Gradient Domain Reconstruction for Monte Carlo PDE Solvers

JIAQI WU, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China
XUEJUN HU, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China
SHUANG ZHAO, University of Illinois Urbana-Champaign, USA
KUN XU*, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, China

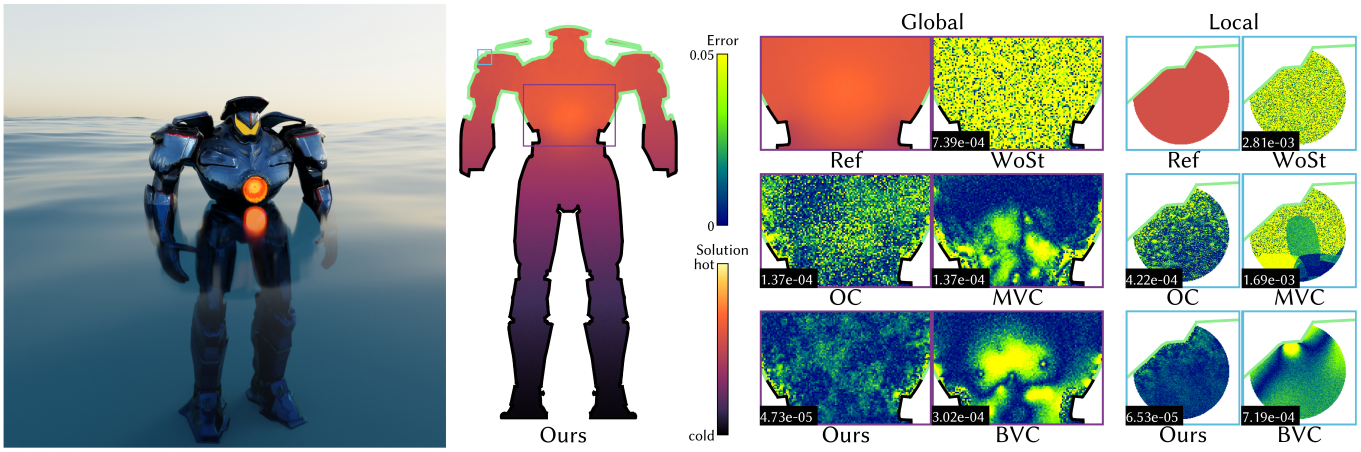

Fig. 1. We simulate heat transfer within a robot by solving the Laplace equation. The robot's lower body is immersed in water and its upper body is exposed to air, powered by an internal reactor. Under a simplified modeling assumption, we assign Dirichlet boundary conditions to the submerged lower-body surfaces to represent the water temperature, and Neumann boundary conditions to the exposed upper-body surfaces to model adiabatic heat transfer at the air interface. Internal sources account for heat generated by the reactor. We present temperature fields under two scenarios: a global slice solution, in which the solution is computed over a planar cross-section of the robot, and a local region-of-interest solution, restricted to a small area near the shoulder. All methods are constrained to a fixed 100-second time budget. Our method achieves lower error than other Monte Carlo PDE solvers in both scenarios.

Grid-free Monte Carlo methods are capable of solving Poisson equations on highly complex domains. However, existing methods operate solely in the primal domain and can converge slowly due to high variance. Inspired by gradient-domain rendering, we introduce a gradient-domain framework for Poisson problems. Specifically, we devise a new Monte Carlo estimator that directly targets differences of the solution between spatially varying query locations. Further, we adopt state-of-the-art reconstruction techniques originated in gradient-domain rendering to allow efficient reconstruction of the solutions without incurring additional bias. We demonstrate the effectiveness of our technique by comparing solutions obtained using our method and several state-of-the-art baselines.

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

*Corresponding author.

Authors' Contact Information: Jiaqi Wu, wujiaqi22@mails.tsinghua.edu.cn, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China; Xuejun Hu, jimmyhu1024@gmail.com, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China; Shuang Zhao, shzhao@illinois.edu, University of Illinois Urbana-Champaign, Champaign, USA; Kun Xu, xukun@tsinghua.edu.cn, BNRist, MOE Key Laboratory of Pervasive Computing, Department of CS&T, Tsinghua University, Beijing, China.

This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 1557-7368/2026/7-ART130
https://doi.org/10.1145/3811295

ACM Reference Format:
Jiaqi Wu, Xuejun Hu, Shuang Zhao, and Kun Xu. 2026. Gradient Domain Reconstruction for Monte Carlo PDE Solvers. *ACM Trans. Graph.* 45, 4, Article 130 (July 2026), 11 pages. https://doi.org/10.1145/3811295

1 Introduction

Poisson equations are a fundamental building block across science and engineering, modeling steady-state and quasi-static phenomena in a wide range of applications, including heat transfer, electrostatics and magnetostatics, incompressible flow, diffusion processes, and potential-based formulations in acoustics and elasticity. In practice, these problems are often posed on domains with intricate geometry—e.g., complex boundaries, thin features, heterogeneous materials, or domains defined implicitly or procedurally. Monte Carlo (MC) methods are attractive in such settings because they can operate without structured discretizations. On the other hand, the utility of MC solvers is frequently limited by variance: achieving acceptable accuracy can require prohibitively many samples.

In physics-based rendering, one approach to reduce Monte Carlo variance is to operate on the *gradient domain*. The key idea is to combine estimates of a primal quantity (i.e., pixel values) with estimates of its *differences* (e.g., spatial gradients or pairwise differences) and to reconstruct the final signal from these coupled measurements. Compared with conventional primal-domain estimators,

state-of-the-art gradient-domain techniques can provide unbiased estimates with markedly lower noise by carefully weighting the primal and difference estimates [Yan et al. 2025].

In parallel, *grid-free* Monte Carlo solvers such as Walk on Spheres (WoS) [Muller 1956] and Walk on Stars (WoSt) [Sawhney et al. 2023] enable efficient solutions of Poisson equations without meshing or volumetric discretization. By relying on path-tracing-like random walks, these methods avoid explicit grids and are particularly effective on highly detailed or procedurally defined domains where classical discretizations can be expensive or brittle. Recent work has shown that WoS can be made epsilon-free for 2D Dirichlet Laplace problems [Himmler and Günther 2025]. However, existing grid-free MC approaches operate fundamentally in the *primal domain*: They focus on estimating the solution $u(\mathbf{x})$ at individual query points $\mathbf{x}$, but do not directly estimate *differences* $u(\mathbf{x}) - u(\mathbf{y})$ across space—a key ingredient on which gradient-based rendering methods rely.

We bridge these lines of work by introducing a *gradient-domain Monte Carlo method* for Poisson equations. Instead of estimating only $u(\mathbf{x})$, we construct a Monte Carlo estimator for *differences* of Poisson solutions between spatially varying query locations, while retaining the grid-free advantages of WoS/WoSt-style random walks. We then reconstruct the Poisson solution using a modern gradient-domain reconstruction strategy adapted from gradient-domain rendering. Unlike rendering, however, our reconstruction is not restricted to a dense 2D image lattice, and can instead be defined on arbitrary sets of evaluation points, such as a slice, a local region of interest, or samples on a mesh surface. The resulting pipeline does not incur additional bias beyond that already present in the underlying primal estimator, yet achieves faster convergence in practice than comparable primal-domain methods, producing lower-variance solutions for a fixed sampling budget.

Concretely, our main contributions are:

- We introduce an efficient Monte Carlo method for estimating differences of Poisson solutions at varying query locations, enabling gradient-domain treatment within a grid-free stochastic PDE solver.
- We adapt a state-of-the-art reconstruction methodology from gradient-domain rendering to recover Poisson solutions from jointly estimated primal and difference information in a computationally efficient fashion, without introducing additional bias.

We demonstrate the effectiveness of our technique by comparing solutions obtained using our method and several state-of-the-art baselines.

