LEGENDREIMAGE: GENERALIZED FOURIER EXPANSION SPLATTING FOR IMAGE REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing 2D Gaussian image representations mainly scale their representations through the number and spatial allocation of primitives, while the function space represented by each individual primitive remains comparatively limited. We revisit primitive representation and investigate whether richer local function spaces can provide an additional dimension of representation design. We introduce *Generalized Fourier Expansion Splatting* (GFES), which views the signal carried by each primitive as a function over an intrinsic domain and parameterizes it through a truncated generalized Fourier expansion. This formulation separates the representation of the primitive-carried signal from its spatial localization: a spatial weighting function determines where the primitive contributes, while the function expansion determines how its signal varies over the corresponding domain. The basis is chosen according to the domain, measure, and function space of the represented signal. Following this principle, we instantiate GFES as LegendreImage by combining compact raised-cosine localization with a two-dimensional Legendre polynomial expansion to represent spatially varying color within individual primitives. Adaptive primitive allocation yields LegendreImage+. On DIV2K, LegendreImage+ surpasses GaussianImage++ from the 2.4M-parameter regime onward and reaches 51.83 dB at a 4M trainable-parameter budget, outperforming GaussianImage++ by 3.77 dB. These results show that the per-primitive function space is an important dimension of representation design, complementary to scaling through the number and spatial allocation of primitives.

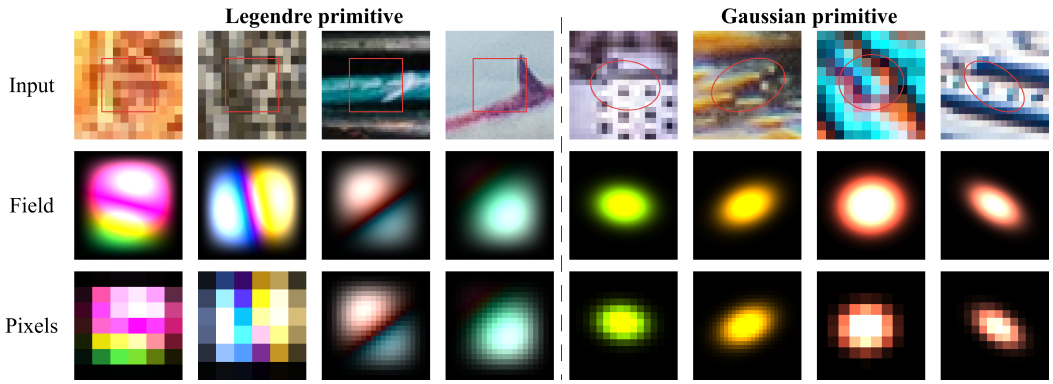


Figure 1: **Local structure represented by individual Legendre and Gaussian primitives.** Each column shows one learned primitive and its corresponding local region in the target image, with the red outline indicating the spatial footprint of the selected primitive. *Field* visualizes the continuous primitive field; for Legendre primitives, we display $\mathrm{clip}(|\Phi_i|, 0, 1)$. *Pixels* show the rasterized field on the discrete pixel grid. Legendre primitives represent spatially varying local patterns within a compact support, whereas Gaussian primitives represent a constant color modulated by a Gaussian spatial profile.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Despite the success of 2D Gaussian Splatting image representation Zhang et al. (2024); Li et al. (2026); Zhu et al. (2025), a fundamental tension remains: improving the representation capacity of splatting systems has mainly relied on primitive count and spatial allocation, while the design of individual primitive representations remains comparatively underexplored. Existing methods typically adopt Gaussian kernels as primitives, benefiting from their smoothness, locality, and efficient differentiable rendering. For direct 2D image representation, however, primitives are defined and rendered directly in the image plane, without the 3D-to-2D projection considerations that make Gaussian primitives particularly convenient in 3D splatting. This removes an important structural constraint on the primitive function and provides greater freedom to reconsider its representation. Meanwhile, recent studies have explored extensions and alternatives to Gaussian primitives Hamdi et al. (2024); Wurster et al. (2024); Zhang et al. (2025a); Yu et al. (2025); Huang et al. (2025), suggesting that the functional form of primitives itself can affect reconstruction quality. However, a broader question remains open: how should a primitive be designed to effectively balance local representation capacity, parameter efficiency, and spatial locality?

Existing 2D Gaussian primitives can be viewed as a combination of spatial localization and local signal representation. Each primitive defines its spatial influence through a Gaussian kernel parameterized by its position and covariance, while its appearance is typically represented by a low-dimensional signal such as a constant color. This formulation is efficient and stable, but constrains the spatial variation of an individual primitive to the Gaussian profile. As a result, local variations such as asymmetric intensity changes, multiple local extrema, sign-changing patterns, and higher-order spatial structures cannot be directly represented by a conventional Gaussian primitive. Instead, such structures are typically approximated through the overlap and combination of multiple Gaussian primitives Zhang et al. (2024); Li et al. (2026); Zhu et al. (2025). Increasing the number of primitives enlarges the capacity of the overall representation through composition, but leaves the function class of each individual primitive unchanged.

Our starting point is the representation of appearance in 3D Gaussian Splatting Kerbl et al. (2023). Rather than restricting each primitive to a single RGB value, 3DGS represents its view-dependent color as a function of viewing direction through a truncated spherical-harmonic expansion. Since spherical harmonics form an orthogonal basis for functions on the sphere, this representation can be interpreted as a truncated generalized Fourier expansion Kreyszig (2011) of the signal carried by each primitive over its directional domain. This observation suggests a broader view of splatting primitives: the signal associated with an individual primitive need not be a constant attribute, but can instead be represented as a function over an intrinsic signal domain.

Let i index the splatting primitives and let $\boldsymbol{\xi}$ denote a coordinate in the signal domain. We represent the primitive-carried signal as

$$\boxed{Q_i(\boldsymbol{\xi}) = \sum_{k \in \mathcal{K}} c_{ik} \psi_k(\boldsymbol{\xi}),} \tag{1}$$

where $\{\psi_k\}_{k \geq 0}$ is an orthogonal basis of the corresponding Hilbert space, chosen according to the domain, measure, and structure of the represented signal, $\mathcal{K}$ denotes the retained finite set of basis indices, and c_{ik} are the corresponding expansion coefficients. We refer to this representation principle as *Generalized Fourier Expansion Splatting* (GFES).

Spatial localization is handled separately. Let $\mathbf{r}$ denote a spatial coordinate and $W_i(\mathbf{r})$ the spatial weighting function of primitive i. A general GFES primitive can be written as

$$\Phi_i(\mathbf{r}, \boldsymbol{\xi}) = W_i(\mathbf{r}) Q_i(\boldsymbol{\xi}). \tag{2}$$

The localization domain of W_i and the signal domain of Q_i need not coincide, allowing the primitive-carried signal representation to be treated as a distinct dimension of splatting design.

We instantiate GFES as LegendreImage for direct 2D image representation. Since the primitive-carried color signal is defined over a bounded local spatial domain, we consider the standard Lebesgue measure and adopt a degree-ordered polynomial hierarchy orthogonal under the corresponding inner product, which uniquely determines the Legendre family up to normalization. Combining a two-dimensional Legendre expansion with a compact raised-cosine window provides explicitly bounded spatial influence while allowing each primitive to represent spatially varying local color structures within its support.

In summary, our contributions are threefold. First, we introduce GFES, which represents the signal carried by each primitive as a function expanded in an orthogonal basis associated with its natural domain and function space. This formulation makes the per-primitive signal space an explicit dimension of splatting representation design. Second, we instantiate GFES as LegendreImage for direct 2D image representation. By modeling local color variation over a bounded spatial domain with a two-dimensional Legendre expansion and compact raised-cosine localization, LegendreImage enables individual primitives to represent spatially varying local signals within explicitly bounded support. Third, extensive experiments show that allocating representation capacity to the per-primitive function space can substantially improve reconstruction quality under a fixed trainable-parameter budget. In particular, low-order local expansions provide an effective trade-off between per-primitive expressiveness and primitive density.

