Eurographics Symposium on Rendering 2026
I. Gkioulekas and A. Jarabo
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 4

Optimized and Aligned Anisotropic Monte Carlo Sampling Patterns - Supplemental

Mirco Werner, Johannes Hanika, and Carsten Dachsbacher

Karlsruhe Institute of Technology, Germany

1. Minimizing the Reverse KL With Gradient Descent

In general, to minimize $D_{\mathrm{KL}}(q(\cdot \mid \mu,\kappa)\|p(\cdot))$, the gradients with respect to μ and κ are required. The gradient with respect to μ is given by

$$\begin{aligned}
&\frac{\partial D_{\mathrm{KL}}(q(\cdot \mid \mu,\kappa)\|p(\cdot))}{\partial \mu} = -\frac{\partial}{\partial \mu}\mathbb{E}_{x\sim q(\cdot|\mu,\kappa)}\left[\log p(x)\right] \\
&= -\int_0^{2\pi} \frac{\partial}{\partial \mu} q(x \mid \mu,\kappa)\log p(x)dx \\
&= -\int_0^{2\pi} q(x \mid \mu,\kappa)\kappa\sin(x-\mu)\log p(x)dx \\
&= -\kappa\mathbb{E}_{x\sim q(\cdot|\mu,\kappa)}\left[\sin(x-\mu)\log p(x)\right]. \qquad (1)
\end{aligned}$$

To optimize for κ, we add the term $\lambda\kappa^2$ ($\lambda \geq 0$) for regularization to prevent κ from growing infinitely large. The gradient with respect to κ is then given by

$$\begin{aligned}
&\frac{\partial D_{\mathrm{KL}}(q(\cdot \mid \mu,\kappa)\|p(\cdot)) + \lambda\kappa^2}{\partial \kappa} \\
&= \frac{I_1(\kappa)}{I_0(\kappa)} + \kappa\left(\frac{\left(I_0(\kappa) - \frac{1}{\kappa}I_1(\kappa)\right)I_0(\kappa) - I_1(\kappa)^2}{I_0(\kappa)^2}\right) - \frac{2\pi I_1(\kappa)}{2\pi I_0(\kappa)} \\
&\quad - \int_0^{2\pi} \frac{\partial}{\partial \kappa} q(x \mid \mu,\kappa)\log p(x)dx + 2\lambda\kappa \\
&= \kappa - \frac{I_1(\kappa)}{I_0(\kappa)} - \kappa\frac{I_1(\kappa)^2}{I_0(\kappa)^2} \\
&\quad - \int_0^{2\pi} q(x \mid \mu,\kappa)\left(\cos(x-\mu) - \frac{I_1(\kappa)}{I_0(\kappa)}\right)\log p(x)dx + 2\lambda\kappa \\
&= \kappa - \frac{I_1(\kappa)}{I_0(\kappa)} - \kappa\frac{I_1(\kappa)^2}{I_0(\kappa)^2} \\
&\quad - \mathbb{E}_{x\sim q(\cdot|\mu,\kappa)}\left[\left(\cos(x-\mu) - \frac{I_1(\kappa)}{I_0(\kappa)}\right)\log p(x)\right] + 2\lambda\kappa. \qquad (2)
\end{aligned}$$

Since there is no closed-form solution for the parameters when setting the gradients to zero, we minimize $D_{\mathrm{KL}}(q(\cdot \mid \mu,\kappa)\|p(\cdot))$ using gradient descent with learning rates α_μ and α_κ:

$$\mu_{t+1} = \mu_t - \alpha_\mu \frac{\partial D_{\mathrm{KL}}(q(\cdot \mid \mu_t,\kappa_t)\|p(\cdot))}{\partial \mu_t} \qquad (3)$$

$$\kappa_{t+1} = \kappa_t - \alpha_\kappa \frac{\partial D_{\mathrm{KL}}(q(\cdot \mid \mu_t,\kappa_t)\|p(\cdot)) + \lambda\kappa^2}{\partial \kappa_t} \qquad (4)$$

Please refer to the provided source code for minimizing the reverse KL divergence using gradient descent.

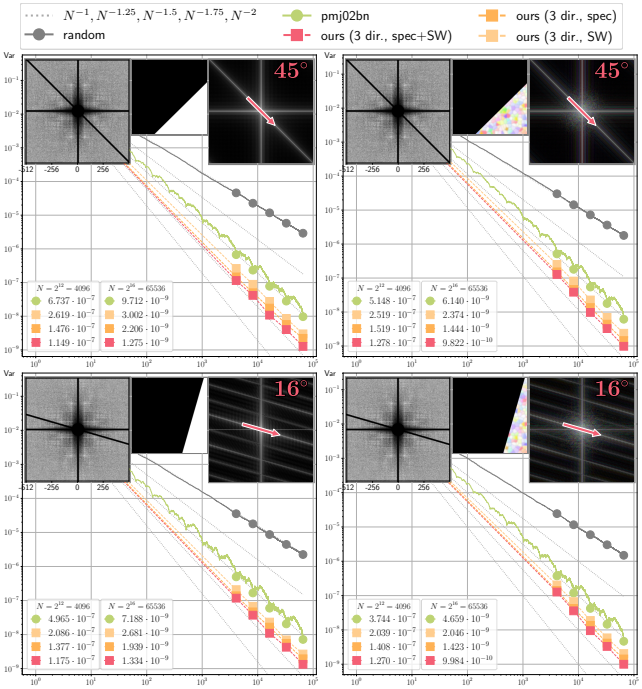


Figure 1: Comparison of our optimized and aligned point sets when using the spectrum loss $\mathcal{L}_{\mathtt{spec}}(S)$, the sliced Wasserstein loss $\mathcal{L}_{\mathtt{SW}}(S)$, or both losses $\mathcal{L}_{\mathtt{spec}}(S) + 10\mathcal{L}_{\mathtt{SW}}(S)$. We observe that combining both losses reduces the variance more significantly.