2 Related Work

Monte Carlo PDE Solvers. Monte Carlo methods such as Walk on Spheres (WoS) provide grid-free solvers for elliptic PDEs and avoid the meshing challenges associated with traditional discretization-based approaches [Muller 1956]. Due to its simplicity and robustness in handling complex geometries, WoS has become increasingly popular in computer graphics. Recent work has extended WoS-type methods to more general equations and boundary conditions, including Neumann and Robin conditions through generalized random-walk formulations [Miller et al. 2024b; Sawhney et al. 2023]. Other work has developed differential and inverse variants of grid-free Monte Carlo PDE solvers, including Differential Walk on Spheres, inverse PDE estimation with grid-free Monte Carlo estimators, and a differential Monte Carlo solver for the Poisson equation [Miller et al. 2024a; Yilmazer et al. 2024; Yu et al. 2024]. More recently, robust derivative estimation under WoSt has also been studied, including improved treatment of pure Neumann settings [Yu et al. 2025]. These advances have broadened the applicability of Monte Carlo PDE solvers to a wide range of forward and inverse problems.

Variance Reduction for Monte Carlo PDE Solvers. Despite their flexibility, Monte Carlo PDE solvers such as WoS produce inherently noisy estimates, motivating the development of variance reduction techniques. Boundary Value Caching (BVC) reduces variance by caching solution estimates at selected boundary points and reusing them across the domain, though its basic formulation can suffer from singularity-related local artifacts near the boundary; the original paper also discusses a bias-corrected clamping strategy to mitigate this issue [Miller et al. 2023]. Mean Value Caching (MVC) exploits the volume mean-value property of harmonic functions to reuse walks within local neighborhoods; however, its effectiveness degrades near boundaries where few cache samples are available, and often leads to correlation artifacts [Bakbouk and Peers 2023].

Several alternative approaches reduce redundant sampling by correlating nearby evaluations. Czekanski et al. [2025] enables explicit communication between neighboring query points to reuse information and reduce variance, and off-centered WoS-type solvers pursue a similar goal by reusing random-walk information across overlapping local neighborhoods/balls [Bao et al. 2025; Czekanski et al. 2025]. These strategies are most effective when queries are sufficiently dense, and can be less beneficial when evaluations are sparse or confined to a small region of interest [Czekanski et al. 2025]. Complementary to neighborhood-based reuse, bidirectional WoS reformulates the estimator in a reverse direction by launching walks from source locations and estimating Green's-function contributions at sensor points, which is particularly beneficial for Poisson problems with sparse sources [Qi et al. 2022]. Harmonic Caching for Walk on Spheres exploits the harmonic structure more directly by building local harmonic representations from random walks and reusing them across space, yielding strong variance reduction even near boundaries while avoiding singular artifacts of point-based caching [Zhou et al. 2025]. In a complementary learning-based direction, neural control variates and neural caches amortize computation using inexpensive learned surrogates to reduce variance and/or reuse solution information across space [Li et al. 2023, 2024]. Overall, existing methods mainly reduce variance through local reuse or estimator reformulation. Our method instead follows a different route based on correlated difference estimation and unbiased reconstruction.

Gradient-Domain Reconstruction. In gradient-domain rendering, and in particular gradient-domain path tracing (GDPT) [Kettunen et al. 2015; Lehtinen et al. 2013], we estimate not only pixel values

but also *image-space finite differences* between neighboring pixels using *correlated sampling* (e.g., shift mapping). These gradient estimates encode first-order structural information and often exhibit substantially reduced variance compared to direct color estimates, but they must ultimately be integrated into a globally consistent image.

Gradient-domain reconstruction aims to recover a scalar field from noisy measurements of its finite differences by assembling these measurements into the right-hand side of a linear operator and solving a corresponding linear system. Classical approaches, such as screened Poisson reconstruction [Pérez et al. 2003], minimize a quadratic functional to balance fidelity to finite difference estimates and direct samples. Image-space control variates further relate gradient-domain reconstruction to variance reduction by exploiting spatial coherence between neighboring pixels within an iterative reconstruction scheme [Rousselle et al. 2016]. Recent work on generalized unbiased reconstruction [Yan et al. 2025] formalizes this process by constructing a general linear sum of finite differences estimates and direct samples and forcing unbiased reconstruction. This estimator-level perspective extends beyond rendering and makes the framework applicable to general PDE reconstruction problems.

Other methods improve reconstruction quality through learned or heuristic techniques [Back et al. 2020, 2023; Guo et al. 2019; Kettunen et al. 2019; Yan et al. 2024], often trading unbiasedness for perceptual quality. Adaptive sampling strategies have also been explored in gradient-domain rendering to allocate samples to difficult regions and improve final reconstruction quality. [Liang et al. 2024] In contrast, the generalized framework preserves statistical correctness while remaining robust to the choice of sampling strategy, motivating its adoption in our Monte Carlo PDE setting.

3 Our Method

We aim at solving Poisson equations

$$\Delta u(x) = -f(x), \text{ in } \Omega, \tag{1}$$

with mixed Dirichlet and Neumann boundary conditions:

$$u(x) = g(x) \text{ on } \partial\Omega_{\mathrm{D}}, \text{ and } \frac{\partial u(x)}{\partial \mathbf{n}_x} = h(x) \text{ on } \partial\Omega_{\mathrm{N}}, \tag{2}$$

where Δ denotes the Laplace operator, $\mathbf{n}_x$ denotes the outward normal on boundary point x, and Ω is the domain with the boundary $\partial\Omega$ (satisfying $\partial\Omega_{\mathrm{D}} \cup \partial\Omega_{\mathrm{N}} = \partial\Omega$ and $\partial\Omega_{\mathrm{D}} \cap \partial\Omega_{\mathrm{N}} = \emptyset$). Further, instead of solving Poisson equations over the entire domain Ω, we focus on estimating the solution $u_i \triangleq u(x_i)$ at a set of N evaluation points $x_1, x_2, \ldots, x_N \in \Omega$.

In this paper, we leverage gradient-domain reconstruction to solve this problem efficiently. To this end, we assume a neighbor relation $\mathcal{N} \subset \{1, 2, \ldots, N\}^2$ that can be defined naturally on regular grids (via fixed-size neighborhoods) and on meshes (via edge connectivity), or more generally via k-nearest neighbors (kNN) when explicit adjacency is unavailable. Different from traditional gradient-domain rendering methods which typically rely on fixed 4-connected or 8-connected neighborhoods, we allow more flexible connectivity patterns that better suit the geometry and characteristics of the PDE domain.

For each neighboring pair $(i, j) \in \mathcal{N}$, we define the value difference d_{ij} as:

$$d_{ij} \triangleq u_j - u_i. \tag{3}$$

Method overview. Our method consists of three major steps:

(1) **Value estimation.** First, we use existing Monte Carlo PDE solvers such as Walk-on-Sphere (WoS) to obtain initial value estimates $\hat{u}_i$ at all evaluation points x_i (Section 3.1). A key modification involves caching the sampled walk sequences for reuse in subsequent steps.
(2) **Difference estimation.** Next, we compute estimates of value differences $\hat{d}_{ij}$ between neighboring points x_i and x_j using *shared walk sequences* (Section 3.2). We reuse the walk sequences generated in the value estimation step to reduce computational overhead.
(3) **Gradient-domain reconstruction.** Lastly, we perform a gradient-domain reconstruction that integrates the initially estimated values and their differences (Section 3.3). This process yields refined value estimates with substantially reduced variance.

In the following, we present each of these steps in more details.

3.1 Value Estimation

The Walk-on-Spheres (WoS) estimator provides a Monte Carlo method for evaluating the solution at all evaluation points. For a point x within an open ball $B \subseteq \Omega$, the solution $u(x)$ satisfies the integral equation:

$$u(x) = \int_{\partial B} u(z)\, P_B(x, z)\, \mathrm{d}z + \int_B f(y)\, G_B(x, y)\, \mathrm{d}y, \tag{4}$$

where P_B is the *Poisson kernel* and G_B is the *Green's function* for the ball B. The Monte Carlo estimator for $u(x)$ can be expressed using a single-sample formulation:

$$\hat{u}(x) = \frac{\hat{u}(z)\, P_B(x, z)}{p(z)} + \frac{f(y)\, G_B(x, y)}{q(y)}, \tag{5}$$

where a boundary point $z \in \partial B$ is sampled on the sphere from distribution $p(z) \propto P_B(x, z)$, and an interior point $y \in B$ is sampled within the ball from distribution $q(y) \propto G_B(x, y)$. When the ball B is centered at x, the Poisson kernel P_B becomes a constant function and the distribution $p(z)$ simplifies to a uniform distribution. Note that when the source function f is identically zero, the second term in the estimator vanishes entirely and sampling interior points y becomes unnecessary.

