

# Guiding-Based Importance Sampling for Walk on Stars

TIANYU HUANG, School of Software and BNRist, Tsinghua University, China

JINGWANG LING, School of Software and BNRist, Tsinghua University, China

SHUANG ZHAO, University of California Irvine, United States of America

FENG XU\*, School of Software and BNRist, Tsinghua University, China

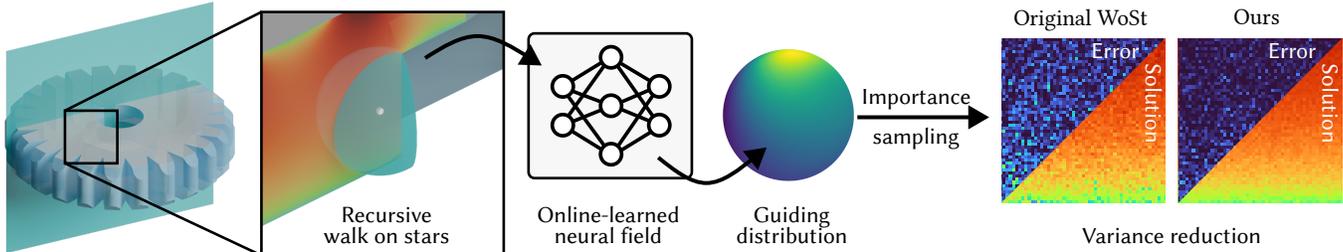


Fig. 1. An overview of our core method: We utilize an online-learned neural field to model the space-conditioned importance sampling distribution (guiding distribution) for the recursive term of the walk on stars (WoSt) estimator, achieving effective variance reduction.

*Walk on stars* (WoSt) has shown its power in being applied to Monte Carlo methods for solving partial differential equations, but the sampling techniques in WoSt are not satisfactory, leading to high variance. We propose a guiding-based importance sampling method to reduce the variance of WoSt. Drawing inspiration from path guiding in rendering, we approximate the directional distribution of the recursive term of WoSt using online-learned parametric mixture distributions, decoded by a lightweight neural field. This adaptive approach enables importance sampling the recursive term, which lacks shape information before computation. We introduce a reflection technique to represent guiding distributions at Neumann boundaries and incorporate multiple importance sampling with learnable selection probabilities to further reduce variance. We also present a practical GPU implementation of our method. Experiments show that our method effectively reduces variance compared to the original WoSt, given the same time or the same sample budget. Code and data for this paper are at <https://github.com/tyanyuy3125/elaina>.

CCS Concepts: • **Mathematics of computing** → **Partial differential equations; Probabilistic algorithms**; • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: Importance sampling, walk on stars, Monte Carlo methods

\*Feng Xu is the corresponding author.

Authors' Contact Information: [Tianyu Huang](mailto:tyanyu@mails.tsinghua.edu.cn), School of Software and BNRist, Tsinghua University, Beijing, China, [huang-ty21@mails.tsinghua.edu.cn](mailto:tyanyu@mails.tsinghua.edu.cn); [Jingwang Ling](mailto:lingjw20@mails.tsinghua.edu.cn), School of Software and BNRist, Tsinghua University, Beijing, China, [lingjw20@mails.tsinghua.edu.cn](mailto:lingjw20@mails.tsinghua.edu.cn); [Shuang Zhao](mailto:shz@ics.uci.edu), University of California Irvine, Irvine, United States of America, [shz@ics.uci.edu](mailto:shz@ics.uci.edu); [Feng Xu](mailto:xufeng2003@gmail.com), School of Software and BNRist, Tsinghua University, Beijing, China, [xufeng2003@gmail.com](mailto:xufeng2003@gmail.com).

*SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada*

© 2025 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada*, <https://doi.org/10.1145/3721238.3730593>.

## ACM Reference Format:

Tianyu Huang, Jingwang Ling, Shuang Zhao, and Feng Xu. 2025. Guiding-Based Importance Sampling for Walk on Stars. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3721238.3730593>

## 1 Introduction

Partial differential equations (PDEs) are commonly used to model various phenomena such as heat transfer, wave propagation, and fluid dynamics in physics, engineering, and computer graphics. Recently, Monte Carlo methods for solving PDEs have been gaining attention due to their advantages, such as eliminating the need for meshing, handling intricate geometries, and supporting local evaluations. This series of methods began with [Sawhney and Crane \[2020\]](#), who applied *walk on spheres* (WoS) [[Muller 1956](#)] to the Dirichlet Poisson problem, which was later extended by *walk on stars* (WoSt) [[Miller et al. 2024b](#); [Sawhney et al. 2023](#)] to support more boundary conditions. These advancements have found broad applications, including volume rendering [[Qi et al. 2022](#)], fluid simulation [[Jain et al. 2024](#); [Rioux-Lavoie et al. 2022](#)], heat simulation [[De Lambilly et al. 2023](#)], robotics [[Muchacho and Pokorny 2024](#)], machine learning [[Nam et al. 2024](#)], shape modeling [[de Goes and Desbrun 2024](#)], and infrared rendering [[Bati et al. 2023](#)].

Unfortunately, WoSt suffers from high variance and slow convergence, especially for problems with complex domains or boundary conditions. To address this problem, several recent methods [[Bakbouk and Peers 2023](#); [Li et al. 2023](#); [Miller et al. 2023](#); [Qi et al. 2022](#)] have adopted variance reduction techniques originated in Monte Carlo rendering [[Müller et al. 2021](#); [Veach and Guibas 1995a](#)] for WoSt. The importance sampling family is effective for variance reduction in rendering, and so far, there have been explorations to importance sample the non-recursive Neumann and source contributions [[Sawhney and Crane 2020](#); [Sawhney et al. 2023](#)] of WoSt.

However, the recursive term of WoSt, lacking a closed form or shape information before computation, poses a challenge for traditional importance sampling techniques such as next event estimation [Immel et al. 1986; Kajiya 1986]. As it currently stands, the recursive term can only be sampled using uniform directional sampling.

We believe adaptive importance sampling techniques can handle the recursive term, as their online learning approach enables operation without prior knowledge of the distribution shape or modifications to the formulation. Here, a representative approach in rendering is path guiding [Herholz et al. 2019; Müller et al. 2017; Vorba et al. 2014] which uses the observations from earlier samples to optimize the importance sampling distribution (guiding distribution), and uses the distribution to generate subsequent samples. Furthermore, recent advances using neural fields to encode space-conditioned parametric guiding distributions [Dong et al. 2023; Huang et al. 2024] have demonstrated exceptional performance in variance reduction, indicating the potential to apply path guiding to WoSt for better reducing variance.

In this paper, we introduce a guiding-based importance sampling method to reduce the variance of WoSt. Specifically, we approximate the directional distribution of the recursive term of WoSt using von Mises-Fisher (vMF) mixtures. The mixtures are decoded by a lightweight neural field that takes space coordinates as input. Then we employ gradient-based online learning on the neural field to optimize the mixtures. This approach effectively handles high-frequency variations in the solution field. Besides, we propose a sample reflection technique on the mixtures to represent guiding distributions at Neumann boundaries. This technique effectively reduces variance around Neumann boundaries compared to trivial approaches. Additionally, we explore multiple importance sampling (MIS) with learnable selection probabilities in WoSt, which guarantees unbiasedness and adaptively adjusts the balance between guiding and uniform directional sampling, further reducing variance. We present a practical GPU implementation of our method, enabling efficient parallelization including neural field training and geometric queries.

Our contributions are:

- a guiding-based importance sampling method for the WoSt estimator,
- a sample reflection technique for guiding distributions at Neumann boundaries, and
- learnable selection probabilities to balance guiding and uniform directional sampling in WoSt.

These contributions address the unique challenges of WoSt, which are discussed in detail in Section 2.3. We evaluate our method using several 2D and 3D problems under equal time or equal sample settings, demonstrating its effectiveness in variance reduction.

## 2 Related Work

### 2.1 Monte Carlo Methods for Solving PDEs

*PDE Estimators and Their Applications.* Recent exploration of Monte Carlo methods for solving PDEs [Sawhney and Miller 2024] has gained significant attention in the graphics community, as they offer advantages over traditional methods like finite element (FEM)

and finite difference (FDM) by circumventing spatial discretization challenges while providing better flexibility and performance. The pioneering work, Monte Carlo Geometry Processing [Sawhney and Crane 2020] revisits *walk on spheres (WoS)* [Muller 1956] to solve linear elliptic equations with Dirichlet boundary conditions, which is later extended to *walk on stars (WoSt)* [Ermakov and Sipin 2009; Sawhney et al. 2023; Simonov 2008] to handle Neumann and Robin [Miller et al. 2024b] boundary conditions. Under the WoS(t) frameworks, the methods are further generalized to address problems with spatially varying coefficients [Sawhney et al. 2022], surface PDEs [Sugimoto et al. 2024b], and infinite domains [Nabizadeh et al. 2021]. Monte Carlo methods have demonstrated broad applicability in both forward [Bati et al. 2023; de Goes and Desbrun 2024; De Lambilly et al. 2023; Jain et al. 2024; Muchacho and Pokorny 2024; Nam et al. 2024; Rioux-Lavoie et al. 2022] and inverse [Miller et al. 2024a; Yilmazer et al. 2024; Yu et al. 2024] PDE problems. In parallel with WoS(t), *walk on boundary* [Sugimoto et al. 2023], another Monte Carlo estimator for PDEs, has been revisited and applied to fluid simulations [Sugimoto et al. 2024a].

