EFFICIENT SPATIALLY-VARIANT CONVOLUTION VIA DIFFERENTIABLE SPARSE KERNEL COMPLEX

Anonymous authors
Paper under double-blind review

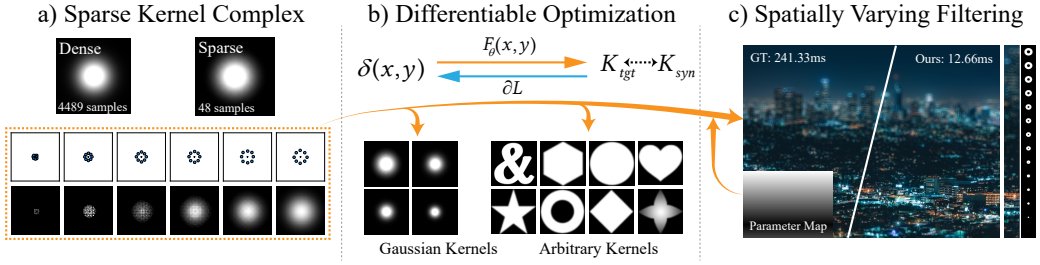

Figure 1: **An overview of our method.** We represent a dense filter as a Sparse Kernel Complex, a sequence of sparse layers whose parameters Θ are learned via Differentiable Optimization. We apply our filter F_Θ to an impulse δ to yield a synthesized kernel K_{syn}, and minimize a loss $\mathcal{L}$ against the target K_{tgt} to learn arbitrary shapes. These optimized kernels serve as a basis for high-performance Spatially Varying Filtering, achieving quality nearly-ground-truth quality at up to a 20× speedup.

ABSTRACT

Image convolution with complex kernels is a fundamental operation in photography, scientific imaging, and animation effects, yet direct dense convolution is computationally prohibitive on resource-limited devices. Existing approximations, such as simulated annealing or low-rank decompositions, either lack efficiency or fail to capture non-convex kernels. We introduce a differentiable kernel decomposition framework that represents a target spatially-variant, dense, complex kernel using a set of sparse kernel samples. Our approach features (i) a decomposition that enables differentiable optimization of sparse kernels, (ii) a dedicated initialization strategy for non-convex shapes to avoid poor local minima, and (iii) a kernel-space interpolation scheme that extends single-kernel filtering to spatially varying filtering without retraining and additional runtime overhead. Experiments on Gaussian and non-convex kernels show that our method achieves higher fidelity than simulated annealing and significantly lower cost than low-rank decompositions. Our approach provides a practical solution for mobile imaging and real-time rendering, while remaining fully differentiable for integration into broader learning pipelines.

1 INTRODUCTION

From rendering realistic depth-of-field effects (Sakurikar; Wu et al., 2022) in computational photography to modeling the intricate point spread functions (Liu et al., 2022; Shajkofci & Liebling, 2020) of optical systems, the ability to apply large, complex convolution kernels is a fundamental building block in modern vision and graphics computing systems. This creates a fundamental tension: while larger, more intricate kernels enable higher-fidelity results, their quadratic computational cost renders direct implementation impractical for interactive applications on devices ranging from mobile phones to high-end GPUs.

To bridge this gap, a rich body of work has focused on approximation strategies. For specific cases like Gaussian blur, elegant solutions (Zing, 2010; Kovesi, 2010) with constant-time complexity exist

by exploiting the filter's analytical properties. However, these specialized methods are not applicable to the arbitrary, often non-convex, kernels required for advanced effects. More general approaches, such as low-rank matrix decomposition (McGraw, 2015), can handle arbitrary kernels but often factor the operation into a series of smaller dense convolutions, limiting the potential for true sparsity and efficiency gains.

A more direct and efficient approach (Schuster et al., 2020) is to approximate a dense kernel with a truly sparse one, drastically reducing the number of required computations. Prominent strategies in this space employ heuristic-based search algorithms, such as parallel simulated annealing, to discover optimal sparse sample patterns for arbitrary kernels. While powerful in their generality, these methods often require a vast number of iterations to converge and can struggle to find high-fidelity solutions due to the non-convex nature of the optimization landscape. This reveals a critical need for a more principled and efficient method to discover high-quality sparse kernel representations.

In this work, we address this challenge by introducing a differentiable kernel decomposition framework. This approach directly optimizes the parameters of a sequence of natively sparse kernels, resulting in a highly efficient representation for runtime inference that stands in contrast to methods like low-rank decomposition. By formulating the decomposition as an end-to-end optimization problem, we can leverage the power of gradient-based methods. This marks a significant departure from heuristic search algorithms like simulated annealing, offering a more robust and efficient optimization that converges to high-fidelity solutions in significantly fewer iterations. To ensure the success of this gradient-based approach, especially for non-convex target kernels, we introduce a two-part initialization strategy that combines a structure-aware sampling method to capture intricate shapes with a deterministic radial initialization for overall stability and rapid convergence.