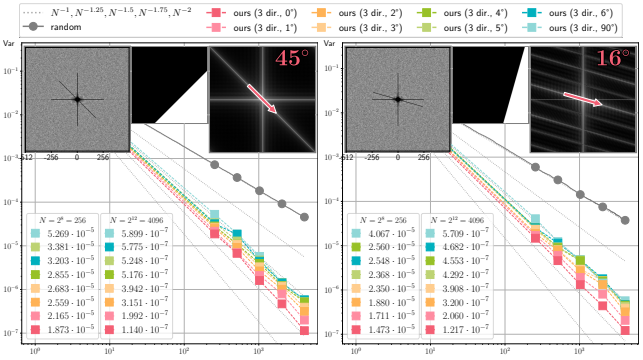


Figure 2: Evaluating misalignment. We show misalignment of $1°, 2°, \ldots, 6°$, and $90°$. This behaves as expected: The larger the misalignment, the worse the convergence rate. However, for slight misalignment, our convergence rate is still superior compared to existing unaligned anisotropic sampling patterns.

2. Spectrum and Sliced Wasserstein Loss

In Fig. 1, we evaluate the influence of both loss terms for our gradient-based point optimization. Combining the spectrum and the Wasserstein loss reduces variance more significantly.

3. Misalignment

In Fig. 2, we evaluate the influence of misalignment, e.g., due to a slightly wrong estimated principal direction. The convergence rate gets worse the more misaligned the points become. However, for slight misalignment, our convergence rate is still superior compared to existing unaligned anisotropic sampling patterns.

4. Increasing Number of Discontinuities

In the main paper, we focus on improving the variance convergence on integrands that exhibit high energies along the horizontal, vertical, and up to two additional directions. Here, in Fig. 3, we evaluate the variance convergence of point sets optimized for integrands with an increasing number of differently oriented discontinuities. With an increasing number of directions, along which frequencies need to be suppressed, fewer frequencies can be suppressed along each individual direction, resulting in increased variance. Consequently, it is only useful to align with additional directions up to a certain number of straight and reliably detectable discontinuities.

5. Discontinuities at Domain Boundaries

The finite integration domain $[0,1)^2$ can cause horizontal and vertical discontinuities that affect the convergence rate. In Fig. 4, we show this behavior using an isotropic (blue noise) point set that suppresses only low frequencies and an anisotropic point set that additionally suppresses higher frequencies along the horizontal and vertical axes. Both point sets are optimized using our method.

6. Parameters for Our Point Optimization

Table 1 lists the suppressed frequencies and optimization times used for our sampling sets with three anisotropy directions with

#Points	#Frequencies (length, width, blue noise)	Optimization Time [min:s]
$2^{12} = 4096$	4303 $(256, 5, 24)$	00:17
$2^{13} = 8192$	9169 $(384, 7, 32)$	00:52
$2^{14} = 16384$	16099 $(512, 9, 48)$	02:49
$2^{15} = 32768$	32708 $(768, 13, 56)$	10:45
$2^{16} = 65536$	55741 $(1024, 17, 64)$	35:32

Table 1: Parameters used to synthesize our sampling sets shown in Figure 5 in the paper. We report the total number of frequencies for which the power spectrum is computed, i.e., the number of frequencies that are suppressed. We suppress frequencies along three directions, each with the same length and width. Additionally, low frequencies are suppressed in a certain radius (blue noise). For optimization with gradient descent, we use PyTorch with CUDA support on an NVIDIA RTX 5080.

$2^{12}, \ldots, 2^{16}$ points shown in the third row in Figure 5 in the paper. We initialize the learning rate of Adam [KB17] with 10^{-5}. We reduce the learning rate by a factor of 10 and revert to the previous state of the sampling points each time when the optimization becomes unstable. This helps with faster convergence. We found out experimentally that using a balancing factor of $c = 10$ for $\mathcal{L}_{\mathrm{SW}}(S)$ achieves good results. We use a set of Owen-scrambled Sobol points [Sob67, Owe95, Bur20] to initialize our optimizer. Although it is possible to start from white noise, we found that this helps with faster convergence. The parameters of the other point sets shown in the paper, as well as our implementation in PyTorch [PGM*19] can be found in the provided source code in our supplemental material.

7. Aligning our Anisotropic Sampling Sets

When optimizing for three directions (horizontal, vertical, and one arbitrary principal direction), we need to pre-compute 46 point sets for arbitrary directions between $0°, 1°, \ldots, 45°$. Then, we select the matching set during runtime. Using symmetries, the 46 point sets cover the entire $180°$ domain. In Algorithm 1, we show how to determine which of the 46 sampling sets to load and what operations need to be applied to align with the requested direction. For four directions (horizontal, vertical, and two arbitrary directions), by exploiting symmetries, we need to pre-compute $4,141$ sets. Please refer to the source code in our supplemental material that shows how to load the matching set. The subsequent operations applied to the set are the same as in the previous case of aligning only with one arbitrary direction.

