mathematics

MDPI

Article

Toward Trustworthy On-Device AI: A Quantization-Robust Parameterized Hybrid Neural Filtering Framework

Sangwoo Hong [1], Seung-Wook Kim [2], Seunghyun Moon [3] and Seowon Ji [1,*]

[1] Department of Computer Science and Engineering, Konkuk University, Seoul 05029, Republic of Korea; swhong06@konkuk.ac.kr
[2] Division of Electrical and Communication Engineering, Pukyong National University, Busan 48513, Republic of Korea; swkim@pknu.ac.kr
[3] Department of Electrical and Electronics Engineering, Konkuk University, Seoul 05029, Republic of Korea; smoon@konkuk.ac.kr
* Correspondence: seowonji@konkuk.ac.kr

Abstract

Recent advances in deep learning have led to a proliferation of AI services for the general public. Consequently, constructing trustworthy AI systems that operate on personal devices has become a crucial challenge. While on-device processing is critical for privacy-preserving and latency-sensitive applications, conventional deep learning approaches often suffer from instability under quantization and high computational costs. Toward a trustworthy and efficient on-device solution for image processing, we present a hybrid neural filtering framework that combines the representational power of lightweight neural networks with the stability of classical filters. In our framework, the neural network predicts a low-dimensional parameter map that guides the filter's behavior, effectively decoupling parameter estimation from the final image synthesis. This design enables a truly trustworthy AI system by operating entirely on-device, which eliminates the reliance on servers and significantly reduces computational cost. To ensure quantization robustness, we introduce a basis-decomposed parameterization, a design mathematically proven to bound reconstruction errors. Our network predicts a set of basis maps that are combined via fixed coefficients to form the final guidance. This architecture is intrinsically robust to quantization and supports runtime-adaptive precision without retraining. Experiments on depth map super-resolution validate our approach. Our framework demonstrates exceptional quantization robustness, exhibiting no performance degradation under 8-bit quantization, whereas a baseline suffers a significant 1.56 dB drop. Furthermore, our model's significantly lower Mean Squared Error highlights its superior stability, providing a practical and mathematically grounded pathway toward trustworthy on-device AI.

Keywords: trustworthy AI; on-device AI; hybrid neural filtering; basis decomposition

MSC: 68T07; 65D15

check for updates

Academic Editor: Huawen Liu

Received: 27 August 2025
Revised: 14 October 2025
Accepted: 22 October 2025
Published: 29 October 2025

Citation: Hong, S.; Kim, S.-W.; Moon, S.; Ji, S. Toward Trustworthy On-Device AI: A Quantization-Robust Parameterized Hybrid Neural Filtering Framework. *Mathematics* **2025**, *13*, 3447. https://doi.org/10.3390/math13213447

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recent advances in deep learning have led to significant progress in image processing and computer vision tasks such as image restoration, enhancement, recognition, and segmentation, enabling applications in mobile photography, medical imaging, and autonomous driving [1–3]. With the growing demand for private and real-time inference, there is an increasing need to deploy AI systems directly on edge devices, including

smartphones and wearable devices. This trend has shifted the research focus from solely maximizing accuracy to ensuring the development of trustworthy AI systems [4–7]. Although on-device processing inherently enhances privacy, a truly trustworthy system must also guarantee robustness and reliability. The full scope of Trustworthy AI includes critical aspects like fairness and uncertainty. However, for on-device applications, operational stability under hardware constraints is paramount. This work therefore focuses on robustness and reliability as the foundational pillars of trustworthiness in this context. These properties are particularly challenging to maintain under the limited computational power of on-device environments. Designing AI systems that satisfy these constraints is therefore essential, not only for practical deployment but also as a well-posed engineering problem under quantifiable resource limitations.

Conventional strategies [8–10] for image enhancement on the edge device can be broadly categorized into two approaches. The first approach adopts end-to-end neural network models, which can produce high-quality outputs but often incur computational and memory costs that exceed the limits of resource-constrained devices. Techniques such as quantization and pruning can help alleviate this issue. However, they often lead to noticeable performance degradation. This reliance on server-side processing to compensate for the quality loss can undermine the core principles of on-device trustworthiness. The second approach includes classical, hand-crafted filters such as the Gaussian Filter, Guided Filter [11], Bilateral Filter [12], and more advanced quasi-linear variants like Anisotropic Diffusion, Domain Transform, and Rolling Guidance Filtering [13,14]. These methods are computationally efficient and mathematically interpretable, but typically require per-image parameter tuning, which limits their reliability in practical applications. Hybrid approaches [15,16] that combine neural and classical filtering aim to alleviate this limitation. However, they often depend on deep architectures or suffer performance degradation under low-bit quantization. Therefore, achieving a well-balanced AI solution that satisfies performance, efficiency, and trustworthiness under on-device constraints remains an open challenge.