The Walk-on-Spheres (WoS) algorithm is recursive. Starting from a point x, we construct the largest ball B centered at x that remains entirely within the domain, then sample a boundary point $z^{(1)}$ uniformly on ∂B and an interior point $y^{(1)}$ within B according to the distribution $q(y)$. We then repeat the same procedure from $z^{(1)}$: construct the largest ball centered at $z^{(1)}$, sample a new boundary point $z^{(2)}$ and a new interior point $y^{(2)}$, and continue recursively. This generates a walk sequence of boundary points $(z^{(1)}, z^{(2)}, \ldots)$ and a corresponding sequence of interior points $(y^{(1)}, y^{(2)}, \ldots)$, until the sampled boundary point falls within a small tolerance distance of the Dirichlet boundary, at which point the boundary value is returned and used to compute the solution estimate.

In practice, to reduce variance in the estimation, for each evaluation point x_i, we sample M independent walk sequences and compute their average as our initial value estimate $u(x_i)$.

Caching sampled walk sequences. To minimize computational overhead in subsequent difference estimation steps, we implement a caching strategy that preserves essential information from the Walk-on-Spheres (WoS) computations. The primary motivation is to enable efficient reuse of already-generated random walk data when estimating value differences between neighboring points.

For every evaluation point x_i, we have sampled M independent walk sequences. For each walk sequence, rather than storing the complete walk, we cache only the essential information from the first walk step, including:

(1) the initial maximum inscribed ball B_i centered at x_i;
(2) the first boundary point sampled on the initial ball $z_i^{(1)}$;
(3) the solution estimate at the first boundary point, denoted as $\hat{u}(z_i^{(1)})$;
(4) the first interior point sampled within the initial ball $y_i^{(1)}$;
(5) the function estimate at the first interior point $f(y_i^{(1)})/q(y_i^{(1)})$.

Their subsequent usage on difference estimation will be detailed in Section 3.2. Figure 2 provides an illustration of the cached information.

Handling mixed boundary conditions. For problems with Neumann boundaries, we adopt the Walk-on-Star (WoSt) algorithm after the first step. In the first step, we still use WoS to construct the maximum inscribed ball to ensure simplicity and cache compatibility. Subsequent steps leverage star-shaped domains for accelerated convergence, as later computations do not interfere with the cached data. This approach aligns with methods like the one by Bao et al. [2025] and Zhou et al. [2025], which also use inscribed balls in the first step.

3.2 Difference Estimation

In gradient-domain rendering, shift mapping techniques effectively create correlated path samples for low-variance difference estimation. However, shift mappings are specifically designed for light transport simulation and are not applicable to our WoS context.

Recall the WoS integral formulation and its Monte Carlo estimator in Equation 4 and Equation 5. While in practice the ball B is typically centered at the evaluation point x, the formulation remains valid even when the center of B differs from x [Bao et al. 2025]. In such cases, however, the Poisson kernel is no longer a constant function, and the Green's function will take a more complicated form.

This motivates us to estimate value differences between points using *shared walk sequences*. For two points x_i, x_j contained within a common ball $B \subseteq \Omega$, we leverage correlated sampling to construct an efficient difference estimator. The estimation begins by sampling a boundary point z on ∂B and an interior point $y \in B$ from given distributions $p(z), q(y)$, respectively. The difference estimator is then derived by applying the Monte Carlo formulation

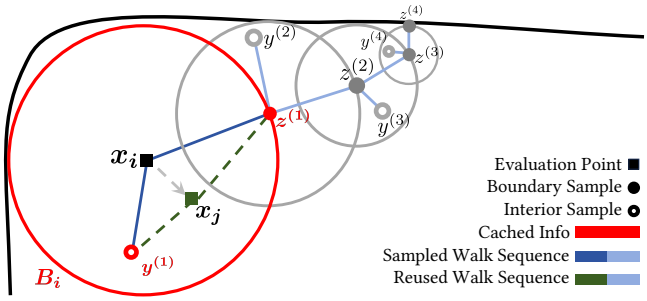

Fig. 2. Illustration of *shared walk sequences*. During the *value estimation* stage, a walk sequence is sampled using WoS/WoSt. This sequence is then reused to form a pair of *shared walk sequences*, which yields a difference estimate according to Equation 6. The cached information used for difference estimation is highlighted in red.

in Equation 5 to both points and computing their difference:

$$\hat{d}_{ij} = \frac{\hat{u}(z)\big(P_B(x_j, z) - P_B(x_i, z)\big)}{p(z)} + \frac{f(y)\big(G_B(x_j, y) - G_B(x_i, y)\big)}{q(y)}. \tag{6}$$

In this equation, $\hat{u}(z)$ is computed recursively using the standard WoS/WoSt method, continuing the random walk sequence from the sampled point $z \in B$ as in Equation 5, using the same distribution functions (p and q). This difference estimator essentially enables both x_i and x_j to share the same walk sequence—the same initial point z and all subsequent walk points in the recursive evaluation. This shared sampling strategy ensures strong correlation between the estimates, effectively canceling out common noises and achieving significant variance reduction compared to independent evaluations. We provide an illustration of estimating difference with shared walk sequences in Figure 2.

A straightforward method for estimating $\hat{d}_{ij}$ using Equation 6 involves finding a common ball B_{ij} that contains both points x_i and x_j, then sampling new boundary points and interior points to compute the difference. While feasible, this approach requires additional computational effort to identify a suitable ball for each pair of points. However, recall that during value estimation, we already construct the maximum inscribed ball B_i centered at each point x_i. For a neighboring point x_j, it is highly probable that x_j also lies within B_i. This observation enables us to directly reuse the cached ball B_i and its associated sampled walk sequences for difference estimation.

Difference estimation using cached walk sequences. For a pair of neighboring evaluation points x_i and x_j (where $(i, j) \in \mathcal{N}$), assume that x_j lies inside the ball B_i (the maximum ball centered at x_i). Recall that for each point x_i, we previously generated M independent walk sequences. Each of these sequences can serve as a shared walk sequence for estimating the difference using Equation 6. The final estimate $\hat{d}_{ij}$ is obtained by averaging the results from all M sequences.

It is important to note that while our difference estimator conceptually shares the entire walk sequence, the computation in Equation 6 only requires the information stored in the first walk step,

including: the ball B_i, the first sampled boundary point $z_i^{(1)}$ and the interior point $y_i^{(1)}$ from each sequence, and the corresponding solution estimates $\hat{u}(z_i^{(1)})$ and $f(y_i^{(1)})/q(y_i^{(1)})$. This minimal yet sufficient set of cached data explains why we store only these elements, avoiding the overhead of preserving the complete walk sequences.

Symmetry enforcement. The computation described above does not inherently guarantee symmetry for two reasons: first, the condition $x_j \in B_i$ does not necessarily imply $x_i \in B_j$; second, the resulting estimates do not automatically satisfy $\hat{d}_{ij} = -\hat{d}_{ji}$. To ensure robustness, we enforce symmetry as follows:

- If $x_j \in B_i$ and $x_i \in B_j$, we obtain $\hat{d}_{ij}$ and $\hat{d}_{ji}$ using B_i and B_j, respectively, and set the final estimates as $\hat{d}_{ij} = \frac{1}{2}(\hat{d}_{ij} - \hat{d}_{ji})$ and $\hat{d}_{ji} = -\hat{d}_{ij}$.
- If only one point lies inside the other's ball (e.g., $x_j \in B_i$ but $x_i \notin B_j$), we use the valid estimate (e.g., $\hat{d}_{ij}$) and assign $\hat{d}_{ji} = -\hat{d}_{ij}$.
- If neither point lies within the other's ball (i.e., $x_j \notin B_i$ and $x_i \notin B_j$), which rarely occurs in practice, we discard the edge (i, j) from the neighboring relation $\mathcal{N}$.

This approach ensures that the estimated differences form a consistent, antisymmetric set of values suitable for gradient-domain reconstruction.