8. Shearing Existing Anisotropic Sampling Patterns

To align existing anisotropic sampling patterns that have two low-energy directions along the axes in their power spectra, we use shearing as proposed by Singh et al. [SJ17]. By shearing the sampling points either vertically or horizontally, the strata are sheared

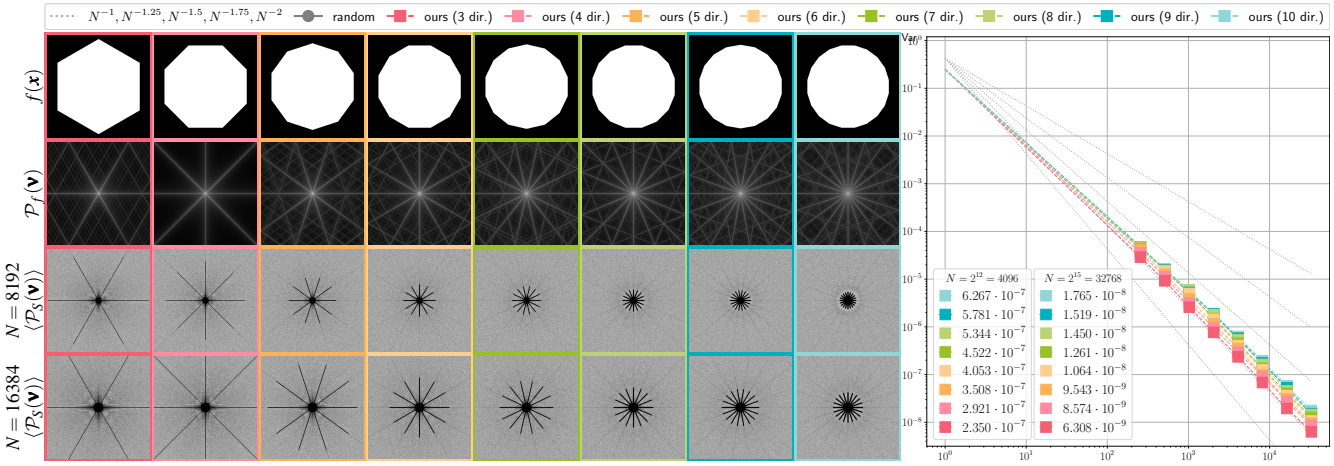


Figure 3: Variance convergence of point sets optimized for integrands with an increasing number $(3,4,\ldots,10)$ of differently oriented discontinuities. With an increasing number of edges of the polygon in the integrand $f(\boldsymbol{x})$, we need to suppress frequencies in more directions to obtain the improved asymptotic convergence rate to $\mathcal{O}(N^{-1.75})$. Additionally, as the length per edge decreases with more edges in total, the angular range (the width of the high-energy directions in the power spectrum of the integrand $\mathcal{P}_f(\mathbf{v})$) increases, which means that we must also suppress further frequencies that are orthogonal to each direction. We use our point synthesis technique to optimize point sets whose expected power spectra $\langle \mathcal{P}_S(\mathbf{v}) \rangle$ closely match the respective power spectrum of the integrand $\mathcal{P}_f(\mathbf{v})$. With an increasing number of discontinuities, fewer high frequencies can be suppressed when using N points, leading to increased variance.

Algorithm 1: Aligning our anisotropic sampling sets with low energies along the vertical, horizontal, and an additional principal direction.

Input: Principal direction $\theta \in \{0°,\ldots,179°\}$
Output: Aligned sampling points

```
// determine which of the sampling points
   optimized for principal directions
   φ ∈ {0°,...,45°} to load and necessary
   operations for aligning
φ, swap, invert ← θ, false, false
if 0° ≤ θ ≤ 45° then
    φ, swap, invert ← θ, false, false
if 45° < θ ≤ 90° then
    φ, swap, invert ← 90° − θ, true, false
if 90° < θ ≤ 135° then
    φ, swap, invert ← θ − 90°, true, true
if 135° < θ < 180° then
    φ, swap, invert ← 180° − θ, false, true
// load and align the sampling points
S ← loadSamplingPoints(φ)
for s ∈ S do
    if swap then
        s ← (s[1], s[0])^T
    if invert then
        s[0] ← 1 − s[0]
return S
```

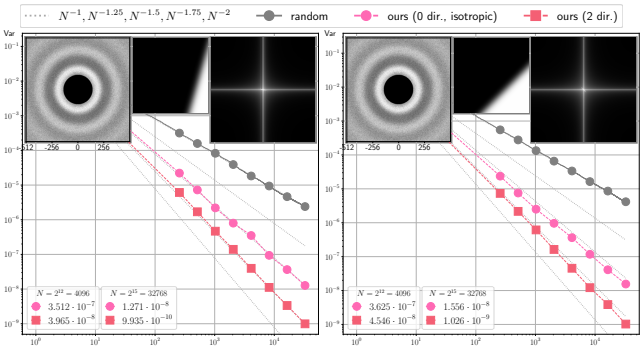


Figure 4: Variance convergence on integrands that feature only horizontal and vertical discontinuities due to the finite integration domain. We optimize an isotropic (blue noise) point set that suppresses only low frequencies and an anisotropic point set that additionally suppresses higher frequencies along the horizontal and vertical axes. Suppressing frequencies along horizontal and vertical improves the convergence rate. In the top left corner, we show the expected power spectrum of our optimized isotropic point set.

as well. As a result, the sampling points are stratified along the desired direction, and the power spectrum exhibits low energies there. Whether we shear horizontally or vertically is based on which of the axes is closer to the principal direction. Algorithm 2 shows pseudocode for performing the shearing operation.