To address this challenge, we propose a parameterized hybrid neural filtering framework. The core idea is to decouple parameter estimation from signal reconstruction: a lightweight neural network predicts a compact parameter map, which then guides a classical filter to synthesize the final image. This design significantly reduces computational costs by limiting the neural network's task to representation learning, rather than full image-to-image transformation. Crucially, this hybrid structure enhances robustness. By delegating the precision-sensitive synthesis stage to the mathematically stable classical filter, the framework is inherently more resilient to quantization errors.

We further amplify this robustness by introducing a basis-decomposed parameterization. The network predicts multiple low-precision basis maps that are combined via fixed coefficients, a design mathematically proven to bound reconstruction errors. This approach, interpretable as a form of low-rank factorization, provides an intrinsically quantization-friendly structure and enables runtime-adaptive precision, making our framework not only efficient but also inherently trustworthy.

In summary, this work makes four key contributions. First, we propose a hybrid framework that integrates the representational power of neural networks with the stability and interpretability of classical image filters, enabling efficient and reliable on-device image enhancement. Second, we introduce a basis-decomposed parameterization, interpretable as a form of multi-resolution basis expansion or low-rank factorization. This design inherently improves robustness under aggressive quantization and supports runtime-adjustable precision, enabling flexible trade-offs between quality and efficiency. Third, we experimentally demonstrate that our mathematically grounded approach achieves

competitive performance while maintaining a lightweight profile suitable for on-device deployment. Lastly, by fully eliminating reliance on server-side computation, our method offers a practical foundation for building trustworthy AI systems that are robust, privacy-preserving, and deployable in real-world cases.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed hybrid framework and basis-decomposed parameterization. Section 4 describes the experimental setup and results. Section 5 discusses about limitations and outlines possible directions for future research.

2. Related Work

In this section, we review prior work across four relevant areas: (1) classical linear image filtering, (2) lightweight neural networks, (3) quantization for on-device inference, and (4) basis expansion (or low-rank parameterization) These domains respectively contribute to mathematical interpretability and differentiability, computational efficiency on edge devices, compact model design for on-device deployment, and structured representations for parameter efficiency and robustness. Each of these components is closely related to our proposed framework. In the following subsections, we summarize representative methods and highlight how our work builds upon and differs from existing approaches in each category.

2.1. Classical Linear Image Filtering

Classical edge-preserving filters, such as the Guided Filter [11] and Bilateral Filter [12], are valued for their mathematical interpretability and predictable behavior. A key property is their differentiability; many can be formulated such that their weights are differentiable with respect to external parameters, allowing them to be integrated into neural networks for end-to-end training. A representative example is the Guided Filter [11], which assumes a local linear model between a guidance image G and the output $\hat{I}$ in each local window ω_k:

$$\hat{I}(x) = a_k G(x) + b_k, \quad \forall x \in \omega_k \tag{1}$$

The coefficients (a_k, b_k) are determined by minimizing a regularized least-squares cost function within the window:

$$(a_k, b_k) = \arg\min_{a_k, b_k} \sum_{x \in \omega_k} \left((a_k G(x) + b_k - I(x))^2 + \epsilon a_k^2 \right), \tag{2}$$

where ϵ is a regularization parameter. This optimization problem has a closed-form solution for the coefficients based on the local mean and variance of the guidance and input images. This formulation is both interpretable and differentiable, making it a natural candidate for hybrid neural–classical frameworks.

In our framework, we adopt the Joint Guided Filter as a case study for the filtering module due to its efficiency, edge-preserving property, and differentiable nature. However, we emphasize that our method is not limited to any specific filter. The design generalizes to other differentiable filtering operators, such as Bilateral Filters, Domain Transform, and even anisotropic diffusion, so long as the filtering operation can be parameterized and differentiated through. By leveraging classical filters for the image synthesis stage and using neural networks solely for parameter prediction, our framework maintains the transparency and efficiency of classical methods while improving adaptability through learned representations. This design also improves robustness to quantization, as the precision-sensitive task of image synthesis is handled by the mathematically stable classical filter.