2 Related Work

Implicit Neural Image Representations. Implicit neural representations (INRs) model visual signals as continuous functions mapping spatial coordinates to signal values. Their ability to capture high-frequency content has been improved through Fourier feature mappings Tancik et al. (2020), periodic activations in SIREN Sitzmann et al. (2020), and Gabor wavelet activations in WIRE Saragadam et al. (2023). Beyond spectral parameterization, localized and multiresolution structures further improve representation efficiency: Instant-NGP Müller et al. (2022) employs multiresolution hash encoding, NeuRBF Chen et al. (2023) introduces adaptive radial basis functions, ACORN Martel et al. (2021) adaptively partitions the signal domain into multiscale blocks, and LIIF Chen et al. (2021) represents images through local latent features queried by an implicit decoder. INRs have also been applied to image compression, where COIN Dupont et al. (2021) stores an image-specific network and COIN++ Dupont et al. (2022) improves efficiency through meta-learned modulations. These approaches encode signals through neural networks or learned features. In contrast, GFES directly parameterizes the signal carried by each splatting primitive through an orthogonal function expansion, without relying on a neural network or learned decoder.

Splatting-based Image Representations. Recent works have adapted Gaussian Splatting to direct 2D image representation. GaussianImage Zhang et al. (2024) represents images using 2D Gaussian primitives parameterized by position, covariance, and color, while Image-GS Zhang et al. (2025b) further explores content-adaptive allocation and progressive optimization. Related multiscale strategies have been investigated in MINER Saragadam et al. (2022), which hierarchically decomposes signals across scales using implicit networks. For Gaussian-based representations, LIG Zhu et al. (2025) introduces a Level-of-Gaussian representation for large-image fitting, GaussianImage++ Li et al. (2026) improves parameter utilization through distortion-driven densification and content-adaptive filtering, Instant GaussianImage Zeng et al. (2025) employs learned initialization and adaptive Gaussian counts, and entropy-aware Gaussian representation Chen et al. (2026) allocates primitives according to spatially varying image complexity. These methods primarily improve the allocation, initialization, optimization, or multiscale organization of Gaussian primitives.

More recent works reconsider the primitive function itself. GES Hamdi et al. (2024) replaces the Gaussian profile with a generalized exponential kernel, while Gabor Splatting Wurster et al. (2024) and WIPES Zhang et al. (2025a) introduce Gabor- and wavelet-based primitives with joint spatial-frequency localization. 2D Gaussian-Hermite Splatting Yu et al. (2025) augments Gaussian primitives with Hermite-polynomial terms for 3D reconstruction, and DRK Huang et al. (2025) learns deformable radial kernels with flexible geometric footprints. These studies demonstrate the importance of primitive expressiveness. GFES takes a function-space view of the signal carried by each primitive, representing it through an orthogonal expansion associated with the signal domain, rather than prescribing a particular expressive kernel family.

GFES is mathematically related to classical localized function approximation, where local polynomial models are combined through spatial weighting, as in Shepard interpolation Shepard (1968) and moving least squares Lancaster & Salkauskas (1981); Levin (1998). GFES adopts this structure from the perspective of splatting representation: the expanded function is the signal carried by each primitive, whose coefficients and spatial allocation are jointly optimized for image representation.

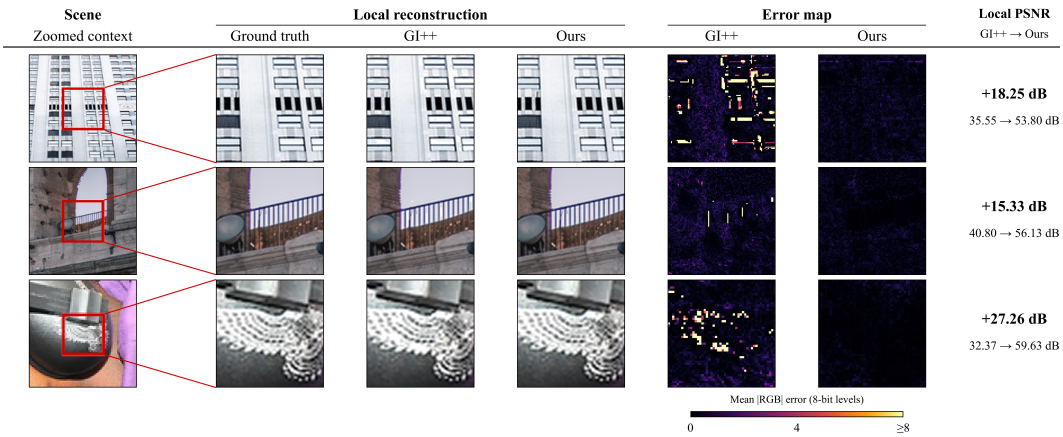


Figure 2: Local reconstruction comparison between GaussianImage++ and LegendreImage+ on representative DIV2K regions under the 4M trainable-parameter budget. For each example, we show the image context, ground-truth crop, reconstructed crops, and corresponding error maps. The error maps visualize the mean absolute RGB reconstruction error in 8-bit intensity levels using a shared scale. The rightmost column reports the local PSNR of GaussianImage++ and LegendreImage+ for the displayed crop, with the bold value indicating their PSNR difference.

3 METHOD

Given a spatial coordinate $\mathbf{r} = (x, y)$, we define the primitive-specific signal-domain coordinate $\boldsymbol{\xi}_i(\mathbf{r}) = (u_i, v_i)$ by normalizing $\mathbf{r}$ relative to the primitive center and support size; within the primitive support, $\boldsymbol{\xi}_i(\mathbf{r}) \in [-1, 1]^2$. Accordingly, LegendreImage evaluates both the local signal expansion and the compact window at $\boldsymbol{\xi}_i(\mathbf{r})$.

3.1 LEGENDRE PRIMITIVE CONSTRUCTION

For each primitive, we normalize the local spatial coordinates to $[-1, 1]$ and use the standard Lebesgue inner product $\langle f, g\rangle = \int_{-1}^{1} f(t)g(t)\,dt$. The polynomial sequence satisfying $\deg P_n = n$ and mutual orthogonality under this inner product is the Legendre family, unique up to normalization.

For $x \in [-1, 1]$, if f is square-integrable, it admits a Legendre polynomial expansion in the L^2 sense:

$$\boxed{\int_{-1}^{1} |f(x)|^2\,dx < \infty \quad \Longrightarrow \quad f(x) \sim \sum_{n=0}^{\infty} a_n P_n(x)} \tag{3}$$

where $P_n(x)$ denotes the Legendre polynomial of degree n, forming a complete orthogonal basis for the Hilbert space $L^2([-1, 1])$ Szegő (1975).

The first few orders are $P_0(x) = 1, P_1(x) = x, P_2(x) = \frac{1}{2}(3x^2 - 1)$.

For a two-dimensional local color signal, we denote by $Q(u, v)$ the local color function represented within a primitive before spatial windowing. We construct a two-dimensional Legendre basis from products of one-dimensional Legendre polynomials and truncate it by total degree:

$$Q(u, v) = \sum_{\substack{m+n \le K \\ m,n \ge 0}} c_{mn} P_m(u) P_n(v), \qquad (u, v) \in [-1, 1]^2, \tag{4}$$

where c_{mn} are the Legendre coefficients and K denotes the maximum total basis order. At first order, this construction spans the affine local function space $\mathrm{span}\{1, u, v\}$, with the Legendre basis providing a domain-consistent orthogonal parameterization of this space.

To spatially localize the expansion, we apply a two-dimensional compact raised-cosine window:

$$W(u,v) = \begin{cases} \frac{1}{4}\left(1+\cos(\pi u)\right)\left(1+\cos(\pi v)\right), & |u|<1,\ |v|<1, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

For primitive i, centered at (x_i, y_i) with support half-size R_i, a pixel (x, y) is mapped to the normalized local coordinates $u_i = \frac{x-x_i}{R_i}, v_i = \frac{y-y_i}{R_i}$. The contribution of primitive i at pixel (x, y) is defined as

$$\mathbf{\Phi}_i(x,y) = W(u_i,v_i)\left[\mathbf{c}_{i,00} + \begin{pmatrix}1\\1\\1\end{pmatrix} \sum_{m,n\geq 0, mn\neq 0}^{m+n\leq K} c_{i,mn} P_m(u_i) P_n(v_i)\right], \tag{6}$$

where $\mathbf{c}_{i,00} = (c_{i,00}^{(R)}, c_{i,00}^{(G)}, c_{i,00}^{(B)})^\top$ denotes the channel-specific zeroth-order RGB coefficients. The higher-order coefficients $c_{i,mn}$ are shared across the three color channels.

In our main implementation, we use a first-order expansion ($K = 1$), for which the only higher-order terms are $P_1(u)P_0(v)$ and $P_0(u)P_1(v)$. Therefore, each primitive contains 7 trainable parameters:

$$\Theta_i = \left\{x_i,\ y_i,\ c_{i,00}^{(R)},\ c_{i,00}^{(G)},\ c_{i,00}^{(B)},\ c_{i,10},\ c_{i,01}\right\}. \tag{7}$$

Although a Legendre primitive can in principle be extended to arbitrarily high orders with an unbounded number of expansion coefficients, concentrating excessive capacity within a single primitive is not necessarily beneficial. We investigate higher-order expansions in Appendix C.