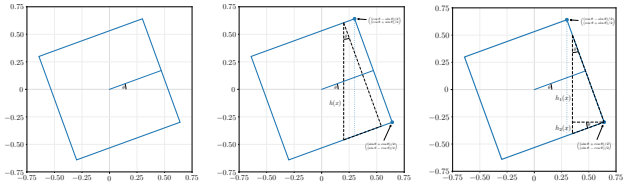
9. Target Locations for the Sliced Wasserstein Loss

In Section 4.2 in the paper, we use the sliced Wasserstein loss to ensure stratification of the synthesized points along a set of directions,

Algorithm 2: Aligning existing anisotropic sampling patterns with low energies along the vertical and horizontal axes.

Input: Principal direction $\theta \in \{0°, \dots, 179°\}$
Output: Aligned sampling points

```
// determine the shear parameters
```
$\boldsymbol{p}_{\text{shear}} \leftarrow (0,0)^T$
$\theta_r \leftarrow \theta \frac{\pi}{180}$ // radians
if $|\cos\theta_r| \geq \frac{1}{\sqrt{2}}$ **then**
  // $0° \leq \theta \leq 45°$ or $135° \leq \theta \leq 180°$
  $\boldsymbol{p}_{\text{shear}}[0] \leftarrow \frac{\sin\theta_r}{\cos\theta_r}$ // $\tan\theta_r$
else
  $\boldsymbol{p}_{\text{shear}}[1] \leftarrow \frac{\cos\theta_r}{\sin\theta_r}$ // $\tan\theta_r{}^{-1}$
// load and align the sampling points, e.g., pmj02bn
$\mathcal{S} \leftarrow$ `loadSamplingPoints(...)`
for $\boldsymbol{s} \in \mathcal{S}$ **do**
  $\boldsymbol{s} \leftarrow \left(\boldsymbol{s} - \boldsymbol{p}_{\text{shear}} \odot (\boldsymbol{s}[1], \boldsymbol{s}[0])^T\right) \mod \mathbf{1}$
return $\mathcal{S}$



(a) (b) (c)

Figure 5: Derivation of the height of a rotated unit square located at the origin along the x-axis. (a) The unit square is rotated by an angle θ from the x-axis. (b) For $0 \leq x \leq \frac{\cos\theta-\sin\theta}{2}$, the height $h(x)$ is constant and depends only on θ. (c) For $\frac{\cos\theta-\sin\theta}{2} < x \leq \frac{\sin\theta+\cos\theta}{2}$, the height $h(x) = h_1(x) + h_2(x)$ decreases and depends on x and θ.

which requires defining target positions $D_{\text{TARGET}}(\boldsymbol{d})$ for each of the directions $\boldsymbol{d}$. To synthesize uniformly distributed points in the unit square, the target positions $D_{\text{TARGET}}(\boldsymbol{d})$ must be distributed proportional to the height of the square projected onto the rotated direction. Therefore, first, a PDF needs to be defined that determines the height of the square at a given position. Using one-dimensional equidistant points u, the target positions that are distributed according to the PDF are then determined using inverse transform sampling. The formula for the inverse CDF $F_\theta^{-1}(u)$ has been derived by Wolfe et al. [Wol23]. However, some steps in the derivation are missing. Therefore, we complete their derivation here.

To facilitate the derivation, the unit square is shifted from $[0,1]^2$ to $[-0.5,0.5]^2$ symmetrically around the origin. Additionally, instead of considering a rotated direction with angle θ inside the square, the square is rotated accordingly, and the heights in the rotated square along the x-axis are calculated. The setting is visualized in Fig. 5a. As a consequence, the CDF outputs values in the range of $[-0.5,0.5]$ instead of $[0,1]$. The problem is symmetric for $x<0$ and $x>0$, which allows us to only consider the case of $x \geq 0$ in the derivation.

The square is defined on $0 \leq x \leq \frac{\sin\theta+\cos\theta}{2}$, where the latter bound can be determined by rotating the corner of the original square by the angle θ. Calculating the height requires distinguishing two cases: In the first case (Fig. 5b) for $0 \leq x < \frac{\cos\theta-\sin\theta}{2}$, the height is constant and depends only on θ. Using trigonometry, we find the relation between the height $h(x)$ and θ as

$$\cos\theta = \frac{1}{h(x)}. \tag{5}$$

In the second case (Fig. 5c) for $\frac{\cos\theta-\sin\theta}{2} \leq x \leq \frac{\sin\theta+\cos\theta}{2}$, the height decreases and depends both on x and θ. The height is decomposed into the sum of two heights $h_1(x)$ and $h_2(x)$ (see Fig. 5c) with the following relations:

$$\tan\theta = \frac{\frac{\cos\theta+\sin\theta}{2} - x}{h_1(x)} \quad \text{and} \quad \tan\theta = \frac{h_2(x)}{\frac{\cos\theta+\sin\theta}{2} - x} \tag{6}$$

The total height is given as

$$\begin{aligned} h_1(x) + h_2(x) &= \left(\frac{\cos\theta+\sin\theta}{2} - x\right)\left(\frac{1}{\tan\theta} + \tan\theta\right) \\ &= \left(\frac{\cos\theta+\sin\theta}{2} - x\right)\frac{1}{\sin\theta\cos\theta}. \end{aligned} \tag{7}$$

