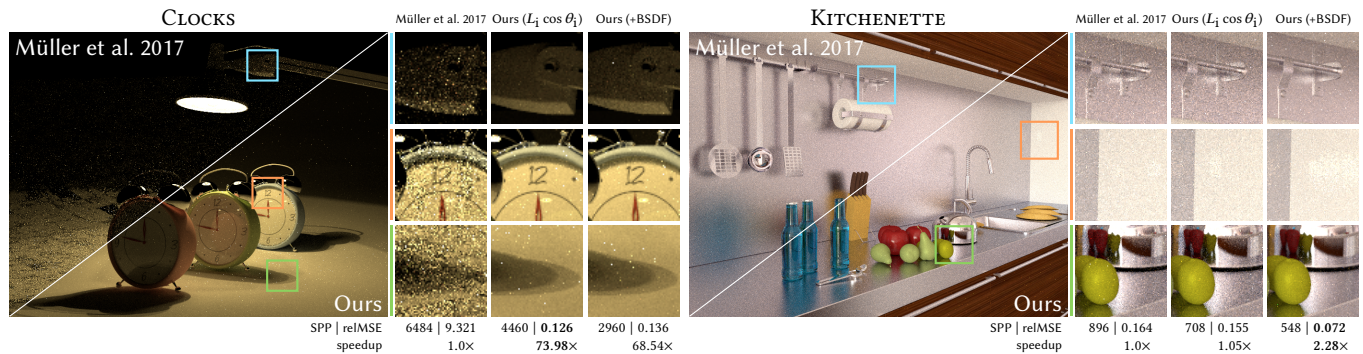
Fig. 9. Visual comparison of the results for the Clocks and Kitchenette scenes listed in Table 3 with additional BSDF-product results.

excessively subdivided. Improving the spatial subdivision strategy in the context of local path guiding is an important direction for future work, which can probably inherit concepts from previous work done in the area of (ir)radiance caching [Křivánek et al. 2008; Schwarzhaupt et al. 2012].

Generalization of Variance-Based Adaptive Fitting to other PMMs. Conceptually, our variance-based adaptive fitting approach (Sec. 4.1) is not bound to the vMF distribution. The used PMM and the sample data need to both support efficient evaluation of the split and merge criteria (i.e., χ^2-divergence). The presented evaluation of C_{split} aims at fitting the VMM proportional to a function, assuming that the sample data are random evaluations of this function with a known PDF. In traditional EM applications, or when learning the L_i-distribution from photons (e.g., [Vorba et al. 2014]), the PDF is usually unknown. In these cases, the PDF needs to be approximated using a rather costly kernel density estimation [Ueda et al. 2000a,b].

Incorporation of Local Path Guiding Extensions. Besides the fitting of local L_i-distributions, there exists a set of orthogonal techniques such as *adjoint-driven Russian roulette and splitting* [Vorba and Křivánek 2016] or *inverse-variance merging* [Vorba et al. 2019], that improve the performance of local path guiding. It should, therefore, be straightforward to integrate them into our presented framework.

Especially the optimization of the BSDF selection weight [Vorba et al. 2019] could lead to a significant improvement, since our framework supports the sampling of the full product distribution.

Relationship to other EM-Algorithms. Using other variants of the EM-algorithm, e.g., *on-line stepwise EM* [Cappé and Moulines 2009; Vorba et al. 2014] or *accelerated EM* [Verbeek et al. 2006; Jakob et al. 2011], the storage of individual samples can be avoided entirely. Instead, in the case of the *on-line stepwise EM*, mixtures are updated immediately with each new sample, where each successive sample's influence is reduced. Adapting to changes due to spatial subdivision or the improved variance in subsequent training iterations would require delicate tuning of the update rate to preserve robustness with this approach. The *accelerated EM* fits mixtures by proxy of tabulated statistics of samples, over which it then iterates instead. While tractable early on, eventually, per component, the tabulated statistics will exceed the number of samples and become inefficient. Moreover, the resolution of the tabulated statistics limit the resolution of local features beyond the approximation capabilities of the mixture.