Beyond single-kernel approximation, our framework provides a powerful foundation for efficient spatially varying filtering. In such applications, a primary challenge is often the prohibitive overhead of generating a unique kernel for each pixel, a cost that can become a significant performance bottleneck. We address this with a novel filter-space interpolation scheme. Our method first precomputes an optimized basis of sparse filters that span a desired range of effects. At runtime, a unique sparse filter is then synthesized for each pixel by simply interpolating this compact set of basis filters. This strategy reduces the per-pixel kernel synthesis cost to a minimal set of multiply-add operations, effectively decoupling the kernel generation complexity from the image resolution and enabling complex, spatially varying effects with negligible performance impact.

Our contributions are as follows:

- A novel differentiable framework for decomposing a dense, arbitrary kernel into a sequence of optimized sparse layers, enabling efficient, high-fidelity approximation.
- A robust initialization scheme, combining a general radial strategy for stable convergence with a sparse sampling method for capturing non-convex kernels.
- A filter-space interpolation method for high-performance, spatially-varying filtering that decouples kernel synthesis cost from image resolution.

2 Related Work

2.1 High-Performance Kernel

Given that Gaussian blur is computationally expensive, numerous methods have been proposed to optimize its performance. Fast O(1) approximations of Gaussian filtering, such as the Extended Binomial Filter (Zing, 2010) and methods based on Summed-Area Tables (Kovesi, 2010), are also common. However, their reliance on pre-computation or inherently sequential processing makes them a poor fit for the massively parallel architecture of modern GPUs. A more suitable approach for modern rendering is Kawase blur (Kawase, 2003), which is a multi-pass (multi-layer) filter that requires only four texture samples per pass. This design significantly reduces the overall sample count, enabling a high-performance blur effect. As an extension to the Kawase blur, Dual Filtering (Martin et al., 2015) introduces downsampling passes followed by upsampling passes. This strategy significantly reduces memory bandwidth and the number of pixels to be processed by operating on lower-resolution textures. However, a significant limitation for the practical application of

these methods is the lack of a systematic way to map a desired Gaussian blur strength (e.g., a specific sigma value) to the corresponding parameters of the Kawase or Dual filters. Our work directly addresses this issue.

2.2 Spatio-Variant Filtering

A significant body of work has focused on learning per-pixel spatially varying convolution kernels, which have been successfully applied to a wide range of tasks, including video prediction, video frame interpolation, denoising, and deblurring (Jia et al., 2016; Niklaus et al., 2017; Mildenhall et al., 2018; Zhou et al., 2019; 2021). Diverging from existing approaches that directly predict a dense map of per-pixel kernels, our method decouples the filter generation from the spatial resolution. We achieve this by learning a highly compact lookup table (LUT) that parameterizes a continuous space of filters, enabling flexible and efficient Spatio-Variant Filtering. Spatiotemporal Variance-Guided Filtering (Schied et al., 2017) using a per-pixel combination of filters guided by estimated variance in spatial and temporal domains. Differently, our method conditions the filter generation process on an input per-pixel blur intensity map. This enables direct synthesis of filters tailored to any desired spatially-variant blur effect, without the need for intermediate statistical analysis of the image content.

2.3 Kernel Approximation and Decomposition

Inspired by Kawase blur (Kawase, 2003), High-Performance Image Filters (Schuster et al., 2020) employs parallel tempering to optimize sample patterns for sparse convolution. However, the high sensitivity of parallel tempering to its numerous hyperparameters compromises the method's overall robustness, in stark contrast to our gradient descent-based approach, which offers superior stability. In video frame interpolation, 2D kernels are decomposed into pairs of 1D kernels to significantly reduce computational complexity (Niklaus et al., 2017). To handle 3D convolutional kernels, which operate over an additional temporal dimension, STDCF (Schied et al., 2017) decomposes them into a group of spatial atoms and temporal atoms. To optimize convolutions with respect to channel correlations, depthwise separable convolution decomposes a standard convolution into two sequential, more efficient operations: a depthwise convolution followed by a 1x1 pointwise convolution (Howard et al., 2017; Chollet, 2017; Ramadhani et al., 2024). Dynamic Convolution Decomposition (Li et al., 2021; 2024) reformulates dynamic convolution by expressing the dynamic weights as a combination of static base kernels and a set of learned residuals. KDLGT (Wu et al., 2023) applies kernel decomposition techniques to accelerate the self-attention mechanism in Graph Transformers. LKD-Net (Luo et al., 2023) decomposes the large depth-wise convolution into a small depth-wise convolution and a depth-wise dilated convolution to increase the effective receptive field.

3 Preliminary

3.1 Kernel-Based Filtering

Kernel-based filtering is fundamental to many image processing tasks. This process takes an input image I_{in} and computes each pixel's value for the output image I_{out} as a weighted average of its local neighbors within I_{in}. Formally, this operation is expressed as a 2D convolution, defined as:

$$I_{out}[x, y] = (I_{in} * K)[x, y] = \sum_{i=-k}^{k} \sum_{j=-k}^{k} I_{in}[x + i, y + j] \cdot K[i, j], \tag{1}$$

where the matrix K is the $M \times M$ convolution with kernel size $M \in \mathbb{R}^+$, whose elements $K[i, j]$ are weights that determine the contribution of each neighboring pixel to the final filtered value.