3.3 Generalized Unbiased Reconstruction

After obtaining the value estimates $\hat{u}_i$ and the value-difference estimates $\hat{d}_{ij}$, we apply gradient-domain reconstruction to compute the final solution. For high-quality results, we adopt the state-of-the-art generalized unbiased reconstruction method by Yan et al. [2025].

This technique reconstructs the value at each evaluation point x_p as a linear combination of all available estimates:

$$\tilde{u}_p = \sum_{i=1}^{N} \alpha_i^{(p)} \hat{u}_i + \sum_{(i,j)\in\mathcal{N}} \beta_{ij}^{(p)} \hat{d}_{ij}, \quad 1 \le p \le N, \tag{7}$$

where $\alpha_i^{(p)}$ and $\beta_{ij}^{(p)}$ are reconstruction weights.

To ensure $\tilde{u}_p$ being an unbiased estimator of the true solution u_p, the weights must satisfy the following constraint for every p and i:

$$\alpha_i^{(p)} + \sum_{\substack{k<i \\ (k,i)\in\mathcal{N}}} \beta_{ki}^{(p)} - \sum_{\substack{j>i \\ (i,j)\in\mathcal{N}}} \beta_{ij}^{(p)} = \begin{cases} 1 & \text{if } i = p \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

Among all weight sets satisfying this unbiasedness condition, the optimal ones minimize the variance of $\tilde{u}_p$. Assuming the input estimates $\hat{u}_i$ and $\hat{d}_{ij}$ are mutually uncorrelated, this variance equals

$$\mathrm{Var}\left[\tilde{u}_p\right] = \sum_{i=1}^{N} \left(\alpha_i^{(p)}\right)^2 \mathrm{Var}\left[\hat{u}_i\right] + \sum_{(i,j)\in\mathcal{N}} \left(\beta_{ij}^{(p)}\right)^2 \mathrm{Var}\left[\hat{d}_{ij}\right]. \tag{9}$$

Minimizing Equation 9 subject to the unbiasedness constraint (Equation 8) forms a quadratic optimization problem under linear constraints. This can be reformulated as solving a sparse linear system. Rather than solving explicitly for all reconstruction weights, this optimization can be reformulated as a sparse linear system directly in the reconstructed values. In practice, individual weights $\alpha_i^{(p)}$ and $\beta_{ij}^{(p)}$ do not need to be computed explicitly. Instead, the reconstructed values $\tilde{I} \triangleq (\tilde{u}_1, \dots, \tilde{u}_N)$ are obtained directly by solving:

$$\left(SF^{-1}S^\top\right)\tilde{I} = SF^{-1}c, \tag{10}$$

where S encodes the unbiasedness constraints, F is a diagonal matrix containing (filtered) variance estimates, and c is the concatenated vector of all input estimates. We solve this system efficiently using the conjugate gradient method, iterating until the residual error falls below a predefined threshold. For complete mathematical details, we refer readers to the original paper [Yan et al. 2025].

Variance estimation and filtering. Since the true variances in the objective function (Equation 9) are unknown, we approximate them using sample variances $\widehat{\mathrm{Var}}[\cdot]$. To reduce noise, we apply spatial filtering to the sample variances. However, unlike image-based gradient-domain rendering, our setting lacks an auxiliary guiding buffer, so our filtering strategy differs slightly.

For each value estimate $\hat{u}_i$, we compute the filtered sample variance by averaging sample variances within a local window $\mathcal{W}(i)$:

$$\widehat{\mathrm{Var}}_{\text{filtered}}\left[\hat{u}_i\right] = \frac{1}{k_i} \sum_{j\in\mathcal{W}(i)} \widehat{\mathrm{Var}}\left[\hat{u}_j\right] G\left(\|x_j - x_i\|, \sigma_s\right), \tag{11}$$

where $G(t, \sigma_s) = \exp(-t^2/(2\sigma_s^2))$ is a Gaussian kernel, and k_i is the normalization factor. The choice of $\mathcal{W}(i)$ depends on the structure of the evaluation points. On regular 2D grids, we use a centered 7×7 window as the local filter support. In other cases, we use kNN to construct a similar local neighborhood instead.

For the difference estimates $\hat{d}_{ij}$, our filtering strategy depends on whether the evaluation points form a regular grid. When dealing with irregularly distributed points, we directly use the sample variance without spatial filtering. When evaluation points are arranged on a regular grid, we use directional filtering. For two neighboring points x_i and x_j, let $o = x_j - x_i$ be their offset vector. The variance of the difference estimate is filtered as:

$$\widehat{\mathrm{Var}}_{\text{filtered}}\left[\hat{d}_{ij}\right] = \frac{1}{k_{ij}} \sum_{i'\in\mathcal{W}(i)} \widehat{\mathrm{Var}}\left[\hat{d}_{i',j'}\right] G\left(\|x_{i'} - x_i\|, \sigma_s\right), \tag{12}$$

where the summation is taken over all points i' within the same local window $\mathcal{W}(i)$ centered at i, and j' is the corresponding neighbor of i' satisfying $x_{j'} - x_{i'} = o$. k_{ij} is the normalization factor.

Single-buffer and dual-buffer variants. The unbiasedness of the generalized reconstruction method relies on the key assumption that the reconstruction weights are independent of the underlying Monte Carlo estimates. In other words, the weights should be chosen without looking at the particular Monte Carlo noise realization of the estimates they are later applied to. To enforce this assumption in practice, the original work [Yan et al. 2025] adopts a *dual-buffer* strategy, which splits samples into two disjoint sets and applies cross-fitting so that the weights are computed from samples independent of those being reconstructed. This guarantees unbiasedness, but at the cost of increased reconstruction noise

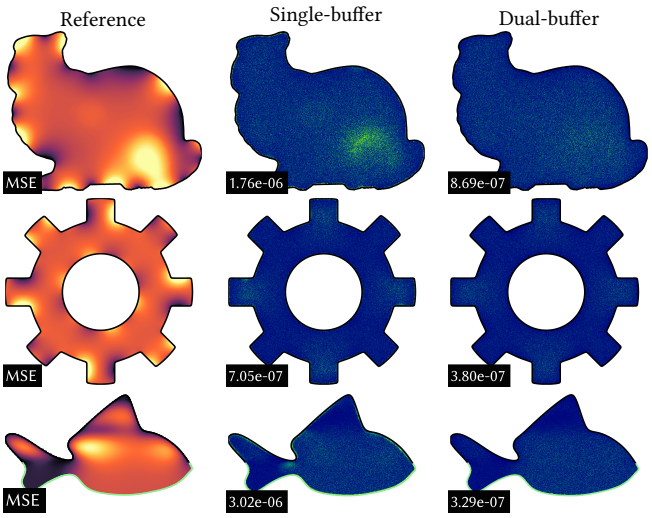

Fig. 3. Bias comparison between the single-buffer variant (biased) and the dual-buffer variant (unbiased). We average 2048 independent runs for each variant, using 16 walks per point in each run, to suppress stochastic noise. The second and third columns show the corresponding error maps. With noise largely suppressed, the bias of the single-buffer variant becomes visible, but remains comparatively small.

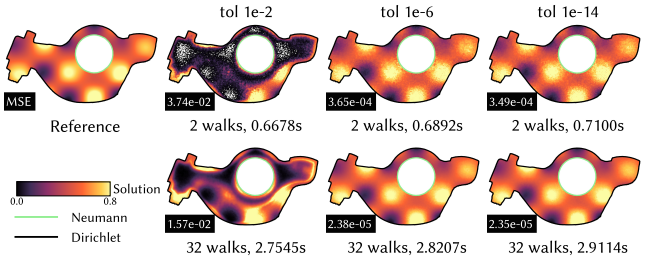

Fig. 4. Comparison of solutions obtained with different tolerance values for the *conjugate gradient* solver. From left to right, results computed with tolerances 10^{-2}, 10^{-6}, and 10^{-14} are shown. Smaller CG tolerances (i.e., more iterations) yield more accurate solutions at a modest increase in computational cost.

due to reduced effective sample count. The authors also describe a *single-buffer* variant that uses all samples to compute reconstruction weights, thereby violating the independence assumption and introducing bias. However, since this variant benefits from a larger effective sample count and can exploit sample correlations to better suppress outliers, it often achieves lower variance. Empirically, the introduced bias is often very small. We adopt the single-buffer variant as a more practical choice, while strict unbiasedness can still be enforced by using the dual-buffer strategy.