8 CONCLUSION

The proposed robust fitting framework adaptively fits von Mises-Fisher mixture models to locally sampled radiance distributions

for path guiding, escaping local maxima and selecting a suitable number of components by splitting and merging. Novel criteria for splitting and merging of components ensure to minimize the resulting radiance estimator's variance. By exploiting closed-form expressions for the product and the multiplicative inverse of vMF-distributions, the criteria are evaluated very efficiently. In addition, instead of fitting on a large number of samples, an incremental update scheme operating on small sample batches is introduced, that achieves the same fitting quality but at the same time features faster learning due to an increased update frequency while minimizing memory cost. Combined with further improvements by a novel parallax-aware extension to the vMF-distribution, our approach significantly boosts quality in forward path guiding.

ACKNOWLEDGMENTS

REFERENCES

Hirotugu Akaike. 1974. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*. Springer, 215–222.

James Arvo et al. 1986. Backward ray tracing. In *Developments in Ray Tracing, Computer Graphics, Proc. of ACM SIGGRAPH 86 Course Notes*. 259–263.

Parthasarathy Bagchi and Irwin Guttman. 1988. Theoretical considerations of the multivariate von Mises-Fisher distribution. *Journal of Applied Statistics* 15, 2 (1988), 149–169.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research* 6, Sep (2005), 1345–1382.

Mark Bangert, Philipp Hennig, and Uwe Oelfke. 2010. Using an infinite von Mises-Fisher mixture model to cluster treatment beam directions in external radiation therapy. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*. IEEE, 746–751.

Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. 2012. A significance cache for accelerating global illumination. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1837–1851.

Christopher M. Bishop. 2006. Mixture Models and EM. In *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, New York.

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

Norbert Bus and Tamy Boubekeur. 2017. Double Hierarchies for Directional Importance Sampling in Monte Carlo Rendering. *Journal of Computer Graphics Techniques (JCGT)* 6, 3 (28 August 2017), 25–37.

Olivier Cappé and Eric Moulines. 2009. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71, 3 (2009), 593–613.

David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1186–1195.

Ken Dahm and Alexander Keller. 2017. Learning light transport the reinforced way. In *ACM SIGGRAPH 2017 Talks*. ACM, 73.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

Philip Dutré, Eric P. Lafortune, and Yves D. Willems. 1993. Monte Carlo light tracing with direct computation of pixel intensities. In *3rd International Conference on Computational Graphics and Visualisation Techniques*. Alvor, Portugal, 128–137.

Luca Fascione, Johannes Hanika, Marcos Fajardo, Per Christensen, Brent Burley, and Brian Green. 2017. Path Tracing in Production - Part 1: Production Renderers. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California). Article 13, 39 pages.

Luca Fascione, Johannes Hanika, Daniel Heckenberg, Christopher Kulla, Marc Droske, and Jorge Schwarzhaupt. 2019. Path tracing in production: part 1: modern path tracing. In *ACM SIGGRAPH 2019 Courses*. ACM, 19.

Luca Fascione, Johannes Hanika, Rob Pieké, Ryusuke Villemin, Christophe Hery, Manuel Gamito, Luke Emrose, and André Mazzone. 2018. Path tracing in production. In *ACM SIGGRAPH 2018 Courses*. ACM, 15.

Nicholas I Fisher, Toby Lewis, and Brian JJ Embleton. 1987. *Statistical analysis of spherical data*. Cambridge university press.

Iliyan Georgiev, Jaroslav Krivánek, Tomas Davidovic, and Philipp Slusallek. 2012. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.* 31, 6 (2012), 192–1.