3.2 Filter Representation

The dense matrix representation for the kernel K in Eq. (1) is straightforward. However, its $O(M^2)$ computational cost presents a significant bottleneck. This is especially true for filters with a large spatial support, such as a Gaussian blur with a large σ, where the cost becomes prohibitively expensive for real-time applications that demand high frame rates.

Our key insight is to approximate this expensive operation by structuring the filter as a sequence of lightweight convolutional layers, where the output of one layer serves as the input for the subsequent one. Each layer applies a highly efficient sparse kernel, K_{sparse}, which we define by a small collection of N samples with offset-weight pairs:

$$K_{sparse} = \{(\mathbf{o}_i, w_i)\}_{i=1}^{N}, \tag{2}$$

where $\mathbf{o}_i \in \mathbb{R}^2$ is the spatial offset and w_i is its corresponding weight.

The complete operation, consisting of L such layers with kernels $(K_1, K_2, ..., K_L)$, can be expressed as a nested convolution:

$$I_{out} = (...((I_{in} * K_1) * K_2) * ... * K_L). \tag{3}$$

This multi-layer filter reduces the cost to $O(\sum_{l=1}^{L} N_l)$ per pixel. Since this sum is far smaller than the number of weights in the target dense kernel ($\sum N_l \ll M^2$), the approach offers a dramatic speedup.

4 METHODOLOGY

4.1 DIFFERENTIABLE MULTI-LAYER KERNEL COMPLEX

Overview Sparse filters offer a computationally efficient alternative to dense kernels; however, they often fail to capture the intricate structure of large, complex filters. The core challenge lies in determining the optimal parameters—the spatial offsets and weights—for a sequence of sparse kernels to accurately reconstruct a target. Manually designing these parameters or using traditional, non-differentiable methods is a formidable task.

To overcome this, our key contribution is to frame the decomposition as a differentiable optimization problem. This enables the simultaneous end-to-end learning of all sparse kernel parameters across all layers. We define the complete set of these learnable parameters as $\Theta = \{(\mathbf{o}_{l,i}, w_{l,i})\}_{l=1,i=1}^{L,N_l}$, which includes the offsets and weights for N_l samples in each of the L layers.

Our goal is to find the optimal parameters Θ^* by minimizing a loss function $\mathcal{L}$ that measures the discrepancy between our approximation and the target kernel:

$$\begin{aligned} \Theta^* &= \arg\min_{\Theta} \mathcal{L}(K_{target}, F_{approx}(\Theta)), \\ F_{approx}(\Theta) &= K_{s,1} * K_{s,2} * ... * K_{s,L}, \end{aligned} \tag{4}$$

where K_{target} is the desired dense filter and $F_{approx}(\Theta)$ is the composite kernel formed by the convolution of the learned sparse kernels.

Learnable Parameter Our optimization strategy treats the offsets and weights of each sample as independent, learnable parameters. Specifically, for each layer l and for each of the N_l sampling points within it, we simultaneously optimize both the 2D offset vector $\mathbf{o}_{l,i}$ and its corresponding scalar weight $w_{l,i}$.

The complete set of learnable parameters for the entire model, denoted by Θ, is therefore the collection of all such offset-weight pairs:

$$\Theta = \bigcup_{l=1}^{L} \{(\mathbf{o}_{l,j}, w_{l,j})\}_{j=1}^{N_l}. \tag{5}$$

Initialization A robust parameter initialization is crucial for the stable convergence of the optimization. Heuristic methods, such as Kawase (Kawase, 2003) and Dual Filtering (Martin et al., 2015), have fixed schemes tailored to specific filter types; however, a general approach is required for arbitrary target kernels of different sizes.

To address this, we propose a radial initialization strategy. The core idea is to initialize the sampling points in each layer to be uniformly distributed on the circumference of a circle, with the radius of this circle increasing linearly with the layer index. This progressive expansion ensures that the

effective receptive field of the composite kernel grows with each subsequent layer, making the initial configuration capable of spanning a large-area target kernel from the outset. The radius for layer l, denoted r_l, is governed by a step size Δ_r derived from the target kernel's spatial extent and the total number of layers L (see Appendix for derivation). The corresponding weights in each layer are initialized uniformly.

This initialization is formally defined as:

$$\begin{aligned} r_l &= l \cdot \Delta_r && \text{for } l = 1, \ldots, L, \\ \mathbf{o}_{l,i} &= \left(r_l \cos\left(\frac{2\pi i}{N_l}\right), r_l \sin\left(\frac{2\pi i}{N_l}\right) \right) && \text{for } i = 1, \ldots, N_l, \\ w_{l,i} &= \frac{1}{N_l}. \end{aligned} \tag{6}$$

4.2 Sparse Sampling of Arbitrary Kernel

A common way to initialize filter offsets is by sampling random positions within a local neighborhood. While this approach is general, it often traps the optimization in poor local optima, especially for kernels with complex or non-convex shapes.