*Variance Reduction Techniques.* Similar to Monte Carlo rendering, Monte Carlo PDE solvers face the challenges of slow convergence and high variance. Various methods have been proposed to address these issues. Reverse walk splatting [Qi et al. 2022] and neural caches [Li et al. 2023] are effective but biased. Neural control variates [Li et al. 2024] achieve good wall-time performance, but incur substantial computational overhead, making them impractical for real-time visualization. Meanwhile, mean value caching [Bakboub and Peers 2023] and boundary value caching [Miller et al. 2023] adopt novel formulations to reduce variance, but their recursive terms are still sampled uniformly, thus orthogonal to our method. So far, no existing method has performed importance sampling on the recursive term of WoSt. Our method explores using online-learned mixtures to perform importance sampling on the recursive term to reduce variance, practically and unbiasedly.

### 2.2 Path Guiding in Rendering

In Monte Carlo rendering, path guiding is a data-driven adaptive importance sampling scheme which learns the importance sampling distribution (guiding distribution) from previous samples to improve subsequent samples, thus reducing variance. Research in path guiding primarily focuses on how to represent, store, and optimize the guiding distribution across spatial, and recently, temporal domain [Dong et al. 2024]. Early attempts in this field include constructing spatially cached histograms [Jensen 1995], cones [Hey and Purgathofer 2002] or Gaussian mixtures [Vorba et al. 2014]. A well-known recent work is Practical Path Guiding [Müller et al. 2017], which utilizes SD-trees to implement path guiding practical for production environments. Subsequent works have considered volume rendering [Herholz et al. 2019], caustics [Fan et al. 2023; Li et al. 2022], path space [Reibold et al. 2018], variance-aware sampling [Rath et al. 2020], spatio-directional mixture models [Dodik et al. 2022], and differentiable rendering [Fan et al. 2024]. In the deep learning era, path guiding based on neural networks have also been explored, such as employing convolutional neural networks to reconstruct radiance fields [Huo et al. 2020; Zhu et al.

2021], or using invertible neural networks to model complex distributions [Müller et al. 2019]. Recently, methods utilizing neural fields to encode parameterized guiding distributions [Dong et al. 2023; Huang et al. 2024] have emerged as the state-of-the-art in the field of path guiding. Our work is inspired by path guiding in rendering, with unique improvements tailored for 2D and 3D PDE problems, including reflection transformations on the guiding distribution to handle Neumann boundaries and learnable selection probabilities to combine uniform directional sampling and guiding adaptively.

### 2.3 WoSt-Specific Challenges Beyond Light Transport

**2.3.1 Parallax Property.** In rendering, the sampling region changes smoothly as viewpoint shifts. In contrast, WoSt’s star-shaped sampling region exhibits abrupt deformations around non-convex geometry, expanding or shrinking dramatically with small evaluation-point movements. This instability challenges traditional guiding structures [Ruppert et al. 2020]. This explains our empirical choice of neural fields.

**2.3.2 Boundary Handling.** Rendering methods often exploit cosine-weighted BRDFs to bound sampling domains [Diolatzis et al. 2020]. However, WoSt lacks a BRDF analogue, and its Neumann boundaries may produce sharp, high-value distributions at grazing angles—unlike rendering, where BRDFs typically diminish at such angles. This discrepancy explains our adoption of sample reflection at Neumann boundaries.

**2.3.3 Selection Probability.** While learnable selection probability is common in rendering [Diolatzis et al. 2020; Huang et al. 2024; Müller et al. 2019], it is used to combine guiding distributions and BSDFs, and the suitable distribution to combine for WoSt has not been proposed. We observe that, as the solution field of linear elliptic equations is smooth, the distributions on many small star-shaped regions near Dirichlet boundaries or Neumann silhouettes are close to uniform distribution. This consideration motivates performing learnable selection with uniform sampling for WoSt.

## 3 Background

### 3.1 Linear Elliptic Equations

Walk on stars (WoSt) primarily targets linear elliptic equations, which encompass a wide variety of forms. However, the structure of WoSt remains largely consistent across different formulations. We thus focus on the most typical case—the Poisson equation with Dirichlet and Neumann boundary conditions:

$$\begin{aligned} \Delta u(x) &= f(x) && \text{on } \Omega, \\ u(x) &= g(x) && \text{on } \partial\Omega_D, \\ \frac{\partial u(x)}{\partial n_x} &= h(x) && \text{on } \partial\Omega_N, \end{aligned} \quad (1)$$

where the boundary of the domain  $\Omega \subset \mathbb{R}^d$  ( $d = 2, 3$ ) is partitioned into a Dirichlet part  $\partial\Omega_D$  and a Neumann part  $\partial\Omega_N$  with prescribed values  $g$  and  $h$  (resp.).  $\Delta$  is the negative-semidefinite Laplacian,  $u : \Omega \rightarrow \mathbb{R}$  is the unknown solution, and  $f : \Omega \rightarrow \mathbb{R}$  is a source term. This equation can describe, e.g., the steady-state temperature distribution, where  $f$  represents the heat source or sink,

$g$  corresponds to the temperature on the boundary, and  $h$  denotes the heat flux on the boundary.

### 3.2 The Walk on Stars (WoSt) Estimator

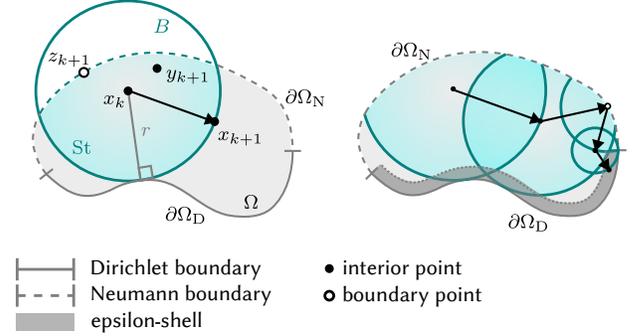


Fig. 2. Illustration of the WoSt estimator in 2D. *Left:* An illustration of a step.  $x_k$  is the query point,  $x_{k+1}$  is the next walk location,  $y_{k+1}$  is the source sample point, and  $z_{k+1}$  is the Neumann boundary sample point. *Right:* An entire walk of WoSt, which iteratively samples the next walk location until it reaches the Dirichlet boundaries’  $\epsilon$ -shell.

We define the  $\epsilon$ -shell  $\partial\Omega_D^\epsilon := \{x \in \Omega : \min_{y \in \partial\Omega_D} \|x - y\| \leq \epsilon\}$  for a manually specified small  $\epsilon > 0$ , which is the region where the walks terminate. The solution  $u(x_k)$  at any query point  $x_k \in \Omega \setminus \partial\Omega_D^\epsilon$  for Eq. 1 can be obtained using the following single-sample WoSt estimator [Sawhney et al. 2023]:

$$\begin{aligned} \langle u(x_k) \rangle &= \frac{P^B(x_k, x_{k+1}) \langle u(x_{k+1}) \rangle}{\alpha(x_k) p^{\partial\text{St}(x_k, r)}(x_{k+1})} - \langle N \rangle + \langle S \rangle, \\ \langle N \rangle &= \frac{G^B(x_k, z_{k+1}) h(z_{k+1})}{\alpha(x_k) p^{\partial\text{StN}(x_k, r)}(z_{k+1})}, \\ \langle S \rangle &= \frac{G^B(x_k, y_{k+1}) f(y_{k+1})}{\alpha(x_k) p^{\text{St}(x_k, r)}(y_{k+1})}, \end{aligned} \quad (2)$$

where  $\langle N \rangle$  and  $\langle S \rangle$  are non-recursive Neumann and source contributions, resp.;  $B$  is a ball centered at  $x_k$  with radius  $r$ ; the radius  $r$  equals to the smaller value between the distance from  $x_k$  to  $\partial\Omega_D$  and the distance from  $x_k$  to the nearest silhouette on  $\partial\Omega_N$ ; the star-shaped region  $\text{St} := B \cap \Omega$ ;  $\partial\text{St}$  denotes the boundaries of  $\text{St}$ ;  $\partial\text{StN}$  denotes the Neumann boundaries of  $\text{St}$ ;  $\alpha(x_k)$  is set to 1 if  $x_k$  lies within  $\text{St}$ ,  $1/2$  if it lies on the boundary of  $\text{St}$ , and 0 if it lies outside  $\text{St}$ ;  $G^B$  denotes the Green’s function defined over the sphere  $B$  [Sawhney et al. 2023, Eq. 24], while  $P^B$ , the Poisson kernel on  $B$ , is defined as  $P^B = \partial G^B / \partial n$ ;  $p$  represents the PDF of the sampler. The illustration of this single-sample Monte Carlo estimator can be found in Fig. 2.

In a walk, the recursive estimator begins from an arbitrary evaluation point within  $\Omega \setminus \partial\Omega_D^\epsilon$ ; at each step, the estimator performs up to three sampling operations around the query point  $x_k$ :

- (1) Sample source contribution at point  $y_{k+1} \in \text{St}$ .
- (2) Sample Neumann contribution at point  $z_{k+1} \in \partial\text{St} \cap \partial\Omega_N$ .
- (3) Sample the next walk location  $x_{k+1} \in \partial\text{St}$ .