As shown in Fig. 1, the learned first-order Legendre primitives can represent spatially varying local color structures within a single compact support, whereas a Gaussian primitive is restricted to a fixed RGB direction modulated by its spatial kernel.

Finally, the rendered color at pixel (x, y) is obtained by aggregating all primitives whose compact supports cover that pixel:

$$\hat{\mathbf{I}}(x,y) = \begin{cases} \dfrac{\sum_{i\in\mathcal{A}(x,y)} \mathbf{\Phi}_i(x,y)}{\sum_{i\in\mathcal{A}(x,y)} W(u_i,v_i)}, & \sum_{i\in\mathcal{A}(x,y)} W(u_i,v_i) > 0, \\ \mathbf{0}, & \sum_{i\in\mathcal{A}(x,y)} W(u_i,v_i) = 0, \end{cases} \tag{8}$$

where $\mathcal{A}(x, y)$ denotes the set of active primitives at pixel (x, y).

3.2 Primitive Optimization and Adaptive Allocation

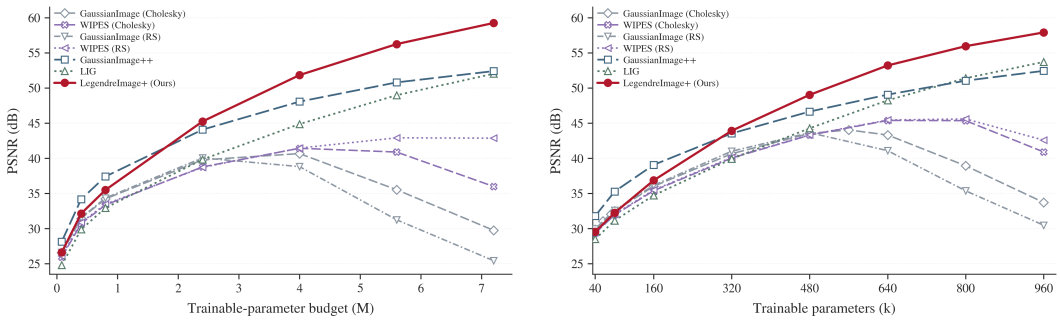
We use *LegendreImage* to denote the formulation with a fixed set of Legendre primitives throughout optimization. The primitives are initialized on a regular image-plane grid. The zeroth-order RGB coefficients are initialized from the mean color within each primitive support, while the higher-order coefficients are initialized to zero. The primitive centers and expansion coefficients are jointly optimized for image reconstruction, with the center positions parameterized as bounded offsets from their initialized locations. The primitive set remains fixed throughout optimization.

We further extend LegendreImage with adaptive primitive allocation and denote the resulting method as *LegendreImage+*. During training, candidate locations are selected from local maxima of the current reconstruction residual. Each candidate primitive is locally fitted and ranked according to the reconstruction gain it brings to the current representation, and the candidates with the largest gains are inserted. The support half-size of newly inserted primitives is determined according to the current primitive density and gradually decreases as the representation becomes denser. Once assigned, the support half-size R_i remains fixed for that primitive. After the target primitive count is reached, fixed-budget relocation is used to redistribute existing primitives according to the reconstruction residual. More details are provided in Appendix B.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate image representation on the DIV2K Agustsson & Timofte (2017) validation set and the Kodak dataset Eastman Kodak (1993). DIV2K contains 100 validation images with an



(a) DIV2K (b) Kodak

Figure 3: Reconstruction PSNR versus trainable-parameter budget on (a) DIV2K and (b) Kodak. LegendreImage+ trails GaussianImage++ at low budgets but surpasses it from 2.4M parameters on DIV2K and 320k on Kodak, with larger gains at higher budgets.

average resolution of approximately 1450×1978, while Kodak contains 24 images at 768×512 resolution. Each image is represented independently by optimizing its own set of primitives, and results are averaged over all images in each dataset. Additional experiments on DIV8K Gu et al. (2019) at a 60M trainable-parameter budget are reported in Appendix E to evaluate representation quality at substantially higher resolutions.

Evaluation Metrics. We use PSNR as the primary metric for reconstruction fidelity and compare methods under matched trainable-parameter budgets: seven budgets from 0.08M to 7.2M on DIV2K and eight budgets from 40k to 960k on Kodak. The parameter budget specifies the target number of trainable scalars in the final representation. We additionally report MS-SSIM, training time, evaluation time, and rendering speed in frames per second (FPS) in Appendix E.

Implementation Details. We optimize all LegendreImage+ models for 30,000 iterations using Adam Kingma & Ba (2014) with an MSE reconstruction loss. The learning rates for the zeroth-order coefficients, higher-order coefficients, and primitive centers are initialized to 0.1, 0.02, and 0.05, respectively. We use a StepLR scheduler that decays the learning rates by a factor of 0.5 every 6,000 iterations and set the Adam epsilon to 10^{-9}. No additional range regularization is applied during training. LegendreImage+ is implemented in PyTorch with a CUDA rasterizer, and all our experiments are conducted on an NVIDIA A100 40GB PCIe GPU.

4.2 Comparison with Splatting-based Image Representations

As shown in Fig. 3, GaussianImage++ performs best in the low-budget regime, whereas LegendreImage+ overtakes it at 2.4M parameters and the margin continues to grow as the budget increases. At the 4M budget, LegendreImage+ reaches 51.83 dB, outperforming GaussianImage++ and LIG by 3.77 dB and 6.97 dB, respectively. This budget-dependent crossover highlights the per-primitive function space as a distinct dimension of representation design beyond primitive count and spatial allocation. A similar budget-dependent crossover is observed on Kodak. LegendreImage+ trails GaussianImage++ at lower budgets but surpasses it from 320k parameters onward. At 480k parameters, it achieves 49.02 dB, compared with 46.64 dB for GaussianImage++. These results show that the benefit of increasing the per-primitive function space is also evident on lower-resolution images.

Table 7 also shows that the improved reconstruction quality comes with additional computation. At the 4M budget, LegendreImage+ requires 566.9 s of training and renders at 238.0 FPS, compared with 231.2 s and 545.7 FPS for GaussianImage++.

Fig. 2 further shows that LegendreImage+ produces smaller local reconstruction errors than GaussianImage++ across representative structured regions, consistent with the corresponding local PSNR values.

Additional results on DIV8K in Table 5 show that the PSNR advantage extends to substantially higher-resolution images.

Table 1: PSNR (dB, ↑) at representative trainable-parameter budgets on DIV2K and Kodak. The best and second-best results at each budget are marked in bold and underlined, respectively. Full results, including training time and rendering speed, are provided in Appendix E.

Parameter Budget	**GaussianImage** (Cholesky)	**GaussianImage** (RS)	**WIPES** (Cholesky)	**WIPES** (RS)	**LIG**	**Gaussian Image++**	**LegendreImage+** (Ours)
				DIV2K			
0.4M	31.45	31.55	30.71	30.86	29.87	**34.17**	<u>32.13</u>
4.0M	40.66	38.81	41.42	41.44	44.86	<u>48.06</u>	**51.83**
7.2M	29.74	25.42	35.98	42.85	52.02	<u>52.40</u>	**59.24**
				Kodak			
160k	36.01	36.15	35.38	35.48	34.72	**39.06**	<u>36.85</u>
480k	43.53	43.64	43.29	43.32	44.27	<u>46.64</u>	**49.02**
960k	33.71	30.47	40.90	42.58	<u>53.70</u>	52.43	**57.89**

Table 2: Ablation studies on DIV2K under a fixed 4M trainable-parameter budget. All values are PSNR in dB. Ours uses first-order Legendre expansions with RGB-shared first-order coefficients and a raised-cosine window.

Basis and localization		*Polynomial order*		*Compact window*			*First-order RGB coefficients*	**Ours**
GaussianImage++ with Hermite	LIG with Hermite	Zeroth	Second	Cubic smoothstep	Triangular	Parabolic	Independent	**LegendreImage+**
43.86	41.07	47.06	51.21	51.77	51.01	50.63	45.52	**51.83**

5 Ablation Study

5.1 Basis Functions and Localization

Consider a general local color signal represented by a one-dimensional function $f(x)$ over an unbounded domain. For $x \in \mathbb{R}$, if f is square-integrable with respect to the Gaussian measure, it admits a Hermite polynomial expansion in the corresponding weighted L^2 sense:

$$\boxed{\int_{-\infty}^{\infty} |f(x)|^2 e^{-x^2/2}\, dx < \infty \quad \Longrightarrow \quad f(x) \sim \sum_{n=0}^{\infty} b_n \,\mathrm{He}_n(x)} \tag{9}$$

where $\mathrm{He}_n(x)$ denotes the Hermite polynomial of degree n, forming a complete orthogonal basis for the Gaussian-weighted Hilbert space $L^2(\mathbb{R}, e^{-x^2/2}dx)$ Szegő (1975). This provides a natural counterpart to the Legendre expansion over the bounded domain $[-1, 1]$: Legendre polynomials are orthogonal under a standard Lebesgue measure on a bounded interval, whereas Hermite polynomials are orthogonal under a Gaussian measure over $\mathbb{R}$.