4 Experiments

We implement our method on CPU using the Zombie library [Sawhney and Miller 2023]. Our approach integrates seamlessly into the existing WoSt solver framework and requires only minimal modifications. All computations are performed in single precision. The final reconstruction step—solving the linear system in Equation 10—is performed on CPU using Eigen [Guennebaud et al. 2010].

For ablation studies and comparisons, we construct several model problem instances on various 3D geometries by specifying the domain functions f, g, and h (as defined in Equation 2). Following the slice-based evaluation strategy of Sawhney and Crane [2020], we uniformly sample evaluation points on a two-dimensional slice, using 512×512 points unless stated otherwise. Reference results are generated using 65,536 random walks. All timings are measured on an Intel Core i9-13900KF CPU.

4.1 Ablation Studies

Single-buffer and dual-buffer variants. As discussed earlier, the generalized reconstruction method supports both a single-buffer variant, which is more efficient in practice, and a dual-buffer variant, which strictly preserves unbiasedness. To empirically evaluate the trade-off between these two variants, we conduct the following experiment. Here, a *walk* refers to a single Monte Carlo sample used by the underlying solver at an evaluation point, whereas a *run* denotes one full execution of the entire reconstruction pipeline. We repeatedly run each variant using 16 walks per evaluation point and average the reconstructed results over multiple independent runs. As the number of repetitions increases, stochastic noise in the error progressively diminishes, and the remaining error converges toward the bias component. Figure 3 visualizes the error maps of the averaged reconstructions after 2048 runs. The single-buffer variant exhibits a slightly larger error due to its inherent bias; however, the magnitude of this bias remains small compared to the noise level typically observed in single-run settings. In contrast, the dual-buffer variant avoids bias but incurs higher variance and additional computational cost. Based on this trade-off, we adopt the single-buffer variant and use it as the default configuration in all subsequent experiments.

Tolerance parameter of the CG solver. The *tolerance* parameter controls the stopping criterion of the iterative *conjugate gradient* (CG) solver. A smaller tolerance yields higher solution accuracy at the cost of additional iterations. Figure 4 illustrates this trade-off, showing that tighter CG tolerances (i.e., more iterations) lead to more accurate solutions with only a modest increase in computational cost. We set the tolerance to 10^{-14} in all experiments.

Neighborhood size. Recall that on regular 2D grids, we use fixed-size local neighborhoods. In Figure 5, we have tested local neighborhoods with different window sizes. The results show that the 3×3 neighborhood consistently achieves the best accuracy while also incurring the lowest computational cost. Hence, we always use this setting for evaluations on regular 2D grids.

Generalized unbiased reconstruction vs Poisson reconstruction. In Figure 6, we evaluate our choice of gradient-domain reconstruction by comparing our adopted method, i.e., the generalized unbiased approach [Yan et al. 2025], against the classical Poisson (L_2) reconstruction. Both methods are evaluated on a 2D scene with complex boundary conditions and source term. Our results demonstrate that

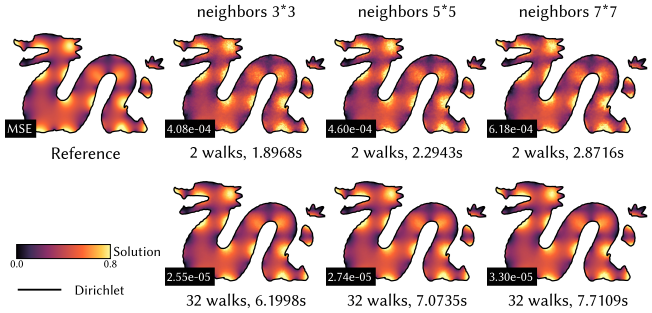

Fig. 5. Comparison of results obtained with different local neighborhood sizes. From left to right, results using 3×3, 5×5, and 7×7 neighborhoods are shown. Smaller neighborhood sizes generally produce more accurate solutions while incurring lower computational cost.

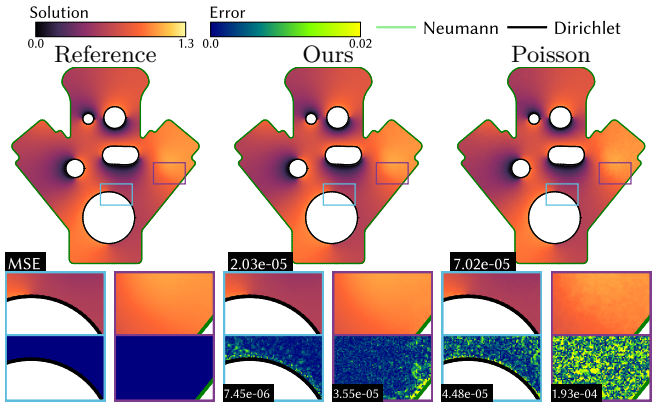

Fig. 6. Comparison between the adopted gradient-domain reconstruction method and classical Poisson reconstruction on a 2D scene. The top row shows the globally reconstructed fields, while the bottom rows present zoomed-in views of two representative high-gradient regions in terms of solution and error, respectively. Classical Poisson reconstruction exhibits increased error near boundaries and in source regions where gradient variations are steep.

the generalized unbiased reconstruction method yields lower error and superior visual quality.

Design trade-offs. Our design balances variance reduction, computational cost, and statistical robustness. By default, we use the single-buffer variant, which shows negligible empirical bias and avoids the variance increase and higher reconstruction cost of the strictly unbiased dual-buffer variant. We further adopt tight CG tolerances, small local neighborhoods, and generalized unbiased reconstruction for improved accuracy at modest cost. For difference estimation, we reuse cached shared walks instead of more aggressive pair-specific couplings, keeping the method lightweight while leaving stronger correlations as a possible higher-cost extension.

4.2 Runtime breakdown

We report the runtime of WoS/WoSt value estimation, difference estimation with smoothing, and unbiased reconstruction separately.

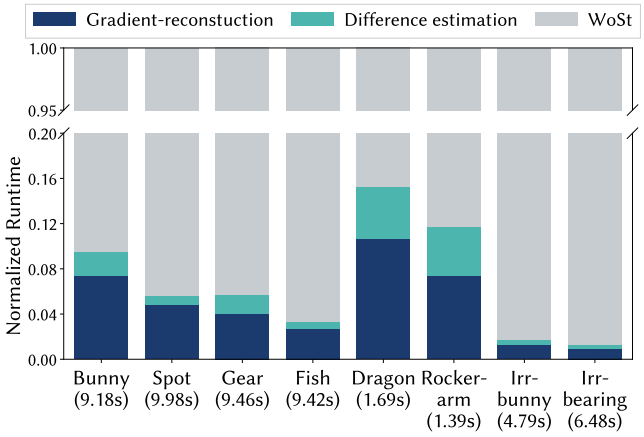

Fig. 7. Normalized runtime breakdown of the three-stage pipeline: WoS/WoSt value estimation, difference estimation with smoothing, and unbiased reconstruction. Across all scenes, the post-processing stages remain low-cost. Their relative contribution is slightly larger when the number of walks M is small, but becomes negligible at higher M, where runtime growth is dominated by WoS/WoSt value estimation.

As shown in Figure 7, the latter two post-processing steps consistently contribute only a small fraction of the total runtime across all examples. Their relative share is slightly more noticeable when the number of walks M is small and the total runtime is low (e.g., Dragon and Rocker-arm). As M increases, the runtime growth is dominated by value estimation, while the post-processing cost remains low and grows much more slowly. Therefore, at higher M, the overhead of post-processing is effectively negligible.