Pascal Grittmann, Arsène Pérard-Gayot, Philipp Slusallek, and Jaroslav Křivánek. 2018. Efficient Caustic Rendering with Lightweight Photon Mapping. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 133–142.

Jerry Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary Sample Space Path Guiding. In *Eurographics Symposium on Rendering - EI & I*, Wenzel Jakob and Toshiya Hachisuka (Eds.). Eurographics, The Eurographics Association, 73–82.

Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic progressive photon mapping. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 141.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. 2012. A Path Space Extension for Robust Light Transport Simulation. *ACM Trans. Graph.* 31, 6, Article 191 (Nov. 2012), 10 pages.

JH Hannay and JF Nye. 2004. Fibonacci numerical integration on a sphere. *Journal of Physics A: Mathematical and General* 37, 48 (2004), 11591.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product importance sampling for light transport path guiding. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 67–77.

Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Křivánek. 2019. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Trans. Graph.* 38, 3, Article 25 (June 2019), 19 pages.

Tim Hesterberg. 1995. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics* 37, 2 (1995), 185–194.

Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th spring conference on Computer graphics*. ACM, 107–114.

David S. Immel, Michael F. Cohen, and Donald P. Greenberg. 1986. A Radiosity Method for Non-Diffuse Environments. *Computer Graphics (Proceedings of SIGGRAPH)* 20, 4 (Aug. 1986), 133–142.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Wenzel Jakob. 2012. Numerically stable sampling of the von Mises-Fisher distribution on Sˆ2 (and other tricks). *Interactive Geometry Lab, ETH Zürich, Tech. Rep* (2012).

Wenzel Jakob, Christian Regg, and Wojciech Jarosz. 2011. Progressive Expectation-Maximization for Hierarchical Volumetric Photon Mapping. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 3, 4 (July 2011), 1287–1297.

Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Rendering Techniques' 95*. Springer, 326–335.

Henrik Wann Jensen. 1996. Global Illumination Using Photon Maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96* (Porto, Portugal). Springer-Verlag, Berlin, Heidelberg, 21–30.

Henrik Wann Jensen. 2001. *Realistic image synthesis using photon mapping*. AK Peters/CRC Press.

G Jona-Lasinio, M Piccioni, and A Ramponi. 1999. Selection of importance weights for monte carlo estimation of normalizing constants. *Communications in Statistics - Simulation and Computation* 28, 2 (1999), 441–462.

James T Kajiya. 1986. The rendering equation. In *ACM Siggraph Computer Graphics*, Vol. 20. ACM, 143–150.

Anton S. Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. 2014. The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation. *ACM Trans. Graph.* 33, 4, Article 102 (July 2014), 13 pages.

Jaroslav Křivánek, Kadi Bouatouch, Sumanta Pattanaik, and Jiří Žára. 2008. Making Radiance and Irradiance Caching Practical: Adaptive Caching and Neighbor Clamping. In *ACM SIGGRAPH 2008 Classes* (Los Angeles, California) (*SIGGRAPH '08*). Association for Computing Machinery, New York, NY, USA, Article 77, 12 pages.

Jaroslav Křivánek, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. 2005a. Improved Radiance Gradient Computation. In *Proceedings of the 21st Spring Conference on Computer Graphics* (Budmerice, Slovakia) (*SCCG '05*). Association for Computing Machinery, New York, NY, USA, 155–159.

Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. 2005b. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 550–561.

Eric P. Lafortune and Yves D. Willems. 1993. Bi-Directional Path Tracing. In *Proceedings of the International Conference on Computational Graphics and Visualization*

Techniques (Compugraphics), Vol. 93. Alvor, Portugal, 145–153.

Eric P Lafortune and Yves D Willems. 1995. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques' 95*. Springer, 11–20.

Julio Marco, Adrian Jarabo, Wojciech Jarosz, and Diego Gutierrez. 2018. Second-Order Occlusion-Aware Volumetric Radiance Caching. *ACM Trans. Graph.* 37, 2, Article 20 (July 2018), 14 pages.