To examine different realizations of local signal expansion, we augment the Gaussian primitives of GaussianImage++ and LIG with first-order Hermite terms while retaining their original Gaussian spatial weighting. LegendreImage+ is evaluated using the corresponding first-order Legendre representation. All variants therefore introduce first-order local variation and are compared under the same 4M trainable-parameter budget.

As shown in Table 2, GaussianImage++ with Hermite and LIG with Hermite achieve 43.86 dB and 41.07 dB, respectively, whereas the first-order Legendre formulation reaches 51.83 dB. These results show that introducing additional local basis coefficients alone does not necessarily provide an effective representation under a fixed parameter budget. In particular, the Gaussian-weighted Hermite formulations are substantially less effective than the compact Legendre formulation in this setting. At first order, both Legendre and Hermite terms span linear local functions; the comparison therefore contrasts how the same-order local variation is realized under different localization mechanisms and underlying function spaces. The results suggest that the local function space and its spatial weighting should be designed jointly.

5.2 PRIMITIVE FUNCTION CAPACITY

The polynomial order directly controls the local function capacity of each primitive. Increasing the order enables a primitive to represent richer local variations, but also introduces additional coefficients per primitive. Under a fixed total parameter budget, this creates a trade-off between the expressive capacity of individual primitives and the number of primitives that can be allocated to represent the image.

We study this trade-off by varying the order of the Legendre representation while fixing the trainable-parameter budget to approximately 4M. The zeroth-order variant uses only the RGB constant term, resulting in 5 trainable parameters per primitive and $800{,}000$ primitives. The first-order variant additionally introduces two shared linear coefficients, resulting in 7 parameters per primitive and $571{,}428$ primitives. The second-order variant further introduces the quadratic terms, increasing the per-primitive parameter count to 10 and reducing the number of primitives to $400{,}000$.

As shown in Table 2, increasing the representation from zeroth to first order substantially improves PSNR from 47.06 dB to 51.83 dB under the same parameter budget. Extending the representation to second order achieves 51.21 dB, which is 0.62 dB lower than the first-order formulation. These results indicate that allocating part of the parameter budget to richer local functions can be considerably more effective than using the same budget solely to increase the number of zeroth-order primitives. However, increasing the polynomial order further is not monotonically beneficial, since additional per-primitive coefficients necessarily reduce the number of primitives under a fixed budget. In our setting, the first-order formulation provides the best trade-off between per-primitive function capacity and primitive count, and is therefore adopted in LegendreImage+.

5.3 SHARING FIRST-ORDER COEFFICIENTS

We compare shared and channel-independent first-order coefficients under the same 4M trainable-parameter budget. Both variants retain channel-specific zeroth-order RGB coefficients. Sharing the two first-order coefficients requires 7 trainable parameters per primitive, whereas assigning separate coefficients to each channel requires 11, leaving fewer primitives available under the same budget. As shown in Table 2, the shared variant achieves 51.83 dB, compared with 45.52 dB for the channel-independent variant. This result favors allocating more of the available budget to spatial coverage rather than channel-specific local slopes, motivating the shared parameterization adopted in LegendreImage+.

5.4 CHOICE OF COMPACT WINDOW

The spatial window determines how the local signal expansion is localized within the support of each primitive. To examine whether the performance of LegendreImage+ depends on a particular window function, we compare four compact windows under the same 4M parameter budget, while keeping the first-order Legendre representation and all other settings unchanged. All windows are defined over the normalized coordinate $t \in [-1, 1]$ and vanish outside the support. Specifically, we consider

$$w(t) = \begin{cases} \frac{1}{2}\left(1 + \cos(\pi t)\right), & \text{raised cosine,} \\ (1 - |t|)^2(1 + 2|t|), & \text{cubic smoothstep,} \\ 1 - |t|, & \text{triangular,} \\ 1 - t^2, & \text{parabolic,} \end{cases} \qquad |t| < 1, \tag{10}$$

and $w(t) = 0$ for $|t| \geq 1$. The corresponding two-dimensional window is constructed separably as $W(u, v) = w(u)w(v)$. The four choices therefore share the same compact support while differing in their attenuation profiles toward the support boundary.

As shown in Table 2, the raised-cosine window achieves 51.83 dB, closely followed by cubic smoothstep at 51.77 dB. Triangular and parabolic windows achieve 51.01 dB and 50.63 dB, respectively. The comparable performance of the raised-cosine and cubic-smoothstep windows shows that the compact local-function representation is robust to the precise attenuation profile, while the lower performance of the other variants indicates that window shape still matters.

6 DISCUSSION

6.1 BASIS FUNCTIONS, DOMAINS, AND MEASURES

The Legendre and Hermite expansions considered in this work correspond to different signal domains and measures: Legendre polynomials are orthogonal on the bounded interval $[-1, 1]$ under the Lebesgue measure, whereas Hermite polynomials are orthogonal on $\mathbb{R}$ under a Gaussian measure. A third example is the spherical-harmonic representation used for view-dependent color in 3D Gaussian Splatting Kerbl et al. (2023). For $f \in L^2(S^2)$,

$$\boxed{\int_0^{2\pi} \int_0^{\pi} |f(\theta, \phi)|^2 \sin\theta \, d\theta \, d\phi < \infty \quad \Longrightarrow \quad f(\theta, \phi) \sim \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_{\ell m} Y_\ell^m(\theta, \phi),} \tag{11}$$

where Y_ℓ^m form a complete orthogonal basis of $L^2(S^2)$ Atkinson & Han (2012); practical 3DGS uses a finite-order truncation of this expansion.

Thus, the Legendre, Hermite, and spherical-harmonic representations can all be viewed as generalized Fourier expansions associated with different signal domains and measures. This perspective suggests that basis selection in GFES should be considered together with the domain, measure, and structure of the represented signal.

6.2 COMPACT SUPPORT AND SPATIAL ADAPTIVITY

Compact support provides each primitive with a well-defined local region of influence, but also makes its spatial extent more difficult to optimize. For a compactly supported window, the weight is exactly zero outside the support, so pixels beyond the current support cannot provide gradient signals for adjusting its spatial extent. In contrast, Gaussian kernels have continuously decaying nonzero tails, while their scale, orientation, and anisotropy can be adjusted through covariance parameters. In our current implementation, we instead assign the support half-size during primitive allocation and keep it fixed during subsequent optimization.

This difference may induce distinct trade-offs at different representation densities. At low parameter budgets, fewer primitives must account for larger local regions, where adaptable spatial geometry may be particularly valuable. As the representation becomes denser, the local spatial scale h decreases and each primitive models a smaller neighborhood. For sufficiently smooth scalar signals, a degree-K Legendre projection on the normalized local domain has approximation error $O(h^{K+1})$, as derived in Appendix A. In particular, first-order local approximation yields an $O(h^2)$ error, suggesting that low-order expansions become increasingly effective at small spatial scales. This provides a possible explanation for why LegendreImage+ becomes more competitive at higher parameter budgets in Fig. 3.

This trade-off also suggests a natural extension of GFES. A learnable local coordinate transformation could adapt spatial extent, anisotropy, and orientation while retaining the local signal expansion, potentially combining the spatial adaptivity of Gaussian primitives with the local signal capacity of GFES.

7 CONCLUSION

We revisit primitive design in splatting-based image representation and introduce GFES, which views the signal carried by each primitive as a function represented through a truncated generalized Fourier expansion. This formulation makes the per-primitive signal space an explicit dimension of representation design. As a concrete instantiation, LegendreImage combines a Legendre polynomial expansion over a bounded local spatial domain with compact raised-cosine localization, enabling individual primitives to represent spatially varying local signals. Experiments show that the choice of per-primitive function space can substantially affect reconstruction quality under a fixed parameter budget, while its functional form, order, and spatial localization should be considered jointly. Further exploring expressive local representations together with effectively optimizable spatial extent remains an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the derivation of key steps in the mathematical proof presented in Appendix A and with the writing of that proof, to provide feedback on research methodology and experimental design, to assist with method implementation, and to support translation.