Our method decomposes a dense kernel into a series of sparse ones. The first of these, $K_{s,1}$ (Eq. 4), has the greatest influence on the final filtered output, so its initialization is critical. A simple improvement over purely random sampling is to confine samples to the minimal bounding box of the kernel's non-zero pixels. This ensures most samples fall near the target shape, but it is still inefficient for non-convex kernels, whose bounding boxes can contain large empty regions.

To overcome this limitation, we propose a more sophisticated initialization strategy leveraging rejection sampling. Instead of drawing samples from the kernel's bounding box, our method samples directly from the support of the kernel, i.e., its non-zero locations. We first quantify the effective sampling area, denoted by S, as the count of these non-zero pixels. A sampling radius r is subsequently derived based on the desired number of samples, N_s:

$$r = \sqrt{\frac{S}{N_s \cdot \pi}}. \tag{7}$$

The detailed procedure is provided in the appendix. This approach ensures that the initial offsets for the first sparse kernel provide a high-fidelity approximation of the target shape. By constraining the sampling to relevant regions, this method effectively circumvents the problem of vanishing gradients and prevents the optimization from converging to poor local optima.

4.3 Spatially Varying Filtering

Next, we propose a decomposition method for spatially varying filtering.

Spatially varying filtering generalizes convolution by applying a unique filter at each pixel (x, y). The filter's properties—such as its blur radius, orientation, or shape—are determined by a corresponding value $P(x, y)$ from a parameter map. The core challenge lies in efficiently synthesizing and applying these unique per-pixel kernels.

Conventional methods for spatially varying filtering are often impractical. Computing dense kernels on-the-fly (Wang et al., 2023) is prohibitively slow, while pre-computing them (Kovesi, 2010) demands excessive memory, rendering both approaches unsuitable for modern parallel hardware. More efficient techniques (Leimkühler et al., 2018) gain speed by restricting filters to simple analytical models, such as a Gaussian. This approach, however, lacks the expressiveness to represent complex, non-convex point spread functions (PSFs).

We observe that the cost of generating or storing spatially varying kernels by previous methods scales linearly with image resolution. To address this significant overhead while still leveraging expressive, sparsely optimized kernels, we introduce *Filter-Space Interpolation*, a method that decouples kernel computational complexity from image size.

Our spatially varying filtering is built on an ordered set of M basis sparse filters, which discretely sample a continuous, one-dimensional space $\mathcal{F}$ of filters. Each basis filter, f_k, corresponds to a scalar parameter p_k (with $p_1 < p_2 < \cdots < p_M$) and consists of a unique set of N sampling offsets and weights. This design allows our basis to represent a wide range of filter behaviors across the parameter space, from applying arbitrary linear transformations to a kernel to simply varying the standard deviation (σ) of a Gaussian. We define the basis as:

$$\mathcal{F} = \{f_k(p_k) \mid k = 1, \ldots, M\}, \quad \text{where} \quad f_k = \{(\mathbf{o}_{ki}, w_{ki})\}_{i=1}^{N} \tag{8}$$

We divide the approach into an offline pre-computation stage and a runtime inference stage. In the offline stage, we optimize each basis filter f_k individually to represent the ideal filter effect at its parameter value p_k.

At runtime, we synthesize a unique sparse filter for each pixel (x, y), which is guided by a per-pixel parameter map, P. From the parameter value at each coordinate, $P(x, y)$, we determine a corresponding vector of M interpolation weights, $\boldsymbol{\alpha}(x, y) = (\alpha_1, \ldots, \alpha_M)$. These weights specify how to blend a compact set of basis filters, $\{f_k\}_{k=1}^{M}$, to reconstruct the final filter instance.

The final sparse filter for a given pixel, $f(x, y)$, is synthesized as a direct convex combination of the basis filters:

$$f(x, y) = \sum_{k=1}^{M} \alpha_k(x, y) \cdot f_k, \tag{9}$$

subject to the constraint that $\sum_{k=1}^{M} \alpha_k(x, y) = 1$ and $\alpha_k(x, y) \geq 0$.

By directly interpolating basis-filter offsets and weights, we sidestep the costly on-the-fly generation of kernels from analytical functions. This reduces the computational overhead of spatially varying kernel synthesis to a minimal set of parallelizable multiply-add operations. Furthermore, the interpolatable nature of our basis filters makes the entire set highly compressible, allowing us to significantly reduce the memory footprint required to achieve a wide range of expressive effects while offering flexible control over the quality-performance trade-off.

4.4 Implementation Details

Training Process To ensure our learned filter parameters are generalized and not overfit to a specific dataset, we adopt an image-agnostic optimization strategy. We leverage a core principle of Linear Shift-Invariant (LSI) systems (Goodman, 2005): a filter is fully characterized by its impulse response.

First, we synthesize the effective kernel of our multi-pass filter, F_θ, by applying it to a discrete Dirac delta function, δ. The resulting output is the synthesized impulse response, K_{syn}. The impulse δ is an image with a single non-zero pixel at its center coordinate $\mathbf{c}$:

$$K_{\text{syn}} = F_\theta(\delta), \quad \text{where} \quad \delta[\mathbf{n}] = \begin{cases} 1 & \text{if } \mathbf{n} = \mathbf{c} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Here, θ represents the learnable parameters of our filter and $\mathbf{n}$ denotes the discrete pixel coordinates.