Since the area of the unit square is 1, the piecewise-defined heights already integrate to 1 (or to 0.5 for $x \geq 0$) and can be directly used as the PDF p:

$$p_\theta(x) = \begin{cases} \frac{1}{\cos\theta}, & \text{if } x < \frac{\cos\theta-\sin\theta}{2} \\ \left(\frac{\cos\theta+\sin\theta}{2} - x\right)\frac{1}{\sin\theta\cos\theta}, & \text{else if } x < \frac{\cos\theta+\sin\theta}{2} \end{cases} \tag{8}$$

Next, both parts of the PDF are integrated. For $0 \leq x < \frac{\cos\theta-\sin\theta}{2}$ we obtain

$$\int_0^x p_\theta(x')dx' = \int_0^x \frac{1}{\cos\theta}dx' = \frac{x}{\cos\theta}, \tag{9}$$

and for $\frac{\cos\theta-\sin\theta}{2} < x$ we get

$$\begin{aligned} &\int_0^x p_\theta(x')dx' \\ &= \int_0^{\frac{\cos\theta-\sin\theta}{2}} \frac{1}{\cos\theta}dx' + \int_{\frac{\cos\theta-\sin\theta}{2}}^x \left(\frac{\cos\theta+\sin\theta}{2} - x'\right)\frac{1}{\sin\theta\cos\theta}dx' \\ &= \frac{\cos\theta-\sin\theta}{2\cos\theta} - \frac{1}{2}\left(\frac{\cos\theta+\sin\theta}{2} - x\right)^2 \frac{1}{\sin\theta\cos\theta} \\ &\quad + \frac{1}{2}\left(\frac{\cos\theta+\sin\theta}{2} - \frac{\cos\theta-\sin\theta}{2}\right)^2 \frac{1}{\sin\theta\cos\theta} \\ &= \frac{\cos\theta-\sin\theta}{2\cos\theta} - \frac{1}{2}\left(\frac{\cos\theta+\sin\theta}{2} - x\right)^2 \frac{1}{\sin\theta\cos\theta} + \frac{\sin\theta}{2\cos\theta} \\ &= \frac{1}{2} - \frac{1}{2}\left(\frac{\cos\theta+\sin\theta}{2} - x\right)^2 \frac{1}{\sin\theta\cos\theta}. \end{aligned} \tag{10}$$

The CDF F_θ is

$$F_\theta(x) = \begin{cases} \frac{x}{\cos\theta}, & \text{if } x < \frac{\cos\theta-\sin\theta}{2} \\ \frac{1}{2} - \frac{1}{2}\left(\frac{\cos\theta+\sin\theta}{2} - x\right)^2 \frac{1}{\sin\theta\cos\theta}, & \text{else if } x < \frac{\cos\theta+\sin\theta}{2}. \end{cases} \tag{11}$$

The last step is to find the inverse CDF. Inverting the first part using

$u \in [0, 0.5]$ yields

$$\frac{x}{\cos\theta} = u \Leftrightarrow x = u\cos\theta. \tag{12}$$

Inverting the second part, we obtain

$$\begin{aligned}
&\frac{1}{2} - \frac{1}{2}\left(\frac{\cos\theta + \sin\theta}{2} - x\right)^2 \frac{1}{\sin\theta\cos\theta} = u \\
\Leftrightarrow &x^2 - (\cos\theta + \sin\theta)x + \frac{(\cos\theta + \sin\theta)^2}{4} \\
&\quad - \sin\theta\cos\theta + 2u\sin\theta\cos\theta = 0 \\
\Rightarrow &x_{1,2} = \frac{\cos\theta + \sin\theta}{2} \pm \\
&\sqrt{\cancel{\frac{(\cos\theta + \sin\theta)^2}{4}} - \cancel{\frac{(\cos\theta + \sin\theta)^2}{4}} + \sin\theta\cos\theta - 2u\sin\theta\cos\theta} \\
\Leftrightarrow &x_{1,2} = \frac{\cos\theta + \sin\theta}{2} \pm \sqrt{2\sin\theta\cos\theta\left(\frac{1}{2} - u\right)}.
\end{aligned} \tag{13}$$

Since our CDF is only defined for $x \leq \frac{\cos\theta+\sin\theta}{2}$, only the solution with the negative sign makes sense. Finally, we determine the u, where we need to switch between the piecewise definitions:

$$\begin{aligned}
&F_\theta\left(\frac{\cos\theta - \sin\theta}{2}\right) \\
&= \frac{1}{2} - \frac{1}{2}\left(\frac{\cos\theta + \sin\theta}{2} - \frac{\cos\theta - \sin\theta}{2}\right)^2 \frac{1}{\sin\theta\cos\theta} \\
&= \frac{1}{2} - \frac{1}{2}(\sin\theta)^2 \frac{1}{\sin\theta\cos\theta} = \frac{1}{2}\left(\frac{\cancel{\sin\theta}\cos\theta - (\sin\theta)^{\cancel{2}}}{\cancel{\sin\theta}\cos\theta}\right) \\
&= \frac{\cos\theta - \sin\theta}{2\cos\theta}
\end{aligned} \tag{14, 15}$$

Our final inverted CDF defined on $u \in [0, 0.5]$ is

$$F_\theta^{-1}(u) = \begin{cases} u\cos\theta, & \text{if } u < \frac{\cos\theta-\sin\theta}{2\cos\theta} \\ \frac{\cos\theta+\sin\theta}{2} - \sqrt{2\sin\theta\cos\theta\left(\frac{1}{2} - u\right)}, & \text{else} \end{cases} \tag{16}$$