We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, formulate or modify research hypotheses, clean or reformat datasets, conduct qualitative or thematic analysis, or interpret experimental results. The remaining task requiring disclosure, namely synthetic data generation, is not applicable to this work.

We have reviewed all AI-assisted work. The mathematical derivations and proof in Appendix A were independently checked by the authors for correctness. AI-assisted content related to research methodology, experimental design, and method implementation was reviewed and verified by the authors against the actual methodology and implementation. AI-assisted translations were checked and revised by the authors against the intended meaning.

We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the definition of the Legendre primitive and the rendering formulation in Sec. 3. The datasets, evaluation protocol, parameter budgets, optimization settings, and hardware configuration are described in Sec. 4.1. The adaptive primitive allocation procedure and its hyperparameters are detailed in Appendix B.

REFERENCES

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

Kendall Atkinson and Weimin Han. *Spherical harmonics and approximations on the unit sphere: an introduction. Lecture Notes in Mathematics*, volume 2044. Springer Heidelberg, 2012.

Jiacong Chen, Qingyu Mao, Shuai Liu, Chao Li, Jierun Lin, Xiandong Meng, Fanyang Meng, and Yongsheng Liang. Entropy-aware image representation via 2d gaussian splatting. *Signal Processing*, pp. 110521, 2026.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8628–8638, 2021.

Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4182–4194. IEEE, 2023.

Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021.

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022.

Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992). `https://r0k.us/graphics/kodak/`, 1993.

Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3512–3516. IEEE, 2019.

Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. Ges: Generalized exponential splatting for efficient radiance field rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19812–19822. IEEE, 2024.

Yi-Hua Huang, Ming-Xian Lin, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Deformable radial kernel splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21513–21523. IEEE, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Erwin Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, Hoboken, NJ, 10 edition, 2011. ISBN 9780470458365.

Peter Lancaster and Kes Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158, 1981.

David Levin. The approximation power of moving least-squares. *Mathematics of computation*, 67 (224):1517–1531, 1998.

Tiantian Li, Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Jun Zhang, and Yan Wang. GaussianImage++: Boosted image representation and compression with 2d gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6442–6449, 2026.

Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788*, 2021.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):102:1–102:15, 2022.

Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *European Conference on Computer Vision*, pp. 318–333. Springer, 2022.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 18507–18516. IEEE, 2023.

Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524, 1968.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Gábor Szegő. *Orthogonal Polynomials*, volume 23 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, 4 edition, 1975.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Skylar Wurster, Ran Zhang, and Changxi Zheng. Gabor splatting for high-quality gigapixel image representations. In *ACM SIGGRAPH 2024 Posters*, pp. 1–2. 2024.

Ruihan Yu, Tianyu Huang, Jingwang Ling, and Feng Xu. 2dgh: 2d gaussian-hermite splatting for high-quality rendering and better geometry features. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

Zhaojie Zeng, Yuesong Wang, Tao Guan, Chao Yang, and Lili Ju. Instant gaussianimage: A generalizable and self-adaptive image representation via 2d gaussian splatting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27896–27905. IEEE, 2025.

Wenhao Zhang, Hao Zhu, Delong Wu, Di Kang, Linchao Bao, Xun Cao, and Zhan Ma. Wipes: wavelet-based visual primitives. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27338–27347. IEEE, 2025a.

Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, pp. 327–345. Springer, 2024.

Yunxiang Zhang, Bingxuan Li, Alexandr Kuznetsov, Akshay Jindal, Stavros Diolatzis, Kenneth Chen, Anton Sochenov, Anton Kaplanyan, and Qi Sun. Image-gs: Content-adaptive image representation via 2d gaussians. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025b.

Lingting Zhu, Guying Lin, Jinnan Chen, Xinjie Zhang, Zhenchao Jin, Zhao Wang, and Lequan Yu. Large images are Gaussians: High-quality large image representation with levels of 2d gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10977–10985, 2025.

APPENDIX

A LOCAL APPROXIMATION ERROR OF TRUNCATED LEGENDRE EXPANSIONS

We analyze how the approximation error of a fixed-order Legendre expansion depends on the spatial extent of a local primitive. Consider a sufficiently smooth one-dimensional signal f around x_0, and define the normalized local signal

$$g_h(u) = f(x_0 + hu), \qquad u \in [-1, 1], \tag{12}$$

where $h > 0$ denotes the spatial scale of the local region.

The Legendre expansion of g_h is

$$g_h(u) = \sum_{n=0}^{\infty} a_n(h) P_n(u), \tag{13}$$

with

$$a_n(h) = \frac{2n+1}{2} \int_{-1}^{1} g_h(u) P_n(u)\, du. \tag{14}$$

Scaling of the Legendre coefficients. Using Rodrigues' formula,

$$P_n(u) = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n, \tag{15}$$

we have

$$a_n(h) = \frac{2n+1}{2^{n+1} n!} \int_{-1}^{1} f(x_0 + hu) \frac{d^n}{du^n} (u^2 - 1)^n\, du. \tag{16}$$

Integrating by parts n times eliminates all boundary terms, since $(u^2 - 1)^n$ has zeros of multiplicity n at $u = \pm 1$. This gives

$$\begin{aligned} a_n(h) &= \frac{2n+1}{2^{n+1} n!} \int_{-1}^{1} \frac{d^n}{du^n} f(x_0 + hu)(1 - u^2)^n\, du \\ &= \frac{2n+1}{2^{n+1} n!} h^n \int_{-1}^{1} f^{(n)}(x_0 + hu)(1 - u^2)^n\, du. \end{aligned} \tag{17}$$

Therefore,

$$|a_n(h)| \le C_n h^n \sup_{|x-x_0|\le h} |f^{(n)}(x)|, \tag{18}$$

where C_n is independent of h. Hence,

$$\boxed{a_n(h) = O(h^n)}. \tag{19}$$

Thus, as the spatial extent decreases, higher-order Legendre modes decay progressively faster.

Fixed-order truncation error. Let

$$\Pi_K^L g_h = \sum_{n=0}^{K} a_n(h) P_n \tag{20}$$

be the $L^2([-1,1])$ orthogonal projection onto the polynomial space $\mathcal{P}_K$ of degree at most K.

For any $v \in H^{K+1}([-1,1])$, polynomial approximation on the fixed interval $[-1,1]$ gives

$$\|v - \Pi_K^L v\|_{L^2([-1,1])} \le C_K \|v^{(K+1)}\|_{L^2([-1,1])}, \tag{21}$$

where C_K depends only on K and not on v.

Applying this inequality to g_h and using

$$g_h^{(K+1)}(u) = h^{K+1} f^{(K+1)}(x_0 + hu), \tag{22}$$

we obtain

$$\begin{aligned} \|g_h - \Pi_K^L g_h\|_{L^2([-1,1])} &\le C_K h^{K+1} \left\| f^{(K+1)}(x_0 + h\,\cdot) \right\|_{L^2([-1,1])} \\ &\le \sqrt{2}\, C_K h^{K+1} \sup_{|x-x_0|\le h} |f^{(K+1)}(x)|. \end{aligned} \tag{23}$$

Consequently, for fixed K,

$$\boxed{\|g_h - \Pi_K^L g_h\|_{L^2([-1,1])} = O(h^{K+1})}. \tag{24}$$

In particular, the first-order scalar Legendre projection satisfies

$$\|g_h - \Pi_1^L g_h\|_{L^2([-1,1])} = O(h^2). \tag{25}$$

For sufficiently smooth scalar signals, the first-order Legendre projection therefore achieves an $O(h^2)$ local error, compared with $O(h)$ for the zeroth-order projection. More generally, the factor h^{K+1} shows how spatial refinement and polynomial order jointly control local approximation accuracy: reducing the support scale improves accuracy even when the expansion order remains fixed.

B Adaptive Primitive Allocation

We combine progressive densification with fixed-budget relocation to allocate primitives according to the reconstruction residual. Densification adds primitives during early optimization, while relocation redistributes existing primitives after the target count has been reached.