4.3 Comparisons

Compared methods. We compare our method with *Walk-on-Stars (WoSt)* [Sawhney et al. 2023] and several existing variance reduction techniques, including *Boundary Value Caching (BVC)* [Miller et al. 2023], *Mean-Value Caching (MVC)* [Bakbouk and Peers 2023], *Harmonic Caching* (HC) [Zhou et al. 2025], and *Off-Centered WoS Solver (OC)* [Bao et al. 2025]. For *Mean-Value Caching*, we use uniform sampling to generate cache points and fall back to WoSt when no cache point lies within the ball of a given evaluation point. For *Off-Centered WoS Solver*, we adopt the grid-based neighbor selection strategy when the evaluation points form a dense regular grid; otherwise, we construct a KD-tree to accelerate neighbor queries. Unless otherwise specified, all baseline methods are evaluated using their default parameter settings. All comparisons are conducted under an equal-time budget.

Challenging examples. In Figure 1 and Figure 8 we present results on several challenging scenes. Figure 1 and the third scene in Figure 8 focus on cases where evaluation points are highly localized in part of the geometry; the first scene in Figure 8 is dominated by source contributions; and the second scene in Figure 8 considers a gear geometry with a highly nontrivial boundary and internal cavities. These scenes are particularly challenging for grid-free Monte Carlo solvers and existing variance reduction techniques,

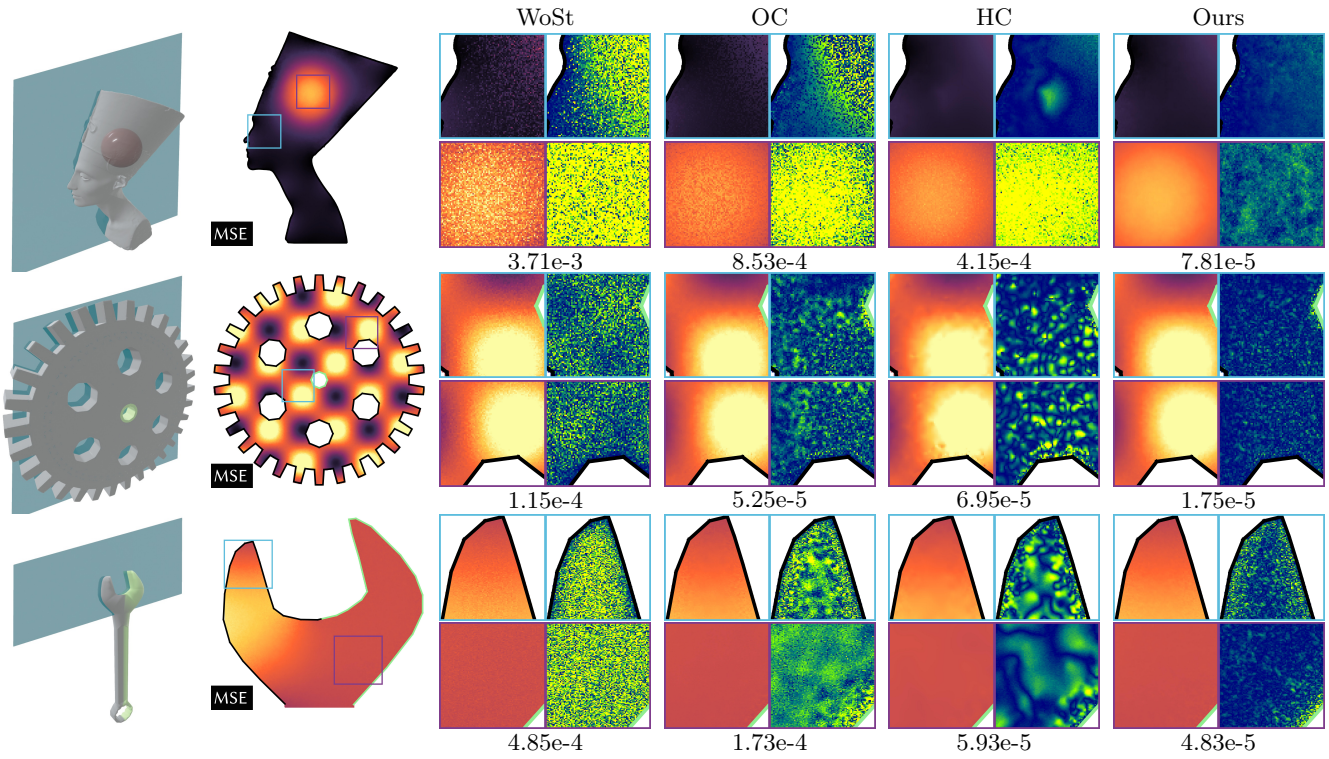

Fig. 8. Comparison on three challenging examples with Walk-on-Stars(WoSt) [Sawhney et al. 2023], *Off-Centered WoS Solver* (Off-center) [Bao et al. 2025] and *Harmonic Caching* (HC) [Zhou et al. 2025]. The first scene evaluates a source-dominated Nefertiti model; the second scene considers a perforated gear with complex boundary conditions; and the third scene focuses on solving the head region of a wrench. Our method achieves the lowest error across all three scenes.

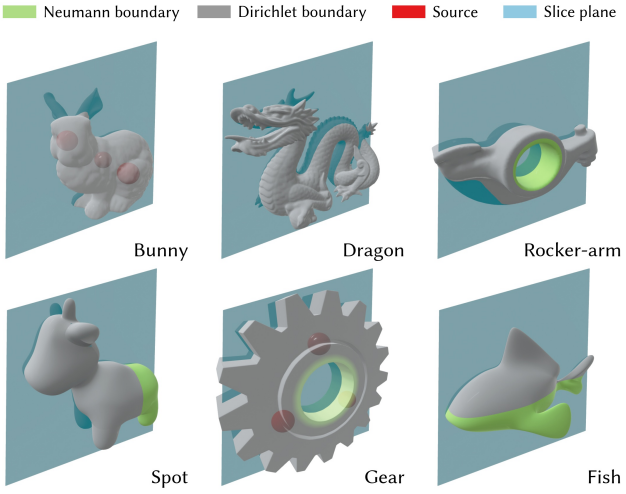

Fig. 9. Illustration of a set of constructed model problems.

since strong source terms, complex geometries with narrow features, and localized queries reduce the effectiveness of random-walk reuse and caching. As shown in the results, both *Harmonic Caching* (HC) and *Off-Centered WoS Solver* (OC) exhibit increased variance in source-driven regions and degraded performance near boundaries, where the size of admissible balls is inherently limited. Our method remains robust and achieves lower error across all scenes by computing additional low-variance difference estimates and effectively reusing information during the reconstruction process.

More results. For a more comprehensive evaluation, we design an additional set of representative model problems, illustrated schematically in Figure 9. An equal-time comparison is shown in Figure 10. Under this setting, our method achieves lower error than the compared techniques. Compared to *Mean-Value Caching*, our approach exhibits substantially reduced noise near boundaries and within narrow geometric features. *Boundary Value Caching* shows noticeable bias, resulting in larger errors, whereas our results do not exhibit evident bias. The *Off-Centered WoS Solver* incurs additional overhead due to its statistical weighting scheme, leading to noisier results under the same time budget.

We further evaluate the convergence behavior of all methods in Figure 11. HC can be competitive at intermediate time budgets in some cases, but under tight budgets it remains less accurate than our method, and at longer runtimes it tends to level off at a higher