To extend F_θ^{-1} to $u \in [-0.5, 0.5]$ we exploit the symmetry by using the absolute value of u to distinguish both cases and return the negative value of the inverted CDF in the second case if u is negative:

$$F_\theta^{-1}(u) = \begin{cases} u\cos\theta, & \text{if } |u| < \frac{\cos\theta-\sin\theta}{2\cos\theta} \\ \frac{\cos\theta+\sin\theta}{2} - \sqrt{2\sin\theta\cos\theta\left(\frac{1}{2} - |u|\right)}, & \text{else if } u > 0 \\ \sqrt{2\sin\theta\cos\theta\left(\frac{1}{2} - |u|\right)} - \frac{\cos\theta+\sin\theta}{2}, & \text{else if } u < 0. \end{cases} \tag{17}$$

By mapping equidistant points $u \in [-0.5, 0.5]$ using Eq. 17, we obtain the target positions $D_{\text{TARGET}}(\boldsymbol{d})$ along the direction $\boldsymbol{d}$ that can be used (by shifting the unit square back to $[0, 1]$) to calculate the loss.

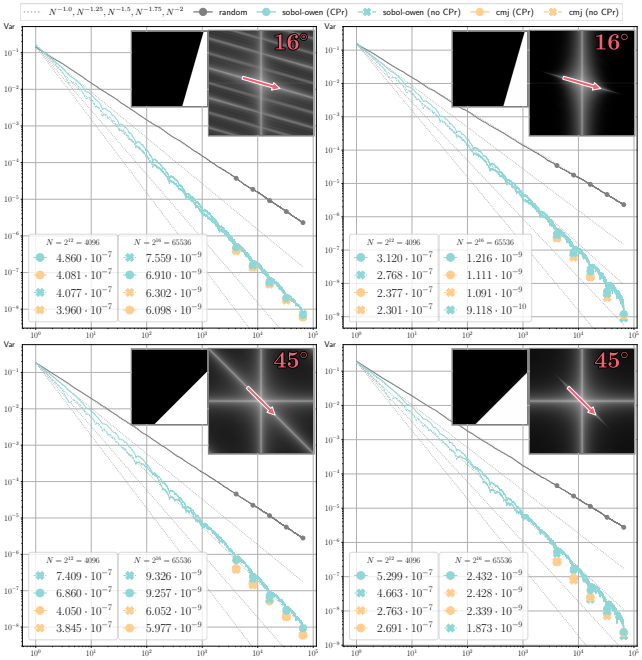


Figure 6: Variance convergence behavior on integrands with discontinuities (left) and smoothed discontinuities (right). We compare Owen-scrambled Sobol and correlated multi-jittered with (CPr) and without (no CPr) homogenization. We use Cranley-Patterson rotations (CPr) [CP76], i.e., toroidal shifts, for homogenization. There is no clear difference between homogenization and non-homogenization.

10. Influence of Homogenization on Discontinuous Integrands

We homogenize all sampling patterns used in Section 6 in the paper using toroidal shifts [CP76]. This allows us to analyze the variance of the integrand and sampling pattern using [SSC*20]

$$\mathrm{Var}(I_N) = \sum_{\mathbf{v} \in \mathbb{Z}^d \setminus \{\mathbf{0}\}} \langle \mathcal{P}_S(\mathbf{v}) \rangle \, \mathcal{P}_f(\mathbf{v}). \tag{18}$$

Eq. 18 is restricted to homogeneous sampling patterns, which is, for example, initially not the case for stratified sampling, as the subdivision into strata is fixed. Homogenization can have negative consequences on the convergence rate as it introduces boundary discontinuities in the sampling points. However, this is mainly the case for stronger smoothed integrands, while for our integrands that already exhibit (slightly smoothed) discontinuities, homogenization has barely any influence on the variance convergence rate, as shown by [SSC*20]. In Fig. 6, we verify their results on our (slightly smoothed) discontinuous integrands and observe no clear difference between homogenization and non-homogenization on our tested 2^{12} to 2^{16} samples. Singh et al. [SSC*20] develop a complete formula with additional terms to determine the variance for sampling patterns without homogenization. For our purposes, using the simplified and visually comprehensible formulation in Eq. 18 is sufficient.