B.1 Allocation Algorithm

Initialization and parameter budget. For a target image $\mathbf{I}$ of size $H \times W$ and a trainable-parameter budget B, the target primitive count is $N^\star = \lfloor B/p \rfloor$. Our first-order RGB-shared primitive uses $p = 7$: three RGB constant coefficients, two shared first-order coefficients, and two center offsets. We initialize a fraction f_0 of the budget on a regular grid. The common stride s is chosen so that $N_0 = \lceil W/s \rceil \lceil H/s \rceil$ approximates $\lfloor f_0 B/p \rfloor$ while satisfying $pN_0 \le B$. Grid anchors are $(0.5 + is, 0.5 + js)$ in pixel coordinates, with initial square-support half-size $R_0 = s$. RGB constant coefficients are initialized to the unweighted target-image mean within each support, and shared first-order coefficients are initialized to zero. Primitive centers follow $\boldsymbol{\mu}_i = \boldsymbol{\mu}_i^0 + \beta s \tanh(\tau \boldsymbol{\theta}_i)$, where $\boldsymbol{\mu}_i^0$ is the anchor and $\boldsymbol{\theta}_i$ is initialized to zero.

Progressive growth schedule. Let $t_1, \ldots, t_L$ denote the growth iterations. Starting with $r_0 = N^\star - N_0$, we allocate half of the remaining count at each event and the full remainder at the last event:

$$a_k = \begin{cases} \lfloor r_{k-1}/2 \rfloor, & k < L, \\ r_{k-1}, & k = L, \end{cases} \qquad r_k = r_{k-1} - a_k, \qquad N_k = N_{k-1} + a_k. \tag{26}$$

Events with zero additions are skipped. At each positive growth event, the new primitives receive half-size

$$R_k = \max\left\{ R_{\min}, \rho\delta^{\ell_k - 1}\sqrt{\frac{HW}{N_k}} \right\}, \tag{27}$$

where ℓ_k counts positive growth events, ρ sets the spatial scale, δ controls decay, and $R_{\min}$ is the minimum half-size. Growth leaves existing half-sizes unchanged, and all half-sizes are fixed during gradient-based optimization.

Residual candidates. At iteration t_k, after the Adam and scheduler updates, we render the updated model and compute $S(\mathbf{x}) = \frac{1}{3}\sum_c |I^{(c)}(\mathbf{x}) - \hat{I}^{(c)}(\mathbf{x})|$ from the unclipped output. We prioritize local maxima in a square neighborhood of radius r_{loc}, ordered by decreasing residual, to construct up to ηa_k candidate anchors at pixel centers. Candidates within distance d_h of anchors inserted at earlier growth events are excluded. If local maxima are insufficient, remaining eligible pixels are selected in residual order. Ties follow raster order, and duplicate locations are excluded within each selection. History exclusion is relaxed only when necessary to fill the requested count.

Candidate fitting and gain. For each candidate j, its anchor and half-size are fixed while its coefficients $\boldsymbol{\gamma}_j = (c_{j,00}^{(R)}, c_{j,00}^{(G)}, c_{j,00}^{(B)}, c_{j,10}, c_{j,01})^\top$ are fitted. Its local function is $\mathbf{Q}_j = \mathbf{c}_{j,00} + \mathbf{1}(c_{j,10}u_j + c_{j,01}v_j)$. Let w_j be its window and $Z = \sum_i W_i$ the current weight sum. The reconstruction after virtual insertion is

$$\hat{\mathbf{I}}_j(\mathbf{x}; \boldsymbol{\gamma}_j) = \frac{Z(\mathbf{x})\hat{\mathbf{I}}(\mathbf{x}) + w_j(\mathbf{x})\mathbf{Q}_j(\mathbf{x}; \boldsymbol{\gamma}_j)}{Z(\mathbf{x}) + w_j(\mathbf{x})}. \tag{28}$$

We split in-image support pixels with window weight above float32 machine epsilon into two checkerboard subsets, Ω_j^0 and Ω_j^1. A pixel with integer indices (x, y) belongs to Ω_j^b when $(x + y) \bmod 2 = b$, for $b \in \{0, 1\}$. Coefficients are obtained by ridge least squares on the first subset:

$$\boldsymbol{\gamma}_j^{\text{fit}} = \arg\min_{\boldsymbol{\gamma}} \sum_{\mathbf{x} \in \Omega_j^0} \|\mathbf{I}(\mathbf{x}) - \hat{\mathbf{I}}_j(\mathbf{x}; \boldsymbol{\gamma})\|_2^2 + \lambda\|\boldsymbol{\gamma}\|_2^2. \tag{29}$$

The gain is evaluated on the second subset:

$$G_j = \sum_{\mathbf{x} \in \Omega_j^1} \left[\|\mathbf{I}(\mathbf{x}) - \hat{\mathbf{I}}(\mathbf{x})\|_2^2 - \|\mathbf{I}(\mathbf{x}) - \hat{\mathbf{I}}_j(\mathbf{x}; \boldsymbol{\gamma}_j^{\text{fit}})\|_2^2 \right]. \tag{30}$$

We additionally fit coefficients on the full valid support for insertion. For the five-coefficient model, a candidate is valid if both solves succeed with finite coefficients and each checkerboard subset contains at least five valid pixels.

Selection and insertion. We select up to a_k valid candidates in decreasing gain order and initialize them with their full-support fitted coefficients. Any remaining allocation is filled by residual selection, excluding already selected locations. Full-support fitting is attempted at these additional anchors while retaining the checkerboard sample-count and solver-validity checks described above. Invalid fits use target-pixel RGB constant coefficients and zero first-order coefficients. When no gain candidate is valid, all insertion locations are selected by the residual-based candidate selector. Candidates are evaluated against the same pre-insertion reconstruction and appended as one batch, with zero raw center offsets. Newly inserted anchors are added to the selection history. Existing Adam moments are preserved and new-row moments are initialized to zero, while the global optimizer step and learning-rate schedule are retained. Geometry and rasterization caches are refreshed before optimization resumes.

Table 3: Parameters for adaptive primitive allocation.

Parameter	Value
Initial budget fraction f_0	0.5
Growth iterations t_k	1,000, 2,000, 3,000, 4,000
Support scale ρ	$\frac{1}{\sqrt{\pi}} \approx 0.56418958$
Support decay δ	0.9
Minimum half-size $R_{\min}$	2 pixels
Local-maximum radius r_{loc}	2 pixels
Candidate multiplier η	4
Anchor exclusion distance d_h	$0.19s$
Residual smoothing	None
Ridge coefficient λ	10^{-4}
Center offset parameters (β, τ)	$(0.4, 0.5)$
Relocation iteration	4,100
Adaptive-row relocation fraction f_r	0.1
Relocation half-size multiplier κ	0.8
Donor weight-sum threshold ϵ_Z	10^{-3}

Fixed-budget relocation. After growth, we relocate a fraction f_r of the adaptive primitives, preserving the initial grid and total count. Donors are ranked by their leave-one-out effect on the current reconstruction,

$$D_i = \sum_{\mathbf{x} \in \text{supp}(i)} \left(\frac{W_i(\mathbf{x})}{Z(\mathbf{x}) - W_i(\mathbf{x})} \right)^2 \frac{\|\mathbf{Q}_i(\mathbf{x}) - \hat{\mathbf{I}}(\mathbf{x})\|_2^2}{3}. \tag{31}$$

We select the smallest scores among adaptive primitives satisfying $Z - W_i \geq \epsilon_Z$ throughout their active support. Recipient anchors are selected from the current L1 residual, excluding locations within d_h of surviving centers. Each relocated primitive receives half-size κR_{last}, where R_{last} is the half-size assigned at the final positive growth event. RGB constant coefficients are initialized from the target pixel; first-order coefficients, raw center offsets, and row-wise Adam moments are reset to zero. The global optimizer step and scheduler are retained. A coverage and normalization check accepts the replacement, retries with anchors targeting coverage gaps, or restores the original primitives.

Settings. Table 3 summarizes the allocation parameters; s denotes the initial grid stride.

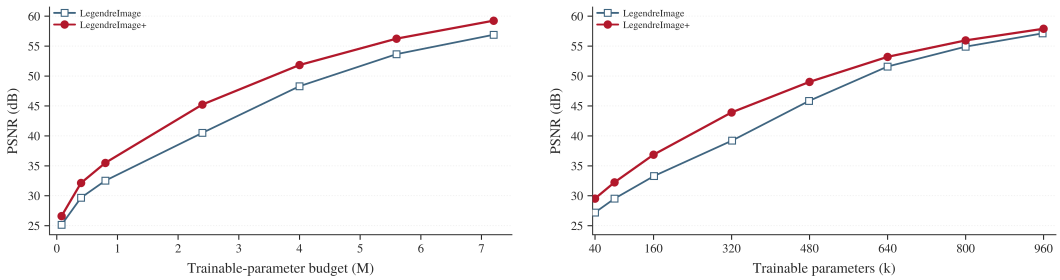
B.2 Ablation of Adaptive Allocation

We compare the full allocation pipeline with a fixed-allocation variant using the same first-order RGB-shared primitives and target trainable-parameter budgets. The fixed-allocation variant constructs a regular grid using the full budget at initialization. Its support sizes and coefficients follow the grid initialization described above. Primitive centers and coefficients are then optimized with Adam for 30,000 iterations, with both densification and relocation disabled.