Geoffrey McLachlan and Thriyambakam Krishnan. 2007. *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. 2019. Finite mixture models. *Annual review of statistics and its application* 6 (2019), 355–378.

Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 91–100.

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages.

Richard F Murray and Yaniv Morgenstern. 2010. Cue combination on the circle and the sphere. *Journal of vision* 10, 11 (2010), 15–15.

Radford M Neal and Geoffrey E Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*. Springer, 355–368.

Jerzy Neyman and Egon Sharpe Pearson. 1933. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231, 694-706 (1933), 289–337.

Vincent Pegoraro, Ingo Wald, and Steven G. Parker. 2008. Sequential Monte Carlo Adaptation in Low-Anisotropy Participating Media. *Computer Graphics Forum* 27, 4 (2008), 1097–1104.

Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbacher. 2018. Selective Guided Sampling with Complete Light Transport Paths. *ACM Trans. Graph.* 37, 6, Article 223 (Dec. 2018), 14 pages.

Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics* 6, 2 (1978), 461–464.

Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. 2012. Practical Hessian-Based Error Control for Irradiance Caching. *ACM Trans. Graph.* 31, 6, Article 193 (Nov. 2012), 10 pages.

Martin Šik and Jaroslav Křivánek. 2019. Implementing One-Click Caustics in Corona Renderer. In *Eurographics Symposium on Rendering - DL-only and Industry Track*, Tamy Boubekeur and Pradeep Sen (Eds.). The Eurographics Association, 61–67.

Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E Hinton. 2000a. SMEM algorithm for mixture models. *Neural computation* 12, 9 (2000), 2109–2128.

Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E Hinton. 2000b. Split and merge EM algorithm for improving Gaussian mixture density estimates. *Journal of VLSI signal processing systems for signal, image and video technology* 26, 1-2 (2000), 133–140.

Eric Veach and Leonidas Guibas. 1995a. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*. Springer, 145–167.

Eric Veach and Leonidas J. Guibas. 1995b. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Annual Conference Series (Proceedings of SIGGRAPH)*, Vol. 29. ACM Press, 419–428.

Jakob J. Verbeek, Jan R. Nunnink, and Nikos Vlassis. 2006. Accelerated EM-Based Clustering of Large Data Sets. *Data Min. Knowl. Discov.* 13, 3 (Nov. 2006), 291–307.

Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. 2019. Path guiding in production. In *ACM SIGGRAPH 2019 Courses*. ACM, 18.

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 (TOG)* 33, 4 (2014), 101.

Jiří Vorba and Jaroslav Křivánek. 2016. Adjoint-driven Russian roulette and splitting in light transport simulation. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 42.

Hai xian Wang, Bin Luo, Quan bing Zhang, and Sui Wei. 2004. Estimation for the number of components in a mixture model using stepwise split-and-merge EM algorithm. *Pattern Recognition Letters* 25, 16 (2004), 1799–1809.

G.J. Ward and P.S. Heckbert. 1992. *Irradiance gradients*. Technical Report.

Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '88)*. Association for Computing Machinery, New York, NY, USA, 85–92.

Quan Zheng and Matthias Zwicker. 2019. Learning to Importance Sample in Primary Sample Space. *Computer Graphics Forum* (2019).

A OPERATIONS OF VMF-DISTRIBUTIONS

In the following, a few useful operations on vMF-distributions, which our method requires, are shortly summarized.