Loss Design Second, we define our loss function, $\mathcal{L}$, as the Charbonnier L1 loss $\mathcal{C}$ (Charbonnier et al., 1994) between the synthesized kernel K_{syn} and a target kernel K_{tgt}:

$$\mathcal{L} = \mathcal{C}(K_{\text{syn}}, K_{\text{tgt}}). \tag{11}$$

This impulse-response-based supervision allows us to "collapse" the entire multi-layer filtering sequence into a single, equivalent kernel for direct and precise approximation of the target.

5 Experiments

In this section, we conduct a series of experiments to evaluate our differentiable kernel decomposition framework thoroughly. We first describe the experiment details and evaluation protocol in Section 5.1. Next, in Section 5.2, we assess our method's ability to approximate single, complex kernels, comparing it against state-of-the-art techniques. We extend this analysis to the more challenging task of spatially varying filtering in Section 5.3. To validate our specific design choices, we present a series of ablation studies in Section 5.4.

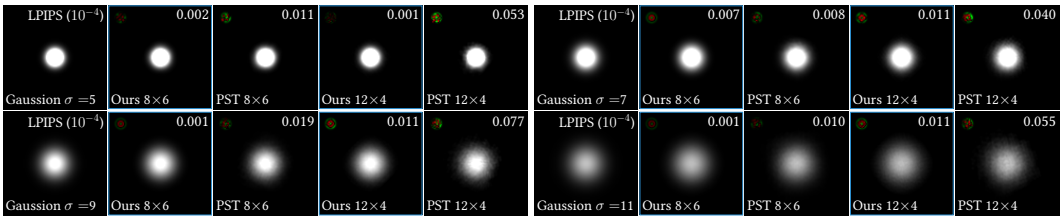

Figure 2: **Comparison of Gaussian kernel approximation with varying σ.** We compare our method against PST using two sparse configurations (8 layers × 6 samples and 12 layers × 4 samples). LPIPS scores appear in the top-right corner (lower is better).

5.1 Setup

Baselines. We compare our method against several baselines. For both single kernel and spatially varying filtering, we include a low-rank decomposition (LowRank) (McGraw, 2015) and the optimization-based method of Parallel Tempering (PST) (Schuster et al., 2020).

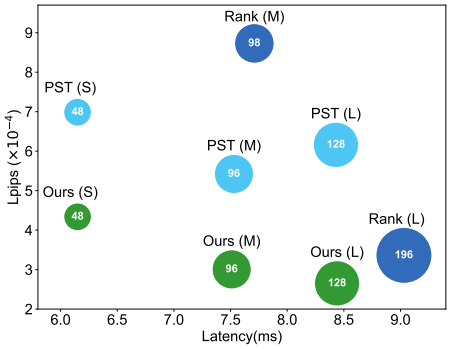

Figure 3: **Speed, accuracy, and samples comparison.** The figure plots quality against latency (lower is better for both). The size of each bubble represents the total sample count.

Datasets and Kernels. To evaluate the versatility of our method, we use a diverse set of target kernels and images. This set includes standard analytical shapes, such as Gaussian kernels (with σ values from 5 to 11). To assess performance on more complex targets, we additionally use a suite of arbitrary kernels comprising simple geometric primitives (disks, rings), regular polygons (4-sided and 6-sided), non-convex shapes (a heart, a four-pointed star, and an ampersand symbol), more complex shapes (animal silhouettes), and optical PSFs (coma and spherical aberration). For the spatially varying filtering experiments, we use five high-resolution photographs selected to represent realistic scenarios with complex textures and both 1D and 2D spatial variations.

Implementation and Evaluation Metric. We implement our methods in PyTorch and perform all optimization on a single GPU with 24 GB of memory, offering computational power comparable to an NVIDIA RTX 4090. For all configurations of kernels and layers, we use the same Adam optimizer with a learning rate linearly decayed from 1×10^{-3} to 1×10^{-4}. We use 1000 optimization steps per kernel for our method. For comparison, we run the PST algorithm for 10,000 iterations with 10 parallel candidates, for a total of 100,000 optimization steps. For the LowRank method, we utilize decompositions with ranks 1,2 and 3, chosen to maintain a comparable number of samplings.

For runtime analysis, we benchmark our approach on a representative mobile device equipped with a Qualcomm Snapdragon 8 Gen 3 SoC, and report latency in milliseconds (ms). We evaluate both numerical fidelity and perceptual similarity using Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), and FLIP-LDR (Andersson et al., 2020). Higher values indicate better performance for PSNR, and lower values are better for LPIPS and FLIP-LDR.

5.2 Single Kernel

Fig. 3 shows that our method consistently achieves a superior balance between reconstruction quality and inference speed compared to all other approaches. For our method and PST, the 'S', 'M', and 'L' correspond to total sample counts of 48 (12×4), 96 (24×4), and 128 (32×4), respectively. The LowRank's 'M' and 'L' use 98 (49×2) and 196 (49×4) parameters.