The walk continues until it reaches  $\partial\Omega_D^\epsilon$ , where it uses the Dirichlet data  $g$  at the closest point  $\bar{x}_k \in \partial\Omega_D$  to set  $u(x_k) := g(\bar{x}_k)$ . The

first term of Eq. 2 is an unknown term that needs to be estimated recursively, similar to the scenario encountered in Monte Carlo rendering, thus we name it as *recursive term*. The recursive term lacks a closed form expression or shape information before computation, posing significant challenges for traditional importance sampling approaches. Therefore, in this paper, we employ guiding-based importance sampling at this term to achieve variance reduction.

### 3.3 von Mises-Fisher Mixture Model

To fit the target distribution (see Section 4.1) online for importance sampling, we adopt the von Mises-Fisher (vMF) mixture model. Since PDE problems are often solved in various dimensions, we adopt the generalized form of the vMF distribution. The vMF distribution on the  $(d-1)$ -sphere  $\mathbb{S}^{d-1}$  in  $\mathbb{R}^d$  ( $d=2,3$ ) is defined as:

$$v(v | \mu, \kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)} \exp(\kappa \mu^T v), \quad (3)$$

where  $v \in \mathbb{S}^{d-1}$  represents a direction,  $\kappa \in [0, +\infty)$  and  $\mu \in \mathbb{S}^{d-1}$  define the concentration and mean of the vMF distribution, *resp.*, and  $I_k$  denotes the modified Bessel function of the first kind at order  $k$ . We refer to Appendix A for specific forms of vMF distribution in different dimensions. The vMF mixture model is thus a convex combination of  $K$  vMF components:

$$\mathcal{V}(v | \Theta) = \sum_{i=1}^K \lambda_i \cdot v(v | \mu_i, \kappa_i), \quad (4)$$

where  $\Theta \in \mathbb{R}^{(2+d) \times K}$  is the set of mixture parameters, containing  $K$  vMF components, each with  $\lambda_i \in [0, 1]$ ,  $\kappa_i$  and  $\mu_i$ . Here, the mixture weight  $\lambda_i$  satisfies  $\sum_i \lambda_i = 1$ . The vMF mixture model is naturally defined on the sphere across various dimensions, which aligns with the directional nature of the recursive term.

For 3D problems, we use the sampling method for the mixtures from Tokuyoshi [2025], while for 2D problems, we use the method from Best and Fisher [1979].

## 4 Method

### 4.1 Importance Sampling the Recursive Term

We reparameterize the recursive term of the WoSt estimator (Eq. 2) using the unit-length vector  $v$ , yielding an estimator of the following form:

$$\langle u(x_k) \rangle = \frac{\langle u(v; x_k) \rangle}{\|\mathbb{S}^{d-1}\| \alpha(x_k) p(v | x_k)} - \langle N \rangle + \langle S \rangle, \quad (5)$$

where  $v = \frac{x_{k+1} - x_k}{\|x_{k+1} - x_k\|} \in \mathbb{S}^{d-1}$  ( $d=2,3$ ),  $\|\mathbb{S}^{d-1}\|$  is the area of  $\mathbb{S}^{d-1}$ , and  $p(v | x_k)$  is the directional PDF of the sampler *conditioned on*  $x_k$ . We refer to Appendix B for a detailed derivation. For  $x_k \in \partial\Omega_N$  with a normal<sup>1</sup>  $n(x_k)$ , the following condition of  $p$  is satisfied, reflecting the fact that Neumann boundaries only have one valid side in random walk:

$$p(v | x_k) = 0 \quad \text{when } v \cdot n(x_k) \leq 0. \quad (6)$$

<sup>1</sup>In this paper, we stipulate that the normal vector points toward the region where the walk is currently located. This assumption also applies to double-sided boundaries.

In this formulation, the PDF  $p_u$  of uniform directional sampling<sup>2</sup> adopted by the original WoSt is thus:

$$p_u(v | x_k) = \begin{cases} \frac{2H(v \cdot n(x_k))}{\|\mathbb{S}^{d-1}\|}, & \text{if } x_k \in \partial\Omega_N, \\ \frac{1}{\|\mathbb{S}^{d-1}\|}, & \text{otherwise,} \end{cases} \quad (7)$$

where  $H(\cdot)$  is the unit step function. To reduce the variance of the recursive term, as illustrated in Fig. 3, we aim to sample from an importance distribution defined on  $\mathbb{S}^{d-1}$ , referred to as the *guiding distribution*  $p_g$ , that approximately satisfies

$$p_g(v | x_k) \propto |u(v; x_k)|. \quad (8)$$

We denote the right-hand side of Eq. 8 as  $p_t(v | x_k)$ , *i.e.*, the *target distribution*. Next, we will discuss how to fit the guiding distribution to the target distribution.

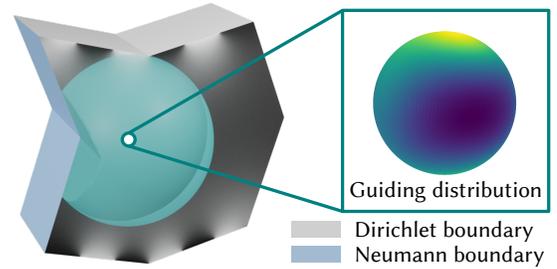


Fig. 3. Illustration of the guiding distribution at an arbitrary step. The guiding distribution is defined on  $\mathbb{S}^{d-1}$  ( $d=2,3$ ), with a goal to be proportional to  $|u|$  on  $\partial\Omega$ . The sampler uses this distribution for importance sampling.

### 4.2 Representing and Learning the Guiding Distribution

We adopt vMF mixture model (Eq. 4) as the guiding distribution (Eq. 8) conditioned on  $x \notin \partial\Omega_N$ :

$$p_g(v | x) = \mathcal{V}(v | \Theta(x)). \quad (9)$$

where  $\Theta(x)$  is the space-conditioned form of  $\Theta$  in Eq. 4. To represent it, we employ a neural field  $\text{NN}(x | \Phi)$  with trainable parameters  $\Phi$  to output its predicted values  $\hat{\Theta}(x)$ :

$$\text{NN}(x | \Phi) = \hat{\Theta}(x), \quad (10)$$

where NN consists of a multi-resolution feature grid [Hadadan et al. 2021] and a lightweight multi-layer perceptron (MLP). The inference and training procedure is illustrated in Fig. 4. Given a query point  $x$ , the neural field first encodes it through the multi-resolution feature grid, producing a feature vector that is then passed into the MLP. The MLP outputs a tensor of dimension  $(\dim(\Theta(x)) + 1)$ , containing the unnormalized parameters  $\hat{\Theta}'(x)$  of the mixtures, and the selection probability  $c'$  (see Section 4.4). To ensure that each output component lies within a valid range, we apply a normalization mapping, as detailed in Tab. 1. The resulting mixtures with valid predicted parameters  $\hat{\Theta}(x)$  are then used to sample the direction of the next walk location.

<sup>2</sup>The WoSt paper refers to this approach as *importance sampling the Poisson kernel*. To avoid confusion with our method, we consistently refer to the sampling approach used in the original WoSt as *uniform directional sampling* throughout this paper.

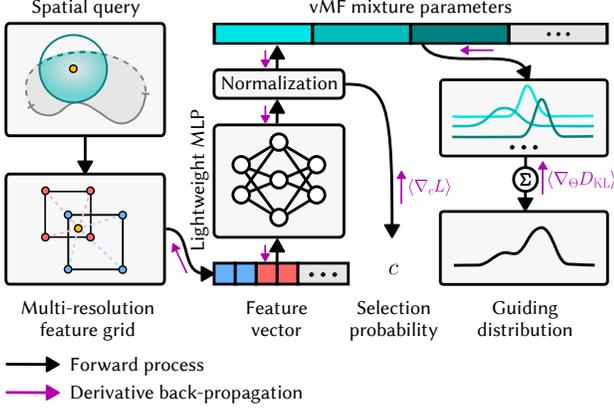


Fig. 4. Our neural field inference and training pipeline. During inference, the spatial query coordinates are encoded by a multi-resolution feature grid to produce feature vectors. These vectors are then fed into a lightweight MLP to obtain the parameters of the guiding distribution  $\hat{\Theta}$  (Eq. 10) and a selection probability  $c$  (Eq. 15). During training,  $\langle \nabla_{\Theta} D_{KL} \rangle$  (Eq. 13) for the guiding distribution and  $\langle \nabla_c L \rangle$  (Eq. 17) for the selection probability are back-propagated all the way to the grid and MLP parameters.

Table 1. Neural field outputs, and their corresponding normalization mappings. We refer to Section 3.3 for explanations of mixture parameters.