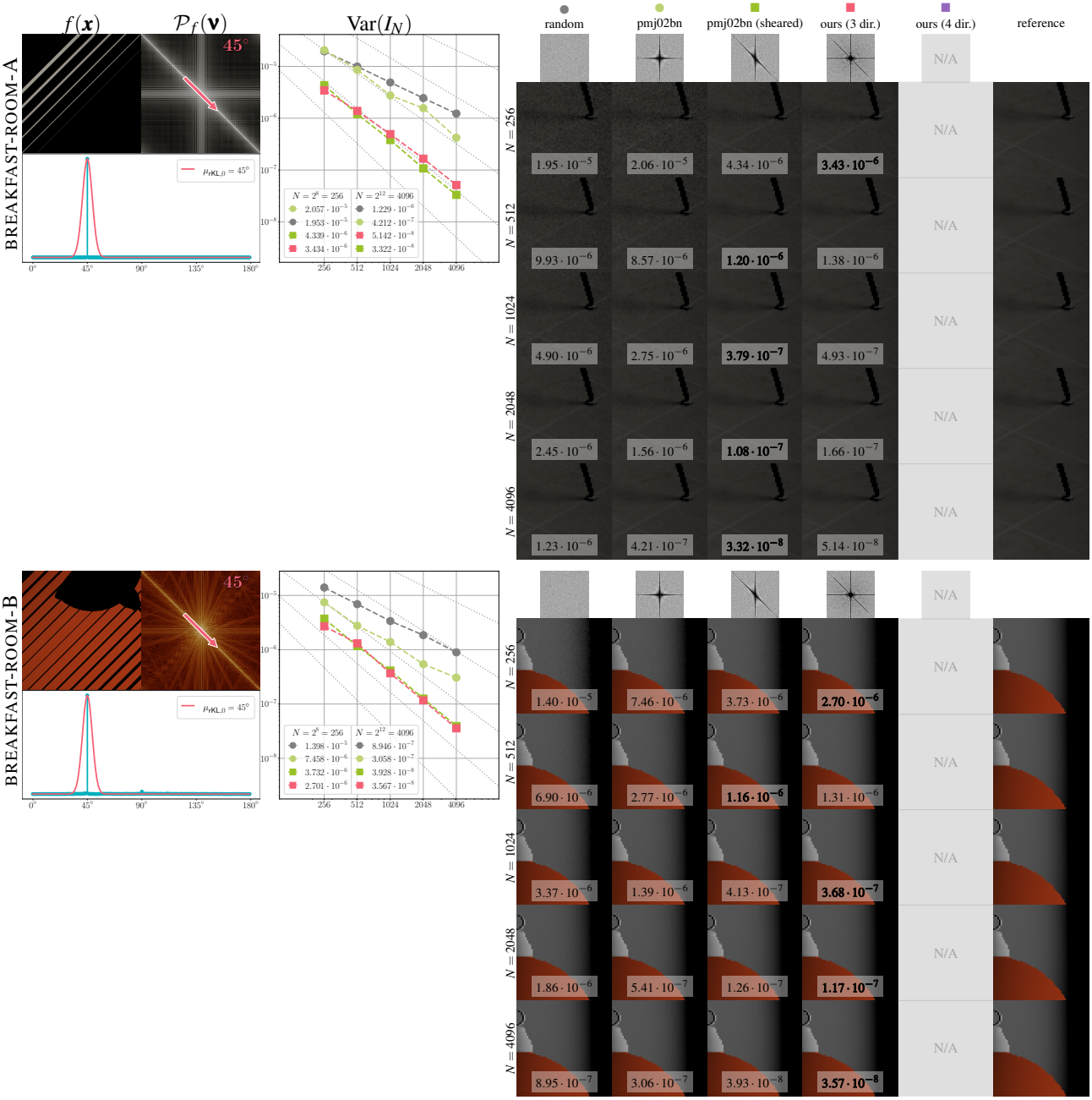


Figure 7: Variance evaluation in offline path tracing on discontinuous integrands. We show the integrand $f(\boldsymbol{x})$, its power spectrum $\mathcal{P}_f(\mathbf{v})$, the principal direction estimation, and the variance $\mathrm{Var}(I_N)$ after accumulating $2^8, 2^9, \ldots, 2^{12}$ samples per pixel. The variance is averaged over the block of pixels shown in Fig. 7. We compare unaligned pmj02bn, Singh et al.'s [SJ17] proposed shearing technique, and our aligned sampling sets with three and four low energy directions. In the BREAKFAST-ROOM scene, shearing pmj02bn and using our points to match our estimated principal direction, both significantly reduce variance and improve the convergence rate as most energy is along the direction of the high-frequency pattern. For the other integrands that feature only a few discontinuities, we observe a similar behavior as in Fig. 6 with our aligned points outperforming existing (sheared) points due to a better convergence rate. Since we improve the convergence *rate*, the difference between samplers increases with a higher number of samples. However, we can already perceive a difference in the noise in the insets with 256 samples per pixel. Please zoom in on the digital version to see all the details clearly.