As shown in Fig. 4, the full pipeline achieves higher PSNR across the evaluated budgets, demonstrating the benefit of residual-guided adaptive allocation. The fixed-allocation variant also improves consistently with increasing budget, showing that additional representation capacity remains useful even without adaptive allocation.

C Higher-Order LegendreImage

To examine how the polynomial order affects representation capacity, we evaluate LegendreImage with a fixed set of 100k primitives and vary the maximum order from $K = 0$ to $K = 6$. For this experiment, all Legendre expansion coefficients are parameterized independently for the three RGB channels. As shown in Table 4, reconstruction quality improves consistently with increasing order, while the additional coefficients increase training and rendering cost.



(a) DIV2K (b) Kodak

Figure 4: **Ablation of adaptive primitive allocation on DIV2K.** PSNR across parameter budgets for the full pipeline with densification and relocation, and the fixed-allocation variant with all primitives allocated at initialization.

Table 4: Higher-order LegendreImage results on DIV2K with a fixed set of 100k primitives.

Order	Params / Primitive	PSNR ↑	MS-SSIM ↑	Train (s) ↓	Eval (ms) ↓	FPS ↑
$K = 0$	5	29.24	0.965	278.41	2.20	456.31
$K = 1$	11	32.53	0.986	367.56	2.43	413.01
$K = 2$	20	35.46	0.993	440.49	2.65	379.23
$K = 3$	32	38.29	0.996	525.57	3.09	325.45
$K = 4$	47	41.07	0.997	621.04	3.42	293.45
$K = 5$	65	43.45	0.998	738.90	3.91	256.54
$K = 6$	86	45.67	0.998	892.17	4.52	222.15

D Additional Examples of Legendre Primitives

Figure 5 presents additional examples of Legendre primitives, showing the input image regions, the corresponding continuous primitive fields, and their discretized pixel representations.

E Complete Quantitative Results

Tables 6 and 7 report the Kodak and DIV2K results, respectively, grouped by trainable-parameter budget, including the DIV2K fixed-allocation variant. Budget denotes the target number of trainable scalars. Evaluation time is expressed in milliseconds. FPS and evaluation time are the separately reported averages.

Table 5 additionally reports results on DIV8K at a target trainable-parameter budget of 60M, with an average image resolution of 5591×7032. LegendreImage+ achieves 54.61 dB, exceeding GaussianImage++ and LIG by 2.19 dB and 8.88 dB, respectively. These results extend the observed PSNR advantage to substantially higher-resolution images.

Table 5: DIV8K results at a target trainable-parameter budget of 60M. The average image resolution is 5591×7032.

Method	PSNR (dB) ↑	MS-SSIM ↑	Train (s) ↓	Eval (ms) ↓	FPS ↑
LIG	45.73	0.999	3050.12	22.20	45.16
GaussianImage++	52.42	1.000	2357.89	21.20	47.63
LegendreImage+ (Ours)	54.61	0.998	6717.95	42.16	23.78

Table 6: Complete Kodak results grouped by trainable-parameter budget.

Parameter Budget	Method	PSNR (dB) ↑	MS-SSIM ↑	Train (s) ↓	Eval (ms) ↓	FPS ↑
40k	GaussianImage (Cholesky)	29.87	0.963	187.90	1.38	736.51
	GaussianImage (RS)	29.95	0.963	110.05	0.65	1535.54
	WIPES (Cholesky)	29.34	0.955	111.39	0.68	1475.16
	WIPES (RS)	29.46	0.956	121.73	0.71	1411.17
	LIG	28.50	0.948	158.07	1.15	868.92
	GaussianImage++	31.77	0.960	137.90	0.65	1540.94
	LegendreImage (Ours)	27.20	0.945	173.20	1.02	985.94
	LegendreImage+ (Ours)	29.53	0.951	196.39	1.02	992.18
80k	GaussianImage (Cholesky)	32.48	0.982	101.77	0.60	1684.19
	GaussianImage (RS)	32.58	0.982	116.08	0.67	1487.30
	WIPES (Cholesky)	31.94	0.978	112.83	0.68	1484.66
	WIPES (RS)	32.03	0.979	128.64	0.77	1312.33
	LIG	31.13	0.976	161.22	1.16	863.42
	GaussianImage++	35.27	0.983	137.97	0.67	1517.45
	LegendreImage (Ours)	29.52	0.973	161.16	1.02	982.34
	LegendreImage+ (Ours)	32.24	0.976	183.68	1.05	955.81
160k	GaussianImage (Cholesky)	36.01	0.993	102.24	0.60	1679.35
	GaussianImage (RS)	36.15	0.993	116.47	0.72	1412.56
	WIPES (Cholesky)	35.38	0.991	105.94	0.62	1632.06
	WIPES (RS)	35.48	0.991	121.72	0.71	1421.40
	LIG	34.72	0.991	156.48	1.13	894.85
	GaussianImage++	39.06	0.993	133.26	0.63	1589.12
	LegendreImage (Ours)	33.29	0.989	152.51	1.07	938.32
	LegendreImage+ (Ours)	36.85	0.989	181.84	1.19	845.92
320k	GaussianImage (Cholesky)	40.64	0.998	104.41	0.60	1673.93
	GaussianImage (RS)	41.00	0.998	113.90	0.67	1484.49
	WIPES (Cholesky)	40.03	0.997	109.34	0.62	1615.27
	WIPES (RS)	40.12	0.997	122.35	0.71	1421.94
	LIG	39.90	0.998	160.86	1.17	857.80
	GaussianImage++	43.53	0.998	133.01	0.67	1515.06
	LegendreImage (Ours)	39.21	0.997	155.25	1.19	837.92
	LegendreImage+ (Ours)	43.91	0.997	182.37	1.28	783.57
480k	GaussianImage (Cholesky)	43.53	0.999	104.33	0.64	1607.43
	GaussianImage (RS)	43.64	0.999	113.12	0.71	1433.94
	WIPES (Cholesky)	43.29	0.998	114.73	0.65	1557.22
	WIPES (RS)	43.32	0.998	121.46	0.70	1431.81
	LIG	44.27	0.999	162.44	1.24	807.83
	GaussianImage++	46.64	0.999	133.92	0.64	1572.17
	LegendreImage (Ours)	45.84	0.998	154.32	1.28	780.20
	LegendreImage+ (Ours)	49.02	0.998	190.01	1.37	728.32
640k	GaussianImage (Cholesky)	43.31	0.998	101.76	0.56	1774.83
	GaussianImage (RS)	41.07	0.997	118.94	0.74	1370.25
	WIPES (Cholesky)	45.38	0.999	113.27	0.62	1616.91
	WIPES (RS)	45.44	0.999	120.04	0.71	1424.94
	LIG	48.24	1.000	171.90	1.27	798.80
	GaussianImage++	49.05	0.999	129.17	0.62	1618.38
	LegendreImage (Ours)	51.58	0.998	155.60	1.33	753.37
	LegendreImage+ (Ours)	53.20	0.998	191.47	1.60	625.36
800k	GaussianImage (Cholesky)	38.92	0.995	105.57	0.58	1746.92
	GaussianImage (RS)	35.36	0.988	116.93	0.72	1404.27
	WIPES (Cholesky)	45.35	0.999	118.19	0.67	1518.86
	WIPES (RS)	45.59	0.999	125.40	0.71	1401.26
	LIG	51.38	1.000	166.13	1.12	892.18
	GaussianImage++	51.03	1.000	133.31	0.64	1558.15
	LegendreImage (Ours)	54.90	0.998	156.64	1.48	676.69
	LegendreImage+ (Ours)	55.94	0.998	200.45	1.81	554.68
960k	GaussianImage (Cholesky)	33.71	0.982	108.75	0.60	1673.21
	GaussianImage (RS)	30.47	0.963	113.58	0.66	1523.53
	WIPES (Cholesky)	40.90	0.997	119.03	0.66	1518.71
	WIPES (RS)	42.58	0.998	124.05	0.69	1452.76
	LIG	53.70	1.000	176.07	1.27	792.94
	GaussianImage++	52.43	1.000	132.82	0.62	1603.73
	LegendreImage (Ours)	57.12	0.999	157.87	1.37	731.48
	LegendreImage+ (Ours)	57.89	0.999	201.86	1.71	585.17

Table 7: Complete DIV2K results grouped by trainable-parameter budget.