Parameter	NN output	Normalization mapping
$\mu_i \in \mathbb{S}^{d-1}$	$\mu'_i \in \mathbb{R}^d$	$\mu_i = \mu'_i / \ \mu'_i\ $
$\kappa_i \in [0, +\infty)$	$\kappa'_i \in \mathbb{R}$	$\kappa_i = \exp(\kappa'_i)$
$\lambda_i \in (0, 1)$	$\lambda'_i \in \mathbb{R}$	$\lambda_i = \exp(\lambda'_i) / \sum_{j=1}^K \exp(\lambda'_j)$
$c \in (0, 1)$ (Eq. 15)	$c' \in \mathbb{R}$	$c = \text{sigmoid}(c')$

To fit the vMF mixture model  $\mathcal{V}$  to the target distribution  $p_t$  (see Section 4.1) at  $x$ , we introduce the Kullback-Leibler (KL) divergence as

$$D_{KL}(p_t \parallel \mathcal{V}; \Theta) = \int_{\mathcal{V}} p_t(v) \log \frac{p_t(v)}{\mathcal{V}(v \mid \hat{\Theta})} dv. \quad (11)$$

The optimization objective of the neural-field parameters  $\Phi$  is thus

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \mathbb{E}_x \left[ D_{KL}(p_t(x) \parallel \mathcal{V}; \Theta(x)) \right]. \quad (12)$$

We use a single-sample Monte Carlo estimator to estimate  $\nabla_{\Theta} D_{KL}$ :

$$\langle \nabla_{\Theta} D_{KL}(p_t \parallel \mathcal{V}; \Theta) \rangle = - \frac{p_t(v) \nabla_{\Theta} \mathcal{V}(v \mid \hat{\Theta})}{\tilde{p}(v \mid \hat{\Theta}) \mathcal{V}(v \mid \hat{\Theta})}, \quad (13)$$

where  $\tilde{p}$  is the PDF of the sampler given by Eq. 15. We back-propagate this derivative along the purple arrows in Fig. 4, updating the parameters  $\Phi$  of both the MLP and the multi-resolution feature grid using gradient-based optimization [Kingma 2014].

### 4.3 Sample Reflection at Neumann Boundaries

For  $x \in \partial\Omega_N$ , samples from the original mixtures span the full sphere thus do not satisfy Eq. 6. Therefore, we reflect invalid samples along

the local tangent plane, resulting in the following PDF:

$$p_g(v \mid x) = \begin{cases} 0, & \text{if } v \cdot n(x) \leq 0, \\ \mathcal{V}(v^+ \mid \Theta(x)) + \mathcal{V}(v^- \mid \Theta(x)), & \text{otherwise,} \end{cases} \quad (14)$$

where  $v^+ = v$ , and  $v^-$  is the reflection of  $v^+$  off  $\partial\Omega_N$  with normal  $n(x)$ , i.e.  $v^- = v^+ - 2(v^+ \cdot n(x))n(x)$ .

We have discussed how the strategy we adopt differs from similar strategies in rendering and fits the PDE problems in Section 2.3.2. Section 6.3.2 gives an ablation.

### 4.4 Multiple Importance Sampling with Learnable Selection Probabilities

Linear elliptic equations often contain relatively smooth regions. In these regions, uniform directional sampling is sometimes the optimal approach, whereas learned vMF mixtures might yield less accurate approximation. Moreover, solely relying on the learned distribution for importance sampling is an unstable strategy, potentially introducing variance or even bias [Owen and Zhou 2000]. To address this, we introduce a learnable multiple importance sampling (MIS) method. Our approach is based on the single-sample balance heuristic [Veach and Guibas 1995b], with the following MIS PDF:

$$\tilde{p}(v \mid x) = c(x)p_g(v \mid x) + (1 - c(x))p_u(v \mid x), \quad (15)$$

where  $c(x)$  is the learnable selection probability, decoded from neural field output (Tab. 1). Following Müller et al. [2019], our method learns  $c(x)$  with the following loss function:

$$L = eD_{KL}(p_t \parallel \tilde{p}) + (1 - e)D_{KL}(p_t \parallel p_g), \quad (16)$$

where  $e$  is a fixed fraction that we set to 0.2. The single-sample Monte Carlo estimator for  $\nabla_c L$  for back-propagation is:

$$\langle \nabla_c L \rangle = - \frac{ep_t(v)(p_g(v \mid x) - p_u(v \mid x))}{\tilde{p}^2(v \mid x)}. \quad (17)$$

We find that performing our learnable MIS in combination with uniform directional sampling is sufficient to achieve excellent performance, as demonstrated in Section 6.3.3. Currently, there are no alternative importance sampling distributions for the recursive term of WoSt. We anticipate that if such distributions are developed, they could be integrated with our method using this approach. We refer to Section 7 for further discussion.

## 5 Implementation Details

### 5.1 Wavefront-style Monte Carlo PDE Solver

Our guiding-based method (see Section 4.2) consists of neural field inference and training, which benefits from batched input for parallelization. A common strategy in rendering for efficient batching is the wavefront-style architecture [Laine et al. 2013], where rays are generated and processed in batches. In light of this, we implement a wavefront-style Monte Carlo PDE solver on GPU.

As illustrated in Fig. 5, at each step of WoSt, our wavefront-style solver is divided into three stages:

- (1) Logic Stage: The distances to the nearest Dirichlet boundary and the Neumann silhouette are queried to compute the ball radius  $r$  (see Section 3.2). Walks are then partitioned based on whether they fall within the  $\epsilon$ -shell.

- (2) Evaluation Stage: This stage resembles the Material Stage in wavefront-style renderers. For walks inside the  $\epsilon$ -shell, contributions are evaluated on the nearest Dirichlet boundary. For walks outside the  $\epsilon$ -shell, contributions from source and Neumann boundaries are evaluated.
- (3) Walk Stage: Analogous to the Ray Cast Stage in rendering, this stage samples the next walk location and updates throughput accordingly.

As each stage consists of multiple GPU kernels, we use a Structure-of-Arrays (SoA) memory layout to transfer data between kernels and employ thread-safe queues to manage tasks for each kernel.

## 5.2 Network Design and Implementation

*Neural Field Architecture.* To effectively capture the spatial variation of the space-conditioned guiding distribution, we use a hybrid architecture combining learnable spatial embeddings and a small MLP to encode the parametric mixtures. We define  $L$  embedding grids  $G_l$ , where  $l = 1, 2, \dots, L$ , to form a multi-resolution representation. Each grid spans the  $d$ -dimensional space of the problem with a spatial resolution of  $D_l^d$ . A learnable embedding vector  $\omega \in \mathbb{R}^F$  is associated at each lattice point of  $G_l$ . To retrieve the spatial embedding for a point  $x$ , we perform bi-linear interpolation at neighboring lattice points for each resolution, and concatenate the resulting embedding vectors to form the full embedding  $G(x)$ :

$$G(x | \Phi) = \bigoplus_{l=1}^L \text{bilinear}(x, V_l[x]), \quad G : \mathbb{R}^d \rightarrow \mathbb{R}^{L \times F} \quad (d = 2, 3), \quad (18)$$

where  $V_l[x]$  is the embedding vectors at the corners of the cell enclosing  $x$  within  $G_l$ , and  $\bigoplus$  is the concatenation operation.  $G(x)$  is subsequently mapped by an MLP with 3 layers, each containing 64 neurons. Trainable parameters  $\Phi$  in Eq. 10 consist of the learnable spatial embeddings and MLP weights. We implement the neural field based on *tiny-cuda-nn* [Müller 2021], adopting DenseGrid as the encoding method of the point coordinates. We employ ReLU as the activation function of the MLP, and the output is normalized by mapping in Tab. 1, producing  $\hat{\Theta}(x)$  in Eq. 10 and  $c(x)$  in Eq. 15. We refer to Appendix D for detailed configurations of the neural field.

*Online Training Scheme.* Similar to Monte Carlo rendering, walks are evaluated in batches. Each batch consists of one entire walk per evaluation point. Once all walks in a batch are completed, a training stage updates the neural field using the information gathered. The updated neural field is then used to sample the next batch. Once the walks per point (wpp) reaches a certain threshold, typically 256 wpp, the guiding distribution converges. At this time, we terminate the training process and use the neural field exclusively for inference, further improving performance.

*System Integration.* As shown in Fig. 5, we insert the neural field inference stage before the walk stage and integrate the neural field training stage at the end of each walk loop. Our neural field and the WoSt integrator operate on separate CUDA streams, allowing the network inference and the evaluation stages to execute in parallel within the step loop. This approach maximizes the batch size during inference and training, avoiding the inefficiencies associated with single-sample inference or training.

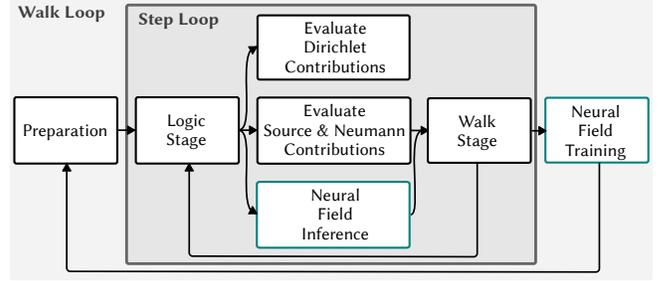


Fig. 5. Illustration of our wavefront-style Monte Carlo PDE solver including neural field inference and training stages.

## 5.3 Geometric and Source Queries

To demonstrate the practicality of our method, it is essential to implement a system with strong performance. The performance bottleneck of Monte Carlo PDE solvers typically lies in geometric queries. The authors of WoSt provide *Zombie* [Sawhney and Miller 2023] as the algorithm implementation and FCPW [Sawhney 2021] for geometric queries. However, *Zombie* is a CPU-only implementation, and although FCPW offers GPU support via Slang [He et al. 2018], integrating Slang with *tiny-cuda-nn* presents significant challenges.