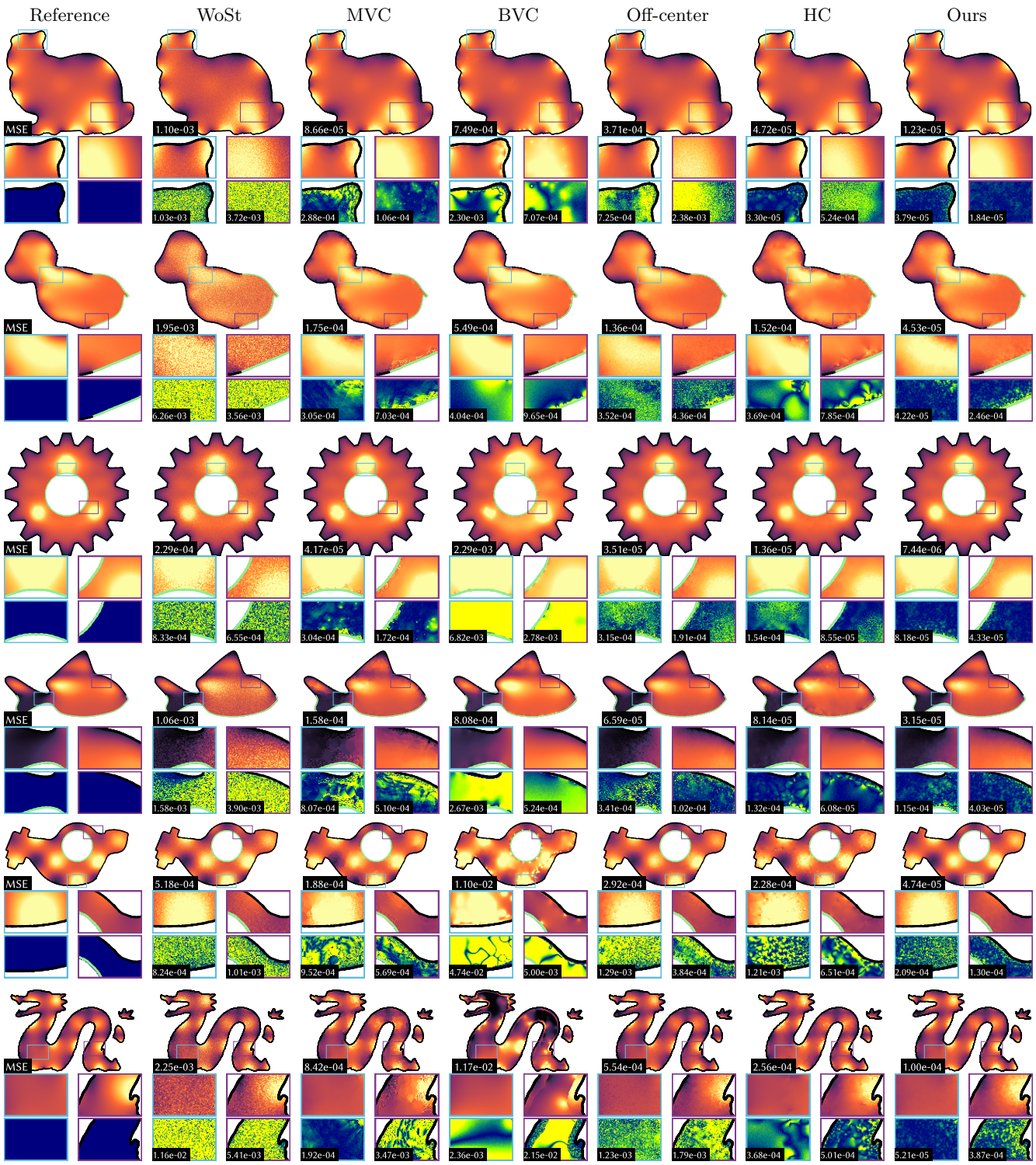

Fig. 10. Equal-time comparison with Walk-on-Star (WoSt) [Sawhney et al. 2023] and selected variance reduction techniques, including *Mean-Value Caching* (MVC) [Bakbouk and Peers 2023], *Boundary-Value Caching* (BVC) [Miller et al. 2023], *Harmonic Caching* (HC), and the *Off-Centered WoS Solver* (OC) [Bao et al. 2025]. Dirichlet and Neumann boundaries are shown in black and green, respectively. Reference solutions are computed using WoSt with 65536 random walks. Results in the top four rows correspond to an approximate runtime of 10 seconds, while the bottom two rows correspond to a 2-second time budget.

error floor, especially in source-driven settings. Overall, both variants of our method converge faster, with the single-buffer variant typically achieving the lowest error.

Irregularly distributed evaluation points. To demonstrate that our method applies to arbitrary sets of evaluation points, we construct two scenes in which the evaluation points are given by vertices of a 3D shape. This setting complements our earlier experiments on regular 2D grids and validates the method under more general and practically relevant conditions. We illustrate the constructed model problems and present an equal-time comparison in Figure 12. In these experiments, we define neighbor relations using different strategies: kNN is used in the first scene, while the input mesh adjacency is used in the second scene, demonstrating the flexibility of our framework in handling different neighborhood definitions. Our method remains robust and achieves lower error than the compared techniques in both cases.

5 Conclusion

We introduced a gradient-domain formulation for Monte Carlo Poisson solvers, bringing ideas from gradient-domain rendering into the grid-free stochastic PDE setting for the first time. Our method constructs unbiased Monte Carlo estimators for solution differences between spatially varying query points, reuses shared random-walk information, and integrates primal and difference estimates through generalized unbiased reconstruction. Across a range of model problems and geometric settings, this approach achieves faster convergence and lower variance than existing variance reduction techniques, suggesting a new and effective direction for Monte Carlo PDE solvers on complex domains.

Limitation and Future work. In our gradient-domain reconstruction framework, we use WoSt as the underlying pointwise estimator. Although WoSt is effective in the interior of the domain, it exhibits higher variance in the vicinity of Neumann boundaries, which can reduce the efficiency of our method when evaluation points lie close to such boundaries. Incorporating complementary techniques such as neural guiding [Huang et al. 2025] may help mitigate this limitation. Another promising avenue is to explore our idea of *shared walk sequences* in a ReSTIR [Bitterli et al. 2020]-based pipeline to facilitate efficient information reuse.

Acknowledgments

We would like to thank all anonymous reviewers for their insightful comments and valuable suggestions. This work was supported by the National Natural Science Foundation of China (Project No. 62372257) and the Beijing Natural Science Foundation (Project No. QY25035).

References

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2020. Deep combiner for independent and correlated pixel estimates. *ACM Trans. Graph.* 39, 6, Article 242 (Nov. 2020), 12 pages. doi:10.1145/3414685.3417847

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2023. Input-Dependent Uncorrelated Weighting for Monte Carlo Denoising. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 9, 10 pages. doi:10.1145/3610548.3618177

Ghada Bakbouk 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. doi:10.2312/sr.20231120

Anchang Bao, Jie Xu, Enya Shen, and Jianmin Wang. 2025. Off-Centered WoS-Type Solvers with Statistical Weighting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 127, 11 pages. doi:10.1145/3757377.3763852

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (Aug. 2020), 17 pages. doi:10.1145/3386569.3392481

Michael Czekanski, Benjamin Faber, Margaret Fairborn, Adelle Wright, and David Bindel. 2025. Walking on Spheres and Talking to Neighbors: Variance Reduction for Laplace's Equation. arXiv:2404.17692 [physics.comp-ph] https://arxiv.org/abs/2404.17692

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen. https://libeigen.gitlab.io.

Jie Guo, Mengtian Li, Quewei Li, Yuting Qiang, Bingyang Hu, Yanwen Guo, and Ling-Qi Yan. 2019. GradNet: unsupervised deep screened poisson reconstruction for gradient-domain rendering. *ACM Trans. Graph.* 38, 6, Article 223 (Nov. 2019), 13 pages. doi:10.1145/3355089.3356538

Paul Himmler and Tobias Günther. 2025. Conformal First Passage for Epsilon-free Walk-on-Spheres. *ACM Trans. Graph.* 44, 4, Article 40 (July 2025), 11 pages. doi:10.1145/3730942

Tianyu Huang, Jingwang Ling, Shuang Zhao, and Feng Xu. 2025. Guiding-Based Importance Sampling for Walk on Stars. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. doi:10.1145/3721238.3730593

Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. 2019. Deep convolutional reconstruction for gradient-domain rendering. *ACM Trans. Graph.* 38, 4, Article 126 (July 2019), 12 pages. doi:10.1145/3306346.3323038

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4, Article 123 (July 2015), 13 pages. doi:10.1145/2766997

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. 2013. Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4, Article 95 (July 2013), 12 pages. doi:10.1145/2461912.2461943

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. doi: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. doi:10.1145/3641519.3657395

Yuzhi Liang, Tao Liu, Yuchi Huo, Rui Wang, and Hujun Bao. 2024. Adaptive sampling and reconstruction for gradient-domain rendering. *Computational Visual Media* 10, 5 (2024), 885–902. doi:10.1007/s41095-023-0361-5

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Trans. Graph.* 42, 4, Article 82 (July 2023), 11 pages. doi:10.1145/3592400