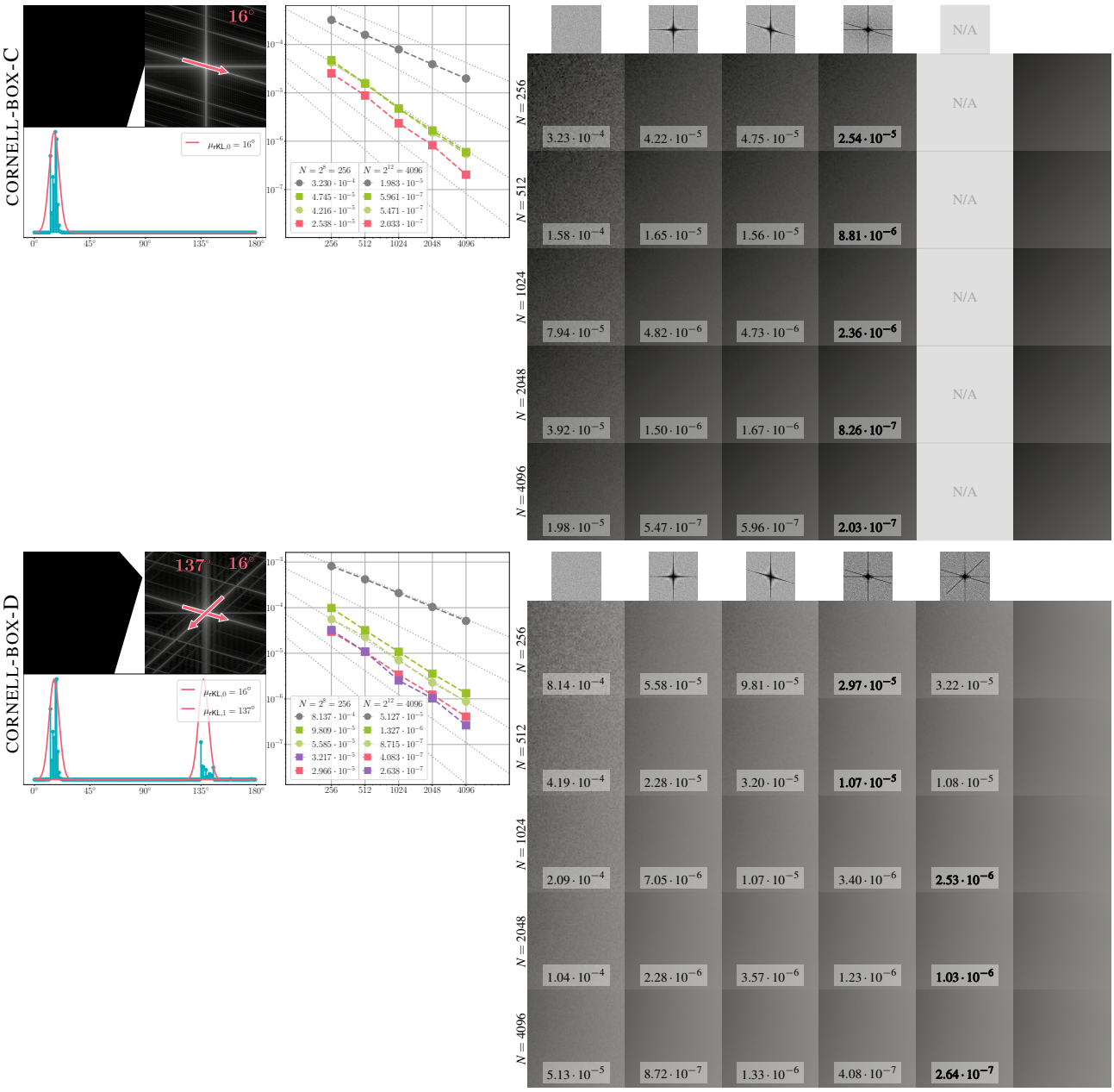


Figure 8: Variance evaluation in offline path tracing on discontinuous integrands. We show the integrand $f(\boldsymbol{x})$, its power spectrum $\mathcal{P}_f(\boldsymbol{\nu})$, the principal direction estimation, and the variance $\mathrm{Var}(I_N)$ after accumulating $2^8, 2^9, \ldots, 2^{12}$ samples per pixel. The variance is averaged over the block of pixels shown in Fig. 7. We compare unaligned pmj02bn, Singh et al.'s [SJ17] proposed shearing technique, and our aligned sampling sets with three and four low energy directions. In the BREAKFAST-ROOM scene, shearing pmj02bn and using our points to match our estimated principal direction, both significantly reduce variance and improve the convergence rate as most energy is along the direction of the high-frequency pattern. For the other integrands that feature only a few discontinuities, we observe a similar behavior as in Fig. 6 with our aligned points outperforming existing (sheared) points due to a better convergence rate. Since we improve the convergence *rate*, the difference between samplers increases with a higher number of samples. However, we can already perceive a difference in the noise in the insets with 256 samples per pixel. Please zoom in on the digital version to see all the details clearly.

11. Aligning Sampling Points in Offline Path Tracing

For a few pixel regions (marked in Fig. 7), we show detailed convergence plots when rendering the scenes with $2^8, 2^9, \ldots, 2^{12}$ samples per pixel in Fig. 9 in the paper. Here, we show the missing insets for $N = 2^9, \ldots, 2^{11}$ in Fig. 7 and Fig. 8.

References

[Bur20] BURLEY B.: Practical hash-based owen scrambling. *Journal of Computer Graphics Techniques 10*, 4 (2020), 1–20. URL: http://jcgt.org/published/0009/04/01/. 2

[CP76] CRANLEY R., PATTERSON T. N. L.: Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis 13*, 6 (1976), 904–914. doi:10.1137/0713071. 5

[KB17] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (2017). URL: https://arxiv.org/abs/1412.6980. 2

[Owe95] OWEN A. B.: Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (1995), pp. 299–317. doi:10.1007/978-1-4612-2552-2_19. 2

[PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., ET AL.: Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems* (2019), vol. 32. URL: https://arxiv.org/abs/1912.01703. 2

[SJ17] SINGH G., JAROSZ W.: Convergence analysis for anisotropic monte carlo sampling spectra. *ACM Trans. Graph. 36*, 4 (2017). doi:10.1145/3072959.3073656. 2, 6, 7

[Sob67] SOBOL' I.: On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics 7*, 4 (1967), 86–112. doi:10.1016/0041-5553(67)90144-9. 2

[SSC*20] SINGH G., SUBR K., COEURJOLLY D., OSTROMOUKHOV V., JAROSZ W.: Fourier analysis of correlated monte carlo importance sampling. *Computer Graphics Forum 39*, 1 (2020), 7–19. doi:10.1111/cgf.13613. 5

[Wol23] WOLFE A.: Deriving the inverse CDF of a rotated square projected onto a line, 2023. URL: https://blog.demofox.org/2023/12/24/deriving-the-inverse-cdf-of-a-rotated-square-projected-onto-a-line/. 4