To address this, we develop a CUDA-based geometric and source query library. For geometric queries, we reference the implementation of WoBToolbox [Sugimoto et al. 2023] and create a comprehensive query library for WoSt using a linear BVH [Karras 2012]. While *Zombie* uses a dense grid for source queries in 2D, which is memory inefficient if applied in 3D, we leverage NanoVDB [Museth 2021], a widely adopted GPU-based library for sparse volumetric storage and efficient source queries in 2D and 3D.

## 6 Results and Discussion

Since we perform validation on visualization tasks and the experimental results exhibit large variations in scale, we employ the relative mean squared error (relMSE) as our quantitative metric. All experiments are performed on an AMD EPYC 9754 128-Core Processor with an RTX 4090D GPU. We set the hyperparameter  $K = 8$  except for Section 6.3.1. We refer to Appendix E for the complete quantitative results at 1024 wpp of all experiments.

Since our method is the first importance sampling approach targeting the recursive term of WoSt, we find it reasonable to validate its effectiveness by comparing with the original WoSt. In Appendix C, we provide additional comparisons with some existing variance reduction methods [Li et al. 2023; Qi et al. 2022].

### 6.1 Comparison with the Original WoSt

For a fair equal-time comparison, we implement both our method and the original WoSt on the same system, and disable other sampling techniques such as control variates, adaptive sampling, and stratified sampling. Russian roulette is enabled only when the walk length exceeds 128 to prevent infinite walks in scenarios involving Neumann boundaries.

All experiments in this section are conducted with an evaluation grid of  $1024 \times 1024$ , running 1024 walks per point (wpp). Our approach does not involve pretraining and uses all samples to form the final result, ensuring no additional samples are required compared to the baseline.

**6.1.1 3D Geometry.** We adopt solving on slices [Sawhney and Crane 2020, Section 5.2] as the visualization method. For the dataset, following the experimental design of Li et al. [2023], we create six 3D problems to compare performance on Dirichlet boundaries (Fig. 7, columns 1–2), source (column 3), Neumann boundaries (columns 4–5), and problems involving signed values (column 6).

Qualitative and quantitative results are presented in Fig. 8 and Fig. 9, while Fig. 10 and Fig. 11 illustrate the relMSE as a function of wpp and as a function of time, *resp.* The results show that our method outperforms the original WoSt both qualitatively and quantitatively in all problems. Although our method incurs some performance costs, requiring longer runtimes at the same wpp, the results under equal time (Fig. 11) demonstrate that the variance reduction effect of our method compensates for the runtime overhead.

**6.1.2 2D Diffusion Curves.** We use the diffusion curves [Orzan et al. 2008] as our 2D dataset, which employs Dirichlet boundaries to fill colors in vector graphics. Neumann boundaries are added at the bounding box to form closed domains. We select two representative problems: *Fille* and *Ladybug*. These problems exhibit distinct characteristics: *Fille* has more pronounced light–dark variations; while *Ladybug* features smoother variations, providing a problem more favorable for the baseline.

We report the qualitative and quantitative results in Fig. 12 (equal-sample) and Fig. 13 (equal-time). For the *Ladybug* problem, initially, uniform directional sampling better matches the problem’s smoothness compared to the initialization of the guiding distribution. The learnable selection probability ensures that, in the early stages of training, unsuitable guiding distributions do not significantly harm the sampling.

## 6.2 Evaluation

**6.2.1 Training Batch Size.** Our method uses a fraction of batches for training (see Section 5.2). To assess the impact of training batch size, we vary the training batch sizes to 64, 128, 256, and 512 wpp, and compare their relMSE and runtime at 1024 wpp. The rest of the experimental setup follows Section 6.1. Fig. 14 indicates that the training steps have a significant impact on runtime, primarily because training steps are blocking (Fig. 5). Besides, our model converges relatively quickly, with the performance improvement from 128 wpp to 256 wpp being smaller than that from 64 wpp to 128 wpp. Therefore, using 256 wpp as the training batch size proves to be a reasonable choice.

**6.2.2 Runtime Breakdown.** We measure the time consumption of training, inference, system overhead, and the original WoSt logic across all problems at 1024 wpp. The results are shown in Fig. 6. The results demonstrate that our method incurs low time overhead, and has substantial room for further system-level optimization.

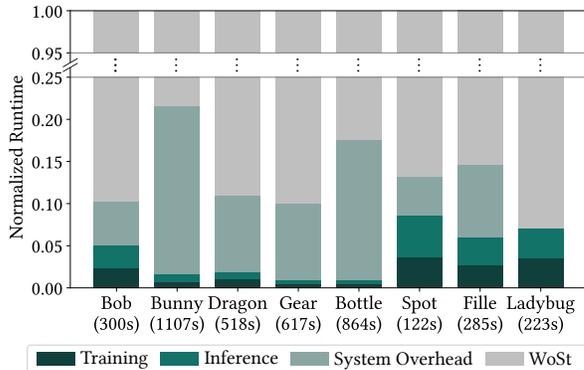


Fig. 6. Runtime breakdown. The runtime of each execution stage is normalized by the total execution time. *Training* and *Inference* refer to the total time spent on the forward and backward passes of the neural network, *resp.* *System Overhead* represents the additional time spent due to synchronization operations and data structure manipulations introduced by *integrating* the neural network. *WoSt* denotes the time consumed by the original logic of WoSt. On average, our method incurs a 13% runtime overhead.

## 6.3 Ablation

**6.3.1 Number of vMF Components in the Mixture Model.** We set the number of vMF components  $K$ , to 4, 8, and 16, and compare their relMSE and runtime at 1024 wpp. As shown in Fig. 15, increasing the number of vMF components from 4 to 8 results in a significant improvement. However, further increasing  $K$  to 16 does not provide noticeable benefits and often leads to increased runtime.

**6.3.2 Sample Reflection at Neumann Boundaries.** We qualitatively evaluate results with or without reflecting samples at Neumann boundaries at 1024 wpp in Fig. 16. The experiment shows that the method without reflection results in worse variance around the Neumann boundary than the original WoSt, while introducing reflection significantly reduces the variance.

**6.3.3 Multiple Importance Sampling (MIS) with Learnable Selection Probabilities.** We conduct experiments on problems in Section 6.1, with the same setup except for different sampling configurations: uniform directional sampling only (selection probabilities  $c(x) \equiv 0$ ), fixed MIS selection probability ( $c(x) \equiv 0.5$ ), guided sampling only ( $c(x) \equiv 1$ ), and our learnable MIS ( $c(x)$  is decoded from the neural field). We report the results in Tab. 2. The results show that our learnable MIS strategy achieves the best performance.

Table 2. Quantitative results (relMSE↓) of the ablation study on MIS with learnable selection probabilities.

Problem	Uniform Only	Fixed $c(x) \equiv 0.5$	Guiding Only	Ours
Bob	0.00385	0.00181	0.00136	0.00098
Bunny	0.01511	0.00690	0.00591	0.00564
Dragon	0.00353	0.00158	0.00135	0.00088
Gear	0.00153	0.00085	0.00075	0.00072
Bottle	0.00459	0.00305	0.00288	0.00274
Fille	0.00523	0.00210	0.00241	0.00208

## 7 Limitations and Future Work

*Positivization.* Our method effectively reduces shape variance; however, for real-valued integrands, we cannot eliminate sign variance with a positive-valued PDF [Owen and Zhou 2000], as shown in Fig. 17. While recent differentiable rendering research [Belhe et al. 2024; Fan et al. 2024; Zeltner et al. 2021] has explored similar problems, we believe they are inapplicable to WoSt. Besides, existing Monte Carlo methods for PDE lack consideration of positivization, we therefore regard this issue as beyond our current scope and defer it to future work.

*Delta Function.* Our method alone cannot handle delta boundaries or source. We anticipate that combining our method with reverse walk splatting [Qi et al. 2022], or developing an analogy to next event estimation [Immel et al. 1986; Kajiya 1986] and integrating it via multiple importance sampling, could address this issue.

*More Estimators.* We omit discussions on Robin boundaries [Miller et al. 2024b], Kelvin-transformed domains [Nabizadeh et al. 2021] and other PDEs [Sawhney et al. 2022]. However, we believe that extending our method to these cases should not pose significant challenges. Additionally, there should also be a guiding method for walk on boundary [Sugimoto et al. 2023]. For the inverse PDE estimators, drawing from differentiable rendering, it is also expected to require different guiding strategies [Fan et al. 2024].

*Importance Sampling Other Terms.* Our method performs importance sampling on the recursive term. For the Neumann contribution term, there is already efficient sampling method based on SNCH trees [Sawhney et al. 2023]. For the source contribution term, using a certain form of light trees [Conty Estevez and Kulla 2018; Lin and Yuksel 2020] may be a good choice. Based on our framework, we may also guide the source sampling using a directional vMF distribution and a radial Beta distribution.

*Combination with Other Variance Reduction Methods.* Our method is theoretically orthogonal to existing methods. Integrating our method with existing approaches holds the potential to further reduce the variance of WoSt.