Figure 4: **Comparison of Single kernel approximation.** Compared to baselines, SVD-based decomposition (LowR.) and Parallel Simulated Tempering (PST), our approach (blue) better preserves sharp features on non-convex targets, resulting in lower LPIPS scores (lower is better).

Next, we present a comparison of Gaussian kernel approximation with varying standard deviations σ in Fig. 2. In a 6-layer, 8-sample (8×6) configuration, our method achieves high-fidelity results with low perceptual error, whereas PST exhibits visible noise and artifacts. This performance gap widens in a sparser 12×4 setup. As σ increases, PST's approximation degrades severely, while our result remains visually coherent and maintains a substantially lower LPIPS error. These results demonstrate that our gradient-based optimization is more robust than stochastic search methods PST, consistently finding stable solutions even in challenging, sparse configurations.

Our method's robustness extends beyond Gaussian kernels to the more general case of arbitrary single-kernel filters, as shown in Fig. 4. Our method achieves superior visual fidelity, accurately preserving structures in both simple and complex shapes. In contrast, LowRank produces blocky artifacts and PST yields noisy results that degrade further at low sample counts. These visual advantages are confirmed quantitatively, as our method obtains the lowest LPIPS error across all tests, often by a significant margin. Note that our method is also far more efficient, requiring only 1/100th the iteration steps of PST.

5.3 Spatially Varying Kernel

We present three spatially varying filtering examples in Fig. 5. The first is a 1D spatially varying blur that uses a pseudo-depth map to simulate a tilt-shift camera effect. The other two are 2D anisotropic effects: a rotational bokeh blur and a radial motion blur, both controlled by two parameters—blur intensity and local blur angle.

Our method achieves results that are nearly indistinguishable from the ground truth. As shown in the red and green insets, our method faithfully reproduces the complex structure of the ground-truth (GT) kernels. In contrast, Parallel Simulated Tempering (PST) and Low-Rank Decomposition (LowRank) either introduce noise (PST) or oversmooth the kernels (LowRank), and both fail to recover the correct kernel shapes, while direct use of GT kernels is prohibitively slow. Quantitatively, our method achieves the highest PSNR among all methods while maintaining real-time performance.

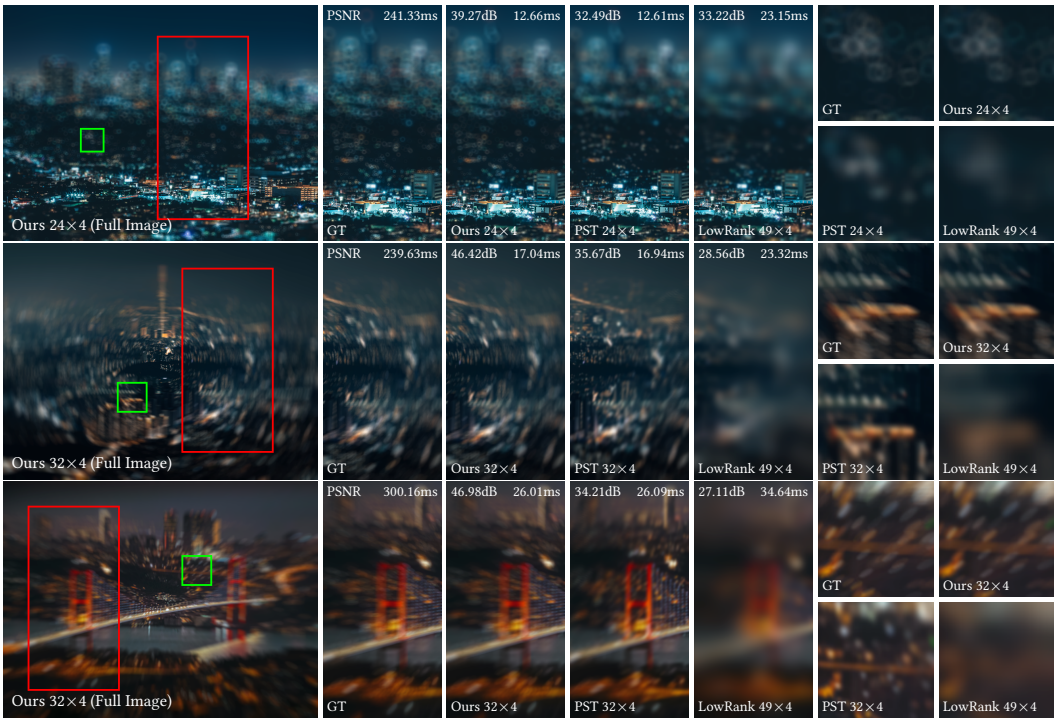

Figure 5: **Visual comparison of diverse spatially varying (SV) effects.** We evaluate three SV configurations: 1D tilt-shift blur (top), 2D rotational blur (middle), and 2D radial motion blur (bottom). We compare our method against Parallel Simulated Tempering (PST) and Low-Rank Decomposition (LowRank).