Parameter Budget	Method	PSNR (dB) ↑	MS-SSIM ↑	Train (s) ↓	Eval (ms) ↓	FPS ↑
0.08M	GaussianImage (Cholesky)	26.54	0.910	125.15	0.75	1347.00
	GaussianImage (RS)	26.61	0.912	130.67	0.77	1311.36
	WIPES (Cholesky)	25.91	0.892	136.52	0.82	1230.50
	WIPES (RS)	26.09	0.897	142.22	0.82	1223.26
	LIG	24.79	0.856	338.13	1.25	804.10
	GaussianImage++	28.15	0.914	158.60	0.75	1333.58
	LegendreImage (Ours)	25.15	0.887	664.94	1.73	596.73
	LegendreImage+ (Ours)	26.61	0.893	774.09	2.36	426.40
0.4M	GaussianImage (Cholesky)	31.45	0.977	143.79	0.85	1177.48
	GaussianImage (RS)	31.55	0.977	147.39	0.86	1171.09
	WIPES (Cholesky)	30.71	0.971	160.38	0.97	1036.39
	WIPES (RS)	30.86	0.971	165.66	0.98	1024.61
	LIG	29.87	0.966	355.20	1.35	743.14
	GaussianImage++	34.17	0.979	178.87	0.96	1052.85
	LegendreImage (Ours)	29.67	0.969	502.72	2.48	405.45
	LegendreImage+ (Ours)	32.13	0.973	625.95	2.73	368.75
0.8M	GaussianImage (Cholesky)	34.21	0.989	155.20	0.95	1052.55
	GaussianImage (RS)	34.33	0.989	161.27	0.96	1040.66
	WIPES (Cholesky)	33.35	0.985	185.46	1.12	901.05
	WIPES (RS)	33.47	0.985	188.42	1.11	903.86
	LIG	32.93	0.985	287.87	1.46	687.91
	GaussianImage++	37.40	0.990	184.75	1.08	934.30
	LegendreImage (Ours)	32.51	0.985	445.64	2.79	360.64
	LegendreImage+ (Ours)	35.49	0.987	564.88	3.06	328.62
2.4M	GaussianImage (Cholesky)	39.77	0.997	201.99	1.33	754.85
	GaussianImage (RS)	40.04	0.997	202.40	1.31	765.42
	WIPES (Cholesky)	38.77	0.996	235.67	1.52	660.75
	WIPES (RS)	38.70	0.995	234.45	1.52	660.29
	LIG	39.83	0.997	380.86	1.76	570.10
	GaussianImage++	44.07	0.998	209.40	1.50	672.08
	LegendreImage (Ours)	40.50	0.996	396.03	3.26	307.05
	LegendreImage+ (Ours)	45.23	0.997	533.27	3.51	285.32
4.0M	GaussianImage (Cholesky)	40.66	0.996	259.09	1.79	559.52
	GaussianImage (RS)	38.81	0.995	247.69	1.68	596.90
	WIPES (Cholesky)	41.42	0.997	296.18	1.99	503.81
	WIPES (RS)	41.44	0.997	289.22	1.97	506.85
	LIG	44.86	0.999	437.01	2.01	497.72
	GaussianImage++	48.06	0.999	231.18	1.84	545.72
	LegendreImage (Ours)	48.29	0.998	406.06	3.53	283.73
	LegendreImage+ (Ours)	51.83	0.998	566.88	4.21	238.02
5.6M	GaussianImage (Cholesky)	35.52	0.989	322.70	2.20	454.26
	GaussianImage (RS)	31.24	0.972	303.15	2.03	493.89
	WIPES (Cholesky)	40.88	0.996	359.70	2.41	415.62
	WIPES (RS)	42.90	0.997	337.83	2.36	423.56
	LIG	48.97	1.000	453.31	2.32	431.74
	GaussianImage++	50.79	1.000	252.40	2.05	489.55
	LegendreImage (Ours)	53.65	0.998	422.19	3.92	255.75
	LegendreImage+ (Ours)	56.24	0.998	625.38	4.80	208.56
7.2M	GaussianImage (Cholesky)	29.74	0.959	391.20	2.66	376.61
	GaussianImage (RS)	25.42	0.903	361.88	2.45	408.06
	WIPES (Cholesky)	35.98	0.990	418.79	2.82	354.52
	WIPES (RS)	42.85	0.997	392.63	2.79	358.38
	LIG	52.02	1.000	490.53	2.58	387.35
	GaussianImage++	52.40	1.000	272.35	2.26	445.33
	LegendreImage (Ours)	56.89	0.998	442.74	4.25	235.74
	LegendreImage+ (Ours)	59.24	0.998	682.20	5.30	188.86

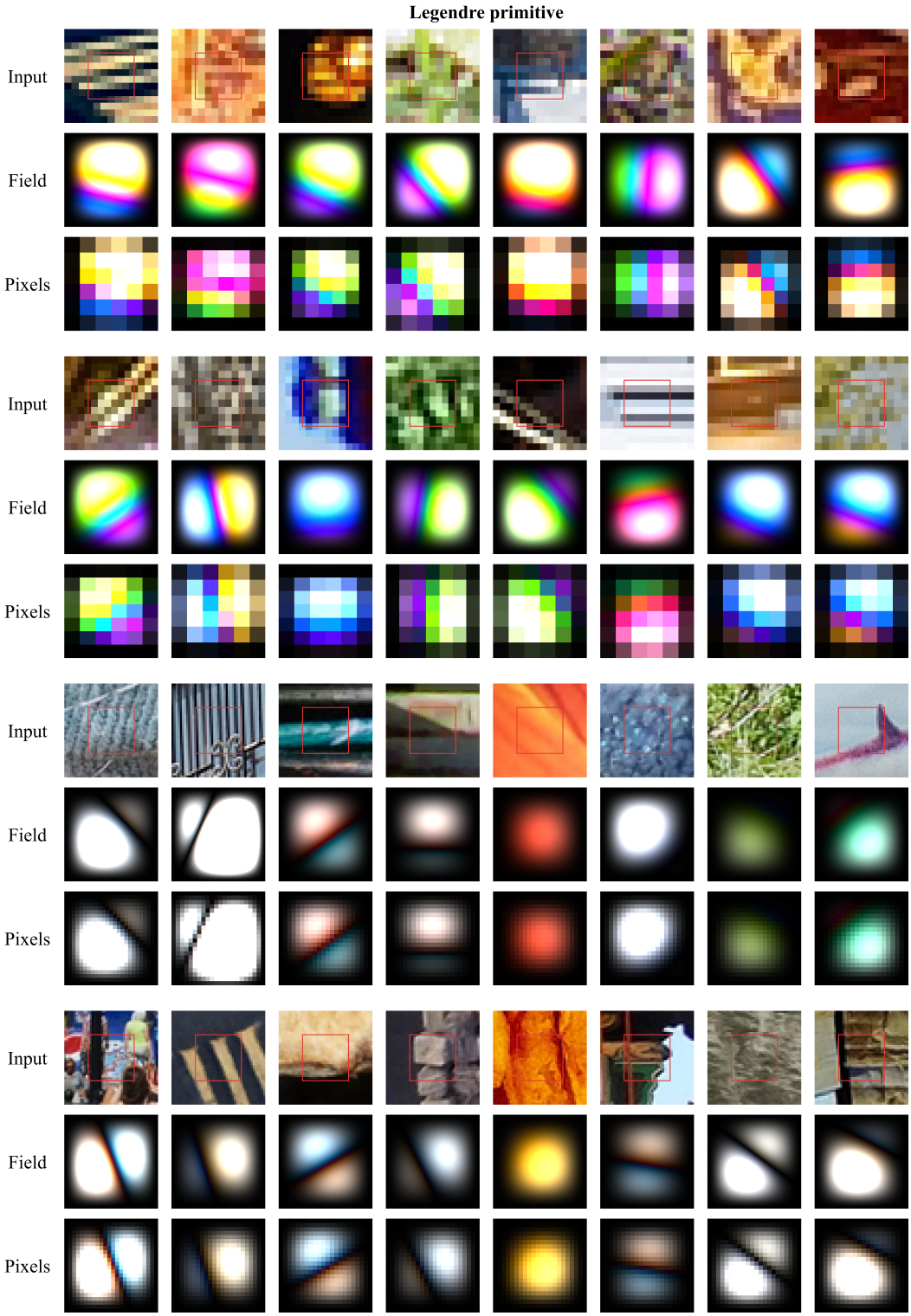


Figure 5: Additional examples of Legendre primitives.