## 8 Conclusion

We propose a guiding-based method to importance sample the recursive term of the walk on stars (WoSt) estimator, drawing connections between Monte Carlo PDE solvers and Monte Carlo rendering. We approximate the directional distribution of the recursive term of WoSt by fitting a guiding distribution, parameterized by a neural field, to observations from previous walks. This distribution is then used to guide subsequent walks with improved sampling efficiency. A sample reflection technique is introduced to better shape the guiding distribution to align with the target distribution at Neumann boundaries. Additionally, the learnable selection probabilities adapts the balance between the uniform directional sampling and guiding to the local properties of the solution field, further reducing variance. Experiments under equal time and equal sample settings show that our method effectively reduces the variance with relatively low overhead. The expressive guiding distribution improves sampling efficiency, particularly at positions where the solution field exhibits

higher frequency. We anticipate further advancements in variance reduction for Monte Carlo PDE solvers and believe rendering techniques will have broader applications across other fields.

## Acknowledgments

We thank anonymous reviewers for their valuable feedback. The *Bob* model is provided by Keenan Crane; the *Dragon* and *Bunny* models are from the Stanford Computer Graphics Laboratory; the *Gear* model is courtesy of Hu et al. [2018]; and the *Spot* model is from Crane et al. [2013]. *Fille* and *Ladybug* are from Orzan et al. [2008].

This work was supported by the National Key R&D Program of China (2023YFC3305600), the Zhejiang Provincial Natural Science Foundation (LDT23F02024F02), and the NSFC (No.61822111, 62021002). This work was also supported by THUICS, Tsinghua University, and BLBCI, Beijing Municipal Education Commission.

Tianyu Huang was supported by Undergraduate Disruptive Innovation Talent Cultivation Program of Tsinghua University. He would like to thank Honghao Dong for valuable discussions, as well as Toshiya Hachisuka, Ryusuke Sugimoto, Li Yi, and EECG for their support.

## References

- Ghada Bakboub and Pieter Peers. 2023. Mean Value Caching for Walk on Spheres. In *Eurographics Symposium on Rendering*, Tobias Ritschel and Andrea Weidlich (Eds.). The Eurographics Association. <https://doi.org/10.2312/sr.20231120>
- Mégane Bati, Stéphane Blanco, Christophe Coustet, Vincent Eymet, Vincent Forest, Richard Fournier, Jacques Gautrais, Nicolas Mellado, Mathias Paulin, and Benjamin Piaud. 2023. Coupling Conduction, Convection and Radiative Transfer in a Single Path-Space: Application to Infrared Rendering. *ACM Trans. Graph.* 42, 4, Article 79 (July 2023), 20 pages. <https://doi.org/10.1145/3592121>
- Yash Belhe, Bing Xu, Sai Praveen Bangaru, Ravi Ramamoorthi, and Tzu-Mao Li. 2024. Importance Sampling BRDF Derivatives. *ACM Trans. Graph.* 43, 3, Article 25 (April 2024), 21 pages. <https://doi.org/10.1145/3648611>
- Donald Best and Nicholas Fisher. 1979. Efficient Simulation of the von Mises Distribution. *Journal of the Royal Statistical Society. Series C. Applied Statistics* 28 (01 1979). <https://doi.org/10.2307/2346732>
- Alejandro Conty Estevez and Christopher Kulla. 2018. Importance Sampling of Many Lights with Adaptive Tree Splitting. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2, Article 25 (Aug. 2018), 17 pages. <https://doi.org/10.1145/3233305>
- Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.
- Fernando de Goes and Mathieu Desbrun. 2024. Stochastic Computation of Barycentric Coordinates. *ACM Trans. Graph.* 43, 4, Article 42 (July 2024), 13 pages. <https://doi.org/10.1145/3658131>
- Auguste De Lambilly, Gabriel Benedetti, Nour Rizk, Chen Hanqi, Siyuan Huang, Junnan Qiu, David Louapre, Raphael Granier De Cassagnac, and Damien Rohmer. 2023. Heat Simulation on Meshless Crafted-Made Shapes. In *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games (Rennes, France) (MIG '23)*. Association for Computing Machinery, New York, NY, USA, Article 9, 7 pages. <https://doi.org/10.1145/3623264.3624457>
- Stavros Diolatzis, Adrien Gruson, Wenzel Jakob, Derek Nowrouzezahrai, and George Drettakis. 2020. Practical Product Path Guiding Using Linearly Transformed Cosines. In *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)* 39, 4 (July 2020).
- Ana Dodik, Marios Papas, Cengiz Öztireli, and Thomas Müller. 2022. Path Guiding Using Spatio-Directional Mixture Models. *Computer Graphics Forum* 41, 1 (2022), 172–189. <https://doi.org/10.1111/cgf.14428> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14428>
- Honghao Dong, Rui Su, Guoping Wang, and Sheng Li. 2024. Efficient Neural Path Guiding with 4D Modeling. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 21, 11 pages. <https://doi.org/10.1145/3680528.3687687>
- Honghao Dong, Guoping Wang, and Sheng Li. 2023. Neural Parametric Mixtures for Path Guiding. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 29, 10 pages. <https://doi.org/10.1145/3588432.3591533>