This performance difference stems from how well each method's base kernels handle filter-space interpolation. While all approaches use interpolation to generate the varying filter parameters, our optimization-based kernels are better conditioned for this process and appear to vary more linearly. Consequently, they interpolate smoothly to form sharp, complex patterns. PST's kernels, however, suffer from poor optimization quality, and interpolating between them simply produces more noise. Similarly, interpolating the basis kernels from LowRank's decomposition causes them to average into indistinct blurs rather than preserving the target structure.

5.4 Ablations

We conduct ablation studies to validate our main design choices, focusing on both initialization strategies and different layer configurations.

We first evaluate different initialization schemes across multiple kernels, as shown in Fig. 6. Both our method and Parallel Simulated Tempering (PST) benefit from the proposed Sparse Sampling (SS) initialization, which consistently outperforms the Increasing Radial (IR) initialization, while the Random (Rand) initialization performs worst. Although SS accelerates convergence for both our method and PST, PST still requires more than 30× the number of iterations to converge compared with ours, and our final reconstruction quality is significantly higher.

We further study the influence of different configurations, varying the number of layers and the number of samples, as shown in Fig. 7. The convergence curves show that all configurations converge stably, and configurations with more samples and layers tend to achieve higher quality. Compared with PST, our method delivers more consistent behavior and better quality across all tested configurations.

For additional results, please refer to the Appendix, which includes ablations on Gaussian kernels with fewer samples and quantitative evaluations of initialization and regularization strategies on arbitrary kernels.

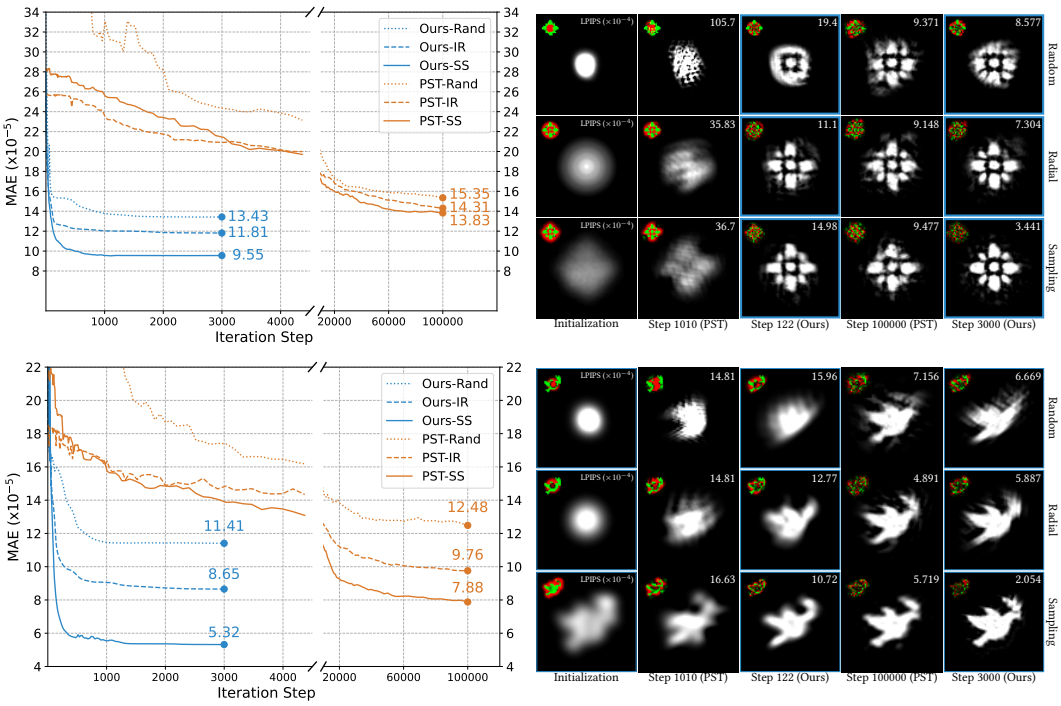

Figure 6: **Ablation of initialization strategies on the *Flower* and *Dove* kernel.** We evaluate both our method and Parallel Simulated Annealing (PST) combined with three initialization schemes: Random (Rand), Increasing Radial (IR), and Sparse Sampling (SS).

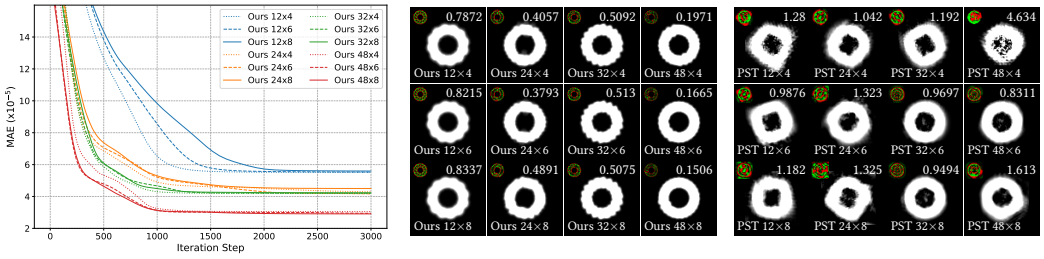

Figure 7: **Ablation results for various configurations of samples and layers on Ring kernel.**