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024a. Differential Walk on Spheres. *ACM Trans. Graph.* 43, 6, Article 174 (Nov. 2024), 18 pages. doi:10.1145/3687913

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2024b. Walkin' Robin: Walk on Stars with Robin Boundary Conditions. *ACM Trans. Graph.* 43, 4, Article 41 (July 2024), 18 pages. doi:10.1145/3658153

Mervin E. Muller. 1956. Some Continuous Monte Carlo Methods for the Dirichlet Problem. *The Annals of Mathematical Statistics* 27, 3 (1956), 569–589. http://www.jstor.org/stable/2237369

Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. In *ACM SIGGRAPH 2003 Papers* (San Diego, California) *(SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 313–318. doi:10.1145/1201775.882269

Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A Bidirectional Formulation for Walk on Spheres. 41, 4 (2022). doi:10/jgzr

Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. 2016. Image-Space Control Variates for Rendering. 35, 6 (Nov. 2016), 169:1–169:12. doi:10/f9cphw

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, Article 123 (Aug. 2020), 18 pages. doi:10.1145/3386569.3392374

Rohan Sawhney and Bailey Miller. 2023. *Zombie: Grid-Free Monte Carlo Solvers for Partial Differential Equations.*

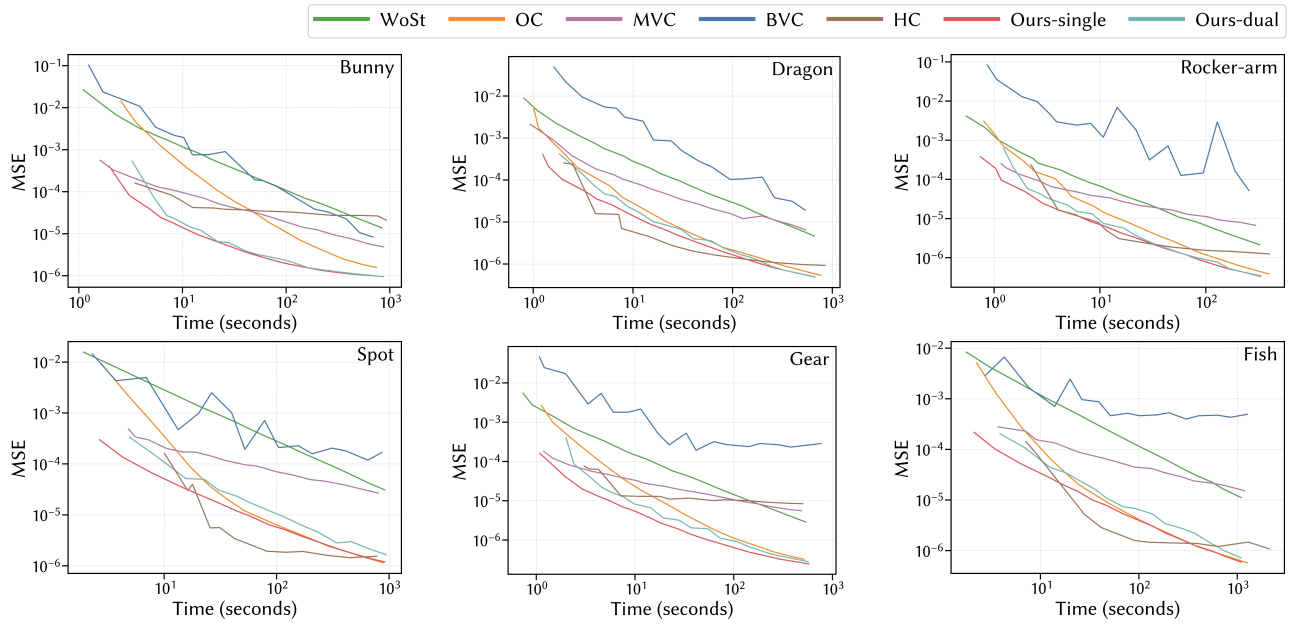

Fig. 11. Convergence curves of the compared methods. The mean squared error (MSE) is plotted as a function of computation time. Both variants of our method are shown: the single-buffer variant (Ours-single) and the dual-buffer variant (Ours-dual).

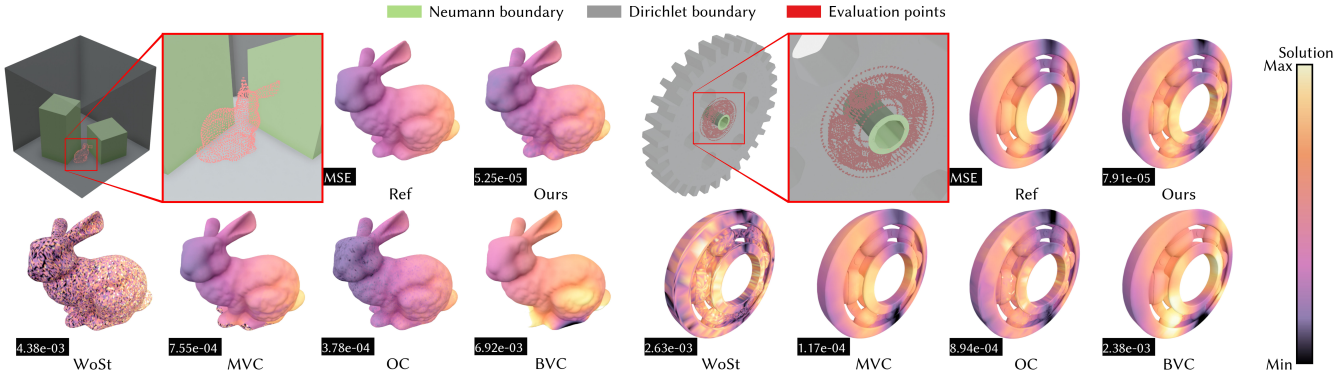

Fig. 12. Results on irregular evaluation points sampled from model vertices. For the first scene, evaluation points are taken from a bunny model placed in a closed Cornell box, with neighborhoods constructed using k-nearest neighbors (k = 6). For the second scene, evaluation points are taken from a bearing model embedded in a gear-shaped domain, where neighborhoods are defined by the input mesh connectivity. We present an equal-time comparison with Walk-on-Star (WoSt) [Sawhney et al. 2023], *Mean-Value Caching* (MVC) [Bakbouk and Peers 2023], *Boundary-Value Caching* (BVC) [Miller et al. 2023], and the *Off-Centered WoS Solver* (OC) [Bao et al. 2025], in which our method consistently achieves lower error.

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, Article 80 (July 2023), 20 pages. doi:10.1145/3592398

Difei Yan, Zengyu Li, Lifan Wu, and Kun Xu. 2025. Generalized Unbiased Reconstruction for Gradient-Domain Rendering. *ACM Trans. Graph.* 44, 6, Article 211 (Dec. 2025), 14 pages. doi:10.1145/3763297

Difei Yan, Shaokun Zheng, Ling-Qi Yan, and Kun Xu. 2024. Filtering-Based Reconstruction for Gradient-Domain Rendering. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 69, 10 pages. doi:10.1145/3680528.3687568

Ekrem Fatih Yilmazer, Delio Vicini, and Wenzel Jakob. 2024. Solving Inverse PDE Problems using Grid-Free Monte Carlo Estimators. *ACM Trans. Graph.* 43, 6, Article 175 (Nov. 2024), 18 pages. doi:10.1145/3687990

Zihan Yu, Rohan Sawhney, Bailey Miller, Lifan Wu, and Shuang Zhao. 2025. Robust Derivative Estimation with Walk on Stars. *ACM Trans. Graph.* 44, 6, Article 254 (Dec. 2025), 16 pages. doi:10.1145/3763333

Zihan Yu, Lifan Wu, Zhiqian Zhou, and Shuang Zhao. 2024. A Differential Monte Carlo Solver For the Poisson Equation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. doi:10.1145/3641519.3657460

Zihong Zhou, Eugene d'Eon, Rohan Sawhney, and Wojciech Jarosz. 2025. Harmonic Caching for Walk on Spheres. *ACM Trans. Graph.* 44, 6, Article 253 (Dec. 2025), 15 pages. doi:10.1145/3763322