- S. Ermakov and A. Sipin. 2009. The “walk in hemispheres” process and its applications to solving boundary value problems. *Vestnik St. Petersburg University: Mathematics* 42 (09 2009), 155–163. <https://doi.org/10.3103/S10663454109030029>
- Zhimin Fan, Pengpei Hong, Jie Guo, Changqing Zou, Yanwen Guo, and Ling-Qi Yan. 2023. Manifold Path Guiding for Importance Sampling Specular Chains. *ACM Trans. Graph.* 42, 6, Article 257 (Dec. 2023), 14 pages. <https://doi.org/10.1145/3618360>
- Zhimin Fan, Pengcheng Shi, Mufan Guo, Ruoyu Fu, Yanwen Guo, and Jie Guo. 2024. Conditional Mixture Path Guiding for Differentiable Rendering. *ACM Trans. Graph.* 43, 4, Article 48 (July 2024), 11 pages. <https://doi.org/10.1145/3658133>
- Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. 2021. Neural radiosity. *ACM Transactions on Graphics* 40, 6 (dec 2021), 1–11. <https://doi.org/10.1145/3478513.3480569>
- Yong He, Kayvon Fatahalian, and Tim Foley. 2018. Slang: language mechanisms for extensible real-time shading systems. *ACM Trans. Graph.* 37, 4, Article 141 (July 2018), 13 pages. <https://doi.org/10.1145/3197517.3201380>
- 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. <https://doi.org/10.1145/3230635>
- Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics (Budmerice, Slovakia) (SCCG '02)*. Association for Computing Machinery, New York, NY, USA, 107–114. <https://doi.org/10.1145/584458.584476>
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages. <https://doi.org/10.1145/3197517.3201353>
- Jiawei Huang, Akito Iizuka, Hajime Tanaka, Taku Komura, and Yoshifumi Kitamura. 2024. Online Neural Path Guiding with Normalized Anisotropic Spherical Gaussians. *ACM Trans. Graph.* 43, 3, Article 26 (April 2024), 18 pages. <https://doi.org/10.1145/3649310>
- Yuchi Huo, Rui Wang, Ruzahng Zheng, Hualin Xu, Hujun Bao, and Sung-Eui Yoon. 2020. Adaptive Incident Radiance Field Sampling and Reconstruction Using Deep Reinforcement Learning. *ACM Trans. Graph.* 39, 1, Article 6 (Jan. 2020), 17 pages. <https://doi.org/10.1145/3368313>
- David S. Immel, Michael F. Cohen, and Donald P. Greenberg. 1986. A radiosity method for non-diffuse environments. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 133–142. <https://doi.org/10.1145/15922.15901>
- Pranav Jain, Ziyin Qu, Peter Yichen Chen, and Oded Stein. 2024. Neural Monte Carlo Fluid Simulation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 9, 11 pages. <https://doi.org/10.1145/3641519.3657438>
- Henrik Wann Jensen. 1995. Importance Driven Path Tracing using the Photon Map. In *Rendering Techniques*. <https://api.semanticscholar.org/CorpusID:9344202>
- James T. Kajiya. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*. Association for Computing Machinery, New York, NY, USA, 143–150. <https://doi.org/10.1145/15922.15902>
- Tero Karras. 2012. Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics Conference on High-Performance Graphics (Paris, France) (EGGH-HPG'12)*. Eurographics Association, Goslar, DEU, 33–37.
- Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- Samuli Laine, Tero Karras, and Timo Aila. 2013. Megakernels considered harmful: wavefront path tracing on GPUs. In *Proceedings of the 5th High-Performance Graphics Conference (Anaheim, California) (HPG '13)*. Association for Computing Machinery, New York, NY, USA, 137–143. <https://doi.org/10.1145/2492045.2492060>
- He Li, Beibei Wang, Changehe Tu, Kun Xu, Nicolas Holzschuch, and Ling-Qi Yan. 2022. Unbiased Caustics Rendering Guided by Representative Specular Paths. In *Proceedings of SIGGRAPH Asia 2022*.
- Zilu Li, Guandao Yang, Xi Deng, Christopher De Sa, Bharath Hariharan, and Steve Marschner. 2023. Neural Caches for Monte Carlo Partial Differential Equation Solvers. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) (SA '23). Association for Computing Machinery, New York, NY, USA, Article 34, 10 pages. <https://doi.org/10.1145/3610548.3618141>
- Zilu Li, Guandao Yang, Qingqing Zhao, Xi Deng, Leonidas Guibas, Bharath Hariharan, and Gordon Wetzstein. 2024. Neural Control Variates with Automatic Integration. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 10, 9 pages. <https://doi.org/10.1145/3641519.3657395>
- Daqi Lin and Cem Yuksel. 2020. Real-Time Stochastic Lightcuts. *Proc. ACM Comput. Graph. Interact. Tech. (Proceedings of I3D 2020)* 3, 1 (2020), 18 pages. <https://doi.org/10.1145/3384543>
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4 (2023).
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024a. Differential Walk on Spheres. *ACM Trans. Graph.* 43, 6 (2024).
- Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024b. Walkin' Robin: Walk on Stars with Robin Boundary Conditions. *ACM Trans. Graph.* 43, 4 (2024).
- Rafael I. Cabral Muchacho and Florian T. Pokorny. 2024. Walk on Spheres for PDE-based Path Planning. arXiv:2406.01713 [cs.RO] <https://arxiv.org/abs/2406.01713>
- Mervin E. Muller. 1956. Some Continuous Monte Carlo Methods for the Dirichlet Problem. *The Annals of Mathematical Statistics* 27, 3 (1956), 569 – 589. <https://doi.org/10.1214/aoms/1177728169>
- Thomas Müller. 2021. *tiny-cuda-nn*. <https://github.com/NVlabs/tiny-cuda-nn>
- Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical Path Guiding for Efficient Light-Transport Simulation. *Computer Graphics Forum (Proceedings of EGSR)* 36, 4 (June 2017), 91–100. <https://doi.org/10.1111/cgf.13227>
- 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. <https://doi.org/10.1145/3341156>
- Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *ACM Trans. Graph.* 40, 4, Article 36 (July 2021), 16 pages. <https://doi.org/10.1145/3450626.3459812>
- Ken Museth. 2021. NanoVDB: A GPU-Friendly and Portable VDB Data Structure For Real-Time Rendering And Simulation. In *ACM SIGGRAPH 2021 Talks* (Virtual Event, USA) (SIGGRAPH '21). Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. <https://doi.org/10.1145/3450623.3464653>
- Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin transformations for simulations on infinite domains. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 97:1–97:15.
- Hong Chul Nam, Julius Berner, and Anima Anandkumar. 2024. Solving Poisson Equations Using Neural Walk-on-Spheres. In *Forty-first International Conference on Machine Learning*.
- Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion curves: a vector representation for smooth-shaded images. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) (SIGGRAPH '08). Association for Computing Machinery, New York, NY, USA, Article 92, 8 pages. <https://doi.org/10.1145/1399504.1360691>
- Art Owen and Yi Zhou. 2000. Safe and Effective Importance Sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000), 135–143. <http://www.jstor.org/stable/2669533>
- Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. *Computer Graphics Forum (Proceedings of EGSR)* 41, 4 (July 2022). <https://doi.org/10.1145/10/jgzz>
- Alexander Rath, Pascal Grittmann, Sebastian Herholz, Petr Věvoda, Philipp Slusallek, and Jaroslav Křivánek. 2020. Variance-aware path guiding. *ACM Trans. Graph.* 39, 4, Article 151 (Aug. 2020), 12 pages. <https://doi.org/10.1145/3386569.3392441>
- 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. <https://doi.org/10.1145/3272127.3275030>
- Damien Rioux-Lavoie, Ryusuke Sugimoto, Tümay Özdemir, Naoharu H. Shimada, Christopher Batty, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2022. A Monte Carlo Method for Fluid Simulation. *ACM Transactions on Graphics* 41, 6 (Dec. 2022). <https://doi.org/10.1145/3550454.3555450>
- Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust fitting of parallax-aware mixtures for path guiding. *ACM Trans. Graph.* 39, 4, Article 147 (Aug. 2020), 15 pages. <https://doi.org/10.1145/3386569.3392421>
- Rohan Sawhney. 2021. *FCPW: Fastest Closest Points in the West*.
- Rohan Sawhney and Keenan Crane. 2020. Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-Based Methods on Volumetric Domains. *ACM Trans. Graph.* 39, 4 (2020).
- Rohan Sawhney and Bailey Miller. 2023. *Zombie: A Grid-Free Monte Carlo Solver for PDEs*.
- Rohan Sawhney and Bailey Miller. 2024. Monte Carlo Geometry Processing. In *SGP 2024 Graduate School Courses*.
- Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Trans. Graph.* 42, 4 (2023).
- Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-free Monte Carlo for PDEs with spatially varying coefficients. *ACM Trans. Graph.* 41, 4, Article 53 (July 2022), 17 pages. <https://doi.org/10.1145/3528223.3530134>
- Nikolai A. Simonov. 2008. Walk-on-Spheres Algorithm for Solving Boundary-Value Problems with Continuity Flux Conditions. <https://api.semanticscholar.org/CorpusID:117970575>
- Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2024a. Velocity-Based Monte Carlo Fluids. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 8, 11 pages. <https://doi.org/10.1145/3641519.3657405>

- Ryusuke Sugimoto, Terry Chen, Yiti Jiang, Christopher Batty, and Toshiya Hachisuka. 2023. A Practical Walk-on-Boundary Method for Boundary Value Problems. *ACM Trans. Graph.* 42, 4, Article 81 (jul 2023), 16 pages. <https://doi.org/10.1145/3592109>
- Ryusuke Sugimoto, Nathan King, Toshiya Hachisuka, and Christopher Batty. 2024b. Projected Walk on Spheres: A Monte Carlo Closest Point Method for Surface PDEs. In *ACM SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (*SIGGRAPH Asia '24*). Association for Computing Machinery, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3680528.3687599>
- Yusuke Tokuyoshi. 2025. *A Numerically Stable Implementation of the von Mises-Fisher Distribution on  $S^2$* . Technical Report. Advanced Micro Devices, Inc. [https://gpuopen.com/download/publications/A\\_Numerically\\_Stable\\_Implementation\\_of\\_the\\_von\\_Mises%E2%80%93Fisher\\_Distribution\\_on\\_S2.pdf](https://gpuopen.com/download/publications/A_Numerically_Stable_Implementation_of_the_von_Mises%E2%80%93Fisher_Distribution_on_S2.pdf)
- Eric Veach and Leonidas Guibas. 1995a. Bidirectional Estimators for Light Transport. In *Photorealistic Rendering Techniques*, Georgios Sakas, Stefan Müller, and Peter Shirley (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 145–167.
- Eric Veach and Leonidas J. Guibas. 1995b. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 419–428. <https://doi.org/10.1145/218380.218498>
- Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. 2014. On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph.* 33, 4, Article 101 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601203>
- Ekrem Fatih Yilmazer, Delio Vicini, and Wenzel Jakob. 2024. Solving Inverse PDE Problems using Monte Carlo Estimators. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 43 (Dec. 2024). <https://doi.org/10.1145/3687990>
- Z. Yu, L. Wu, Z. Zhou, and S. Zhao. 2024. A Differential Monte Carlo Solver For the Poisson Equation. In *ACM SIGGRAPH 2024 Conference Proceedings*.
- Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021). <https://doi.org/10.1145/3450626.3459807>
- Shilin Zhu, Zexiang Xu, Tiancheng Sun, Alexandr Kuznetsov, Mark Meyer, Henrik Wann Jensen, Hao Su, and Ravi Ramamoorthi. 2021. Hierarchical neural reconstruction for path guiding using hybrid path and photon samples. *ACM Trans. Graph.* 40, 4, Article 35 (July 2021), 16 pages. <https://doi.org/10.1145/3450626.3459810>

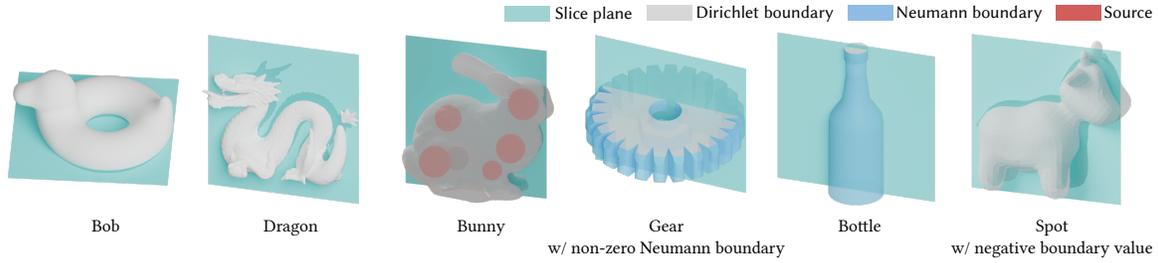


Fig. 7. Dataset for our 3D experiments. We adopt solving on slices [Sawhney and Crane 2020, Section 5.2] as the visualization method.