6 Discussion and Conclusion

We introduced a differentiable framework that recasts the challenging problem of approximating large, complex convolution kernels as an end-to-end optimization task. Our approach robustly handles a wide variety of kernels—from simple Gaussians to complex, non-convex forms—and converges to high-fidelity solutions far more efficiently than prior methods. We extend this with filter-space interpolation, enabling complex, spatially-varying effects with minimal per-pixel overhead. This work opens several promising avenues for future research, including multi-dimensional parameter maps for simultaneous control over kernel attributes and the use of neural architecture search to discover hardware-optimized filter decompositions. In conclusion, our work provides a practical, high-performance solution for advanced image filtering in real-time applications like computational photography, while its fully differentiable nature allows it to serve as a trainable layer within modern deep learning pipelines.

REFERENCES

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. Flip: A difference evaluator for alternating images. *Proc. ACM Comput. Graph. Interact. Tech.*, 3(2):15–1, 2020.

Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, volume 2, pp. 168–172. IEEE, 1994.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company publishers, 2005.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

Masaki Kawase. Frame buffer postprocessing effects in double-steal (wrechless). In *Game Developers Conference 2003, 3*, 2003.

Peter Kovesi. Fast almost-gaussian filtering. In *2010 International conference on Digital image computing: Techniques and applications*, pp. 121–125. IEEE, 2010.

Thomas Leimkühler, Hans-Peter Seidel, and Tobias Ritschel. Laplacian kernel splatting for efficient depth-of-field and motion blur synthesis or reconstruction. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

Yang Li, Bobo Yan, Jianxin Hou, Bingyang Bai, Xiaoyu Huang, Canfei Xu, and Limei Fang. Unet based on dynamic convolution decomposition and triplet attention. *Scientific Reports*, 14(1):271, 2024.

Yunsheng Li, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Ye Yu, Lu Yuan, Zicheng Liu, Mei Chen, and Nuno Vasconcelos. Revisiting dynamic convolution via matrix decomposition. *arXiv preprint arXiv:2103.08756*, 2021.

Cewen Liu, Mengyao Sun, Nanxun Dai, Wei Wu, Yanwen Wei, Mingjie Guo, and Haohuan Fu. Deep learning-based point-spread function deconvolution for migration image deblurring. *Geophysics*, 87(4):S249–S265, 2022.

Pinjun Luo, Guoqiang Xiao, Xinbo Gao, and Song Wu. Lkd-net: Large kernel convolution network for single image dehazing. In *2023 IEEE international conference on multimedia and expo (ICME)*, pp. 1601–1606. IEEE, 2023.

Sam Martin, Andrew Garrard, Andrew Gruber, Marius Bjorge, Renaldas Zioma, Simon Benge, and Niklas Nummelin. Moving mobile graphics. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336345. doi: 10.1145/2776880.2787664. URL https://doi.org/10.1145/2776880.2787664.

Tim McGraw. Fast bokeh effects using low-rank linear filters. *The Visual Computer*, 31(5):601–611, 2015.

Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2502–2510, 2018.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 670–679, 2017.

Kurniawan Nur Ramadhani, Rinaldi Munir, and Nugraha Priya Utama. Improving video vision transformer for deepfake video detection using facial landmark, depthwise separable convolution and self attention. *IEEE Access*, 12:8932–8939, 2024.

Parikshit Vishwas Sakurikar. Epsilon focus photography a study of focus defocus and depth of field.

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*, pp. 1–12. 2017.

Kersten Schuster, Philip Trettner, and Leif Kobbelt. High-performance image filters via sparse approximations. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3 (2):1–19, 2020.

Adrian Shajkofci and Michael Liebling. Spatially-variant cnn-based point spread function estimation for blind deconvolution and depth estimation in optical microscopy. *IEEE Transactions on Image Processing*, 29:5848–5861, 2020.

Chao Wang, Krzysztof Wolski, Xingang Pan, Thomas Leimkühler, Bin Chen, Christian Theobalt, Karol Myszkowski, Hans-Peter Seidel, and Ana Serrano. An implicit neural representation for the image stack: Depth, all in focus, and high dynamic range. Technical report, 2023.

Yi Wu, Yanyang Xu, Wenhao Zhu, Guojie Song, Zhouchen Lin, Liang Wang, and Shaoguo Liu. Kdlgt: A linear graph transformer framework via kernel decomposition approach. In *IJCAI*, pp. 2370–2378, 2023.

Zijin Wu, Xingyi Li, Juewen Peng, Hao Lu, Zhiguo Cao, and Weicai Zhong. Dof-nerf: Depth-of-field meets neural radiance fields. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1718–1729, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang. Decoupled dynamic filter networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6647–6656, 2021.

Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2482–2491, 2019.

A Zing. Extended binomial filter for fast gaussian blur. *Vienna, Austria*, 2010.