A.1 Product

As presented by Herholz et al. [2019], based on Murray and Morgenstern [2010], the product of two vMF-lobes (i, j) is again a vMF lobe ij, scaled by the factor s_{ij}:

$$s_{ij} \, v(\omega|\kappa_{ij}, \mu_{ij}) = v(\omega|\kappa_i, \mu_i) \, v(\omega|\kappa_j, \mu_j), \tag{38}$$

$$\kappa_{ij} = \|\kappa_i\mu_i + \kappa_j\mu_j\|, \quad \mu_{ij} = \frac{\kappa_i\mu_i + \kappa_j\mu_j}{\kappa_{ij}}, \tag{39}$$

where the scaling factor s_{ij} describes the product's integral:

$$
\begin{aligned}
s_{ij} &= \int_{S^2} v(\omega|\kappa_i, \mu_i) \, v(\omega|\kappa_j, \mu_j) \, \mathrm{d}\omega \\
&= \frac{\kappa_i}{4\pi \sinh \kappa_i} \frac{\kappa_j}{4\pi \sinh \kappa_j} \frac{4\pi \sinh \kappa_{ij}}{\kappa_{ij}}
\end{aligned}
\tag{40}
$$

$$= \frac{\kappa_i\kappa_j(1 - \exp(-2\kappa_{ij})) \exp(\kappa_i(\mu_i^T\mu_{ij} - 1) + \kappa_j(\mu_j^T\mu_{ij} - 1))}{2\pi\kappa_{ij}(1 - \exp(-2\kappa_i))(1 - \exp(-2\kappa_j))}.$$

A.2 Multiplicative Inverse

The multiplicative inverse of the vMF distribution can be derived directly from its definition:

$$
\begin{aligned}
\frac{1}{v(\omega|\kappa, \mu)} &= \frac{4\pi \sinh \kappa}{\kappa} \exp(-\kappa\omega^t\mu) = \left(\frac{4\pi \sinh \kappa}{\kappa}\right)^2 v(\omega|\kappa, -\mu) \\
&= \frac{4\pi^2(1 - \exp(-2\kappa))^2}{\kappa^2 \exp(-2\kappa)} v(\omega|\kappa, -\mu).
\end{aligned}
\tag{41}
$$

A.3 Moment-Preserving Merge

When merging the pair of components (i, j) into a new component k, their mixture weights π and sufficient statistics r simply add up:

$$\pi_k = \pi_i + \pi_j, \quad \text{and} \quad r_k = r_i + r_j, \tag{42}$$

where r can be recovered from the vMF's parameters:

$$r = \pi\bar{r}\mu, \quad \text{where} \quad \bar{r} = \coth \kappa - 1/\kappa. \tag{43}$$

The estimation of κ_k and μ_k follow using the regular MLE steps.

A.4 Moment-Preserving Split

To split a component k into two components (i, j), a small offset from the merged mean direction μ_k is required. Our implementation uses PCA based on incrementally computed covariance of the gathered samples in the xy-plane of each component's local coordinate frame. We place the new components at $\pm v_{\lambda_1}\sqrt{\lambda_1}/2$ and then project back into world space, such that they point towards the individual modes of bimodal data. We further initialize both components with the same mixture weight $\pi_i = \pi_j = \pi_k/2$ and the same mean cosine $\bar{r}_i = \bar{r}_j$. To preserve the total sufficient statistics r_k, we enforce that

$$r_k = r_i + r_j \overset{(43)}{\Longleftrightarrow} \bar{r}_k\mu_k = \bar{r}_i(\mu_i + \mu_j)/2, \tag{44}$$

Since μ_j is equal to μ_i mirrored at μ_k by construction, we can solve Eq. 44 for the appropriate mean cosine:

$$\mu_j = 2\mu_k\mu_k^T\mu_i - \mu_i \quad \Leftrightarrow \quad \mu_i + \mu_j = 2\mu_k\mu_k^T\mu_i. \tag{45}$$

$$\Rightarrow \quad \bar{r}_k\mu_k = \bar{r}_i\mu_k\mu_k^T\mu_i \quad \Leftrightarrow \quad \bar{r}_i = \bar{r}_j = \frac{\bar{r}_k}{\mu_k^T\mu_i}. \tag{46}$$

The concentration parameter κ_k follows from the regular MLE step.