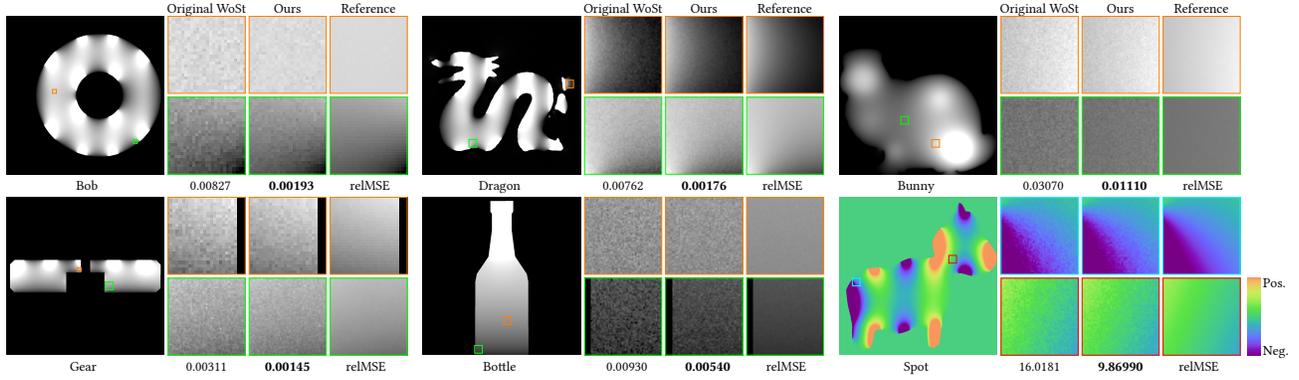


Fig. 8. Equal-sample qualitative results of our 3D experiments at 512 wpp. We follow Sawhney and Crane [2020, Fig. 6], using grayscale images to visualize the solution for positive-only results. Since WoSt typically exhibits variance as uniform salt-and-pepper noise [Sawhney et al. 2023, Section 4], we zoom in on representative regions for visualization. We report relMSE at 512 wpp for each problem below the corresponding images.

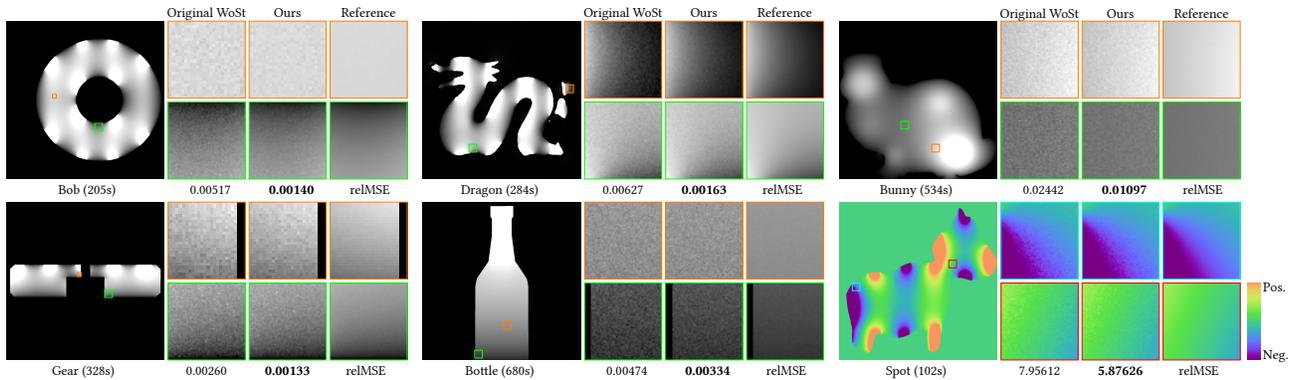


Fig. 9. Equal-time qualitative results of our 3D experiments. We report wall-time results for all 3D problems at their respective given runtime.

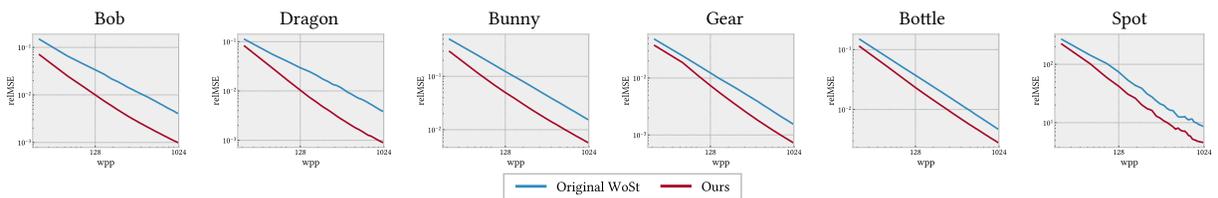


Fig. 10. relMSE plotted as a function of wpp of our 3D experiments. Under the condition of equal wpp, our method comprehensively surpasses the original WoSt in 3D problems. In the best scenario, it achieves more than a 4× reduction in variance, while in problems with relatively uniform solution distributions, it still delivers approximately a 2× improvement.

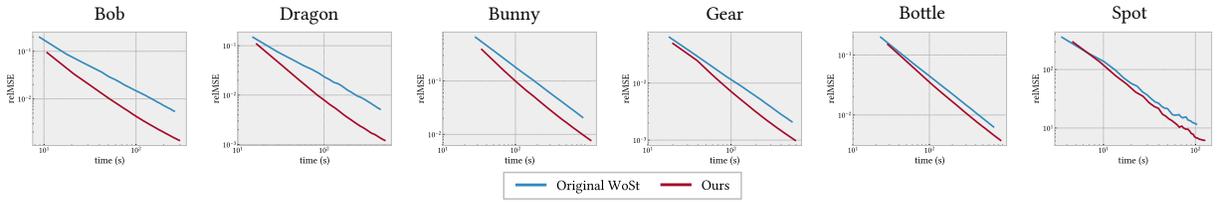


Fig. 11. relMSE plotted as a function of time of our 3D experiments. Despite the performance overhead, our method still maintains an advantage in equal-time comparisons across all 3D problems.

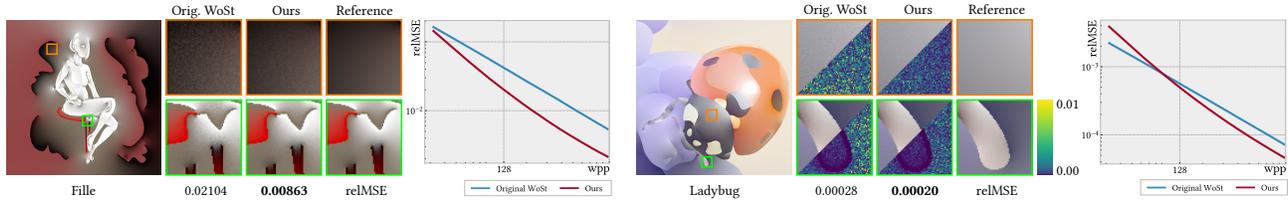


Fig. 12. Equal-sample qualitative and quantitative results of the 2D diffusion curve experiments. We present the rendering results of two examples at 256 wpp and report their relMSE. Due to the rapid convergence of *Ladybug*, the noise of both methods is nearly imperceptible. To facilitate comparison, we also present false-color maps of the per-pixel relative squared error.

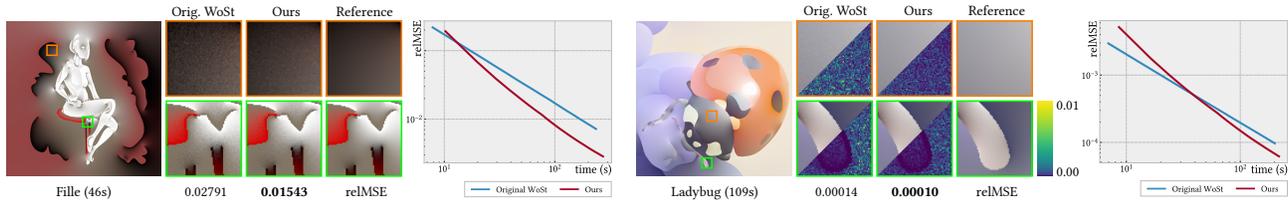


Fig. 13. Equal-time qualitative and quantitative results of the 2D diffusion curve experiments.

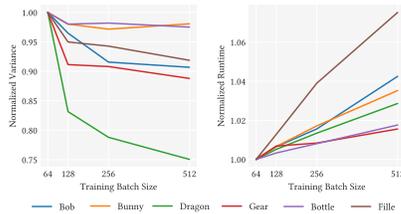


Fig. 14. The evaluation experiment results for training batch size. We normalize the results by the data with the batch size of 64.

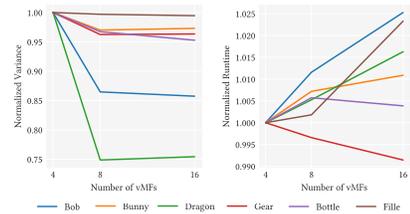


Fig. 15. The ablation experiment results for the number of vMFs. We normalize the results by the data with the number of 4.

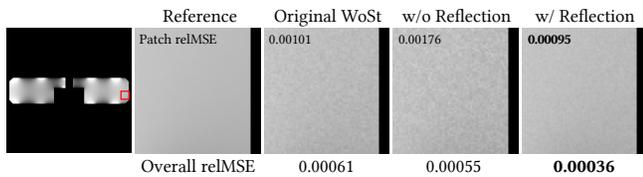


Fig. 16. Ablation experiment results for the sample reflection at Neumann boundaries. We show qualitative results of the patch near a Neumann boundary. We also report their patch relMSE and overall relMSE, *resp.*

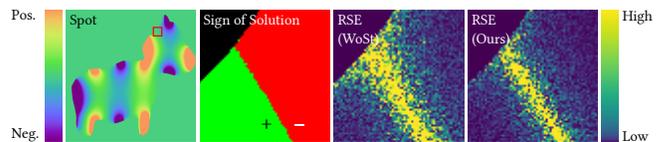


Fig. 17. Limitation: Our method effectively reduces shape variance, but like the baseline original WoSt, it cannot reduce sign variance.