2.2. Lightweight Neural Networks

The need for compact and efficient models on resource-constrained devices has driven significant research into lightweight neural architectures for on-device AI. Representative examples includeMobileNet v1 and v2 [8,17], ShuffleNet v1 and v2 [9,18], and EfficientNet v1 [10], which leverage design principles such as depthwise separable convolutions, channel shuffling, or neural architecture search to significantly reduce computational cost and the number of parameters. These models demonstrate that with careful architectural design, a favorable trade-off between accuracy and efficiency is achievable. For instance, depthwise separable convolution decomposes standard convolutions into depthwise and pointwise operations, reducing complexity from $O(K^2 \cdot C_{\text{in}} \cdot C_{\text{out}})$ to $O(K^2 \cdot C_{\text{in}} + C_{\text{in}} \cdot C_{\text{out}})$, where K is the kernel size and $C_{\text{in}}, C_{\text{out}}$ are the input and output channels, respectively.

Recent task-specific lightweight networks [19–21], such as U-Net variants and transformers, often trade expressive capacity for efficiency to enable real-time applications. Despite these innovations, end-to-end neural solutions might still impose non-trivial memory and compute demands, especially when deployed on lower-end devices or operated under aggressive quantization.

To address these limitations, our framework takes a hybrid design approach: it leverages a compact neural network solely for estimating a low-dimensional parameter map, delegating the more compute-intensive image synthesis to a classical filter. This design not only reduces the computational cost compared to end-to-end neural network approaches but also enables fully trustworthy AI system by eliminating server-side assistance.

2.3. Distinction from End-to-End Hybrid Filtering

Several works have explored hybrid neural–classical filtering, with the Fast End-to-End Trainable Guided Filter (GFN) [16] being a prominent example. While both GFN and our framework leverage the Guided Filter, our objectives and technical approaches are fundamentally different, making them orthogonal contributions to the field. The primary goal of GFN, as its name suggests, is to create a fast and fully differentiable filtering layer that can be integrated into larger end-to-end networks for high-performance tasks like joint upsampling. It focuses on optimizing the speed and memory efficiency of the filtering operation itself, particularly in full-precision (FP32) environments.

In contrast, our framework is explicitly designed to solve the problem of quantization robustness for on-device AI. Our core innovation lies not in the filtering operation but in the parameter prediction stage. We achieve robustness by (1) decoupling parameter estimation from the final, precision-sensitive image synthesis, and (2) introducing a basis-decomposed parameterization that is mathematically proven (Proposition 1) to bound reconstruction errors under quantization. This focus on stability in low-bit environments, which is a critical challenge for trustworthiness, is a problem not explicitly addressed by GFN. Therefore, while GFN provides a powerful tool for fast filtering, our framework offers a novel solution for ensuring the reliability of such systems on resource-constrained hardware.

2.4. Quantization for On-Device Inference

Quantization is a standard technique for deploying neural networks on edge devices by reducing model size and accelerating inference [22,23]. However, unlike tasks such as classification, image processing is highly sensitive to the visual quality degradation caused by low-bit precision [24,25]. Existing mitigation strategies like quantization-aware training (QAT) [26,27] often introduce significant training complexity and data dependencies. Other model compression strategies such as pruning and knowledge distillation have also been explored to reduce model size and inference cost [28]. Nevertheless, these approaches

primarily focus on parameter sparsity or teacher–student training schemes, and are less effective in addressing the quantization-induced instability that motivates our work.

To overcome these limitations, we adopt a fundamentally different strategy: we decouple the learning and synthesis stages. A lightweight neural network predicts a compact parameter map, which can be more robustly quantized, while the final image synthesis is performed by a classical filter operating in the floating-point domain. This structure prevents quantization artifacts from propagating into the output image and inherently improves system reliability. Furthermore, we introduce a basis-decomposed parameterization in which the guidance map is reconstructed as a weighted combination of multiple low-precision sub-maps. This design not only enhances robustness against quantization error but also allows dynamic precision adjustment at runtime, thereby supporting flexible trade-offs between performance and efficiency.

2.5. Basis Expansion and Low-Rank Parameterization

To improve parameter efficiency and quantization robustness, some recent works [29,30] have adopted structured representations like basis expansion or low-rank parameterization. These techniques reduce the parameter of model redundancy by expressing high-dimensional signals with a more compact set of basis functions or low-rank components.

In basis expansion, a target signal can be expressed as a weighted combination of K basis elements:

$$P(x) = \sum_{k=1}^{K} \alpha_k \cdot B_k(x), \tag{3}$$

where $P(x)$ denotes the final parameter map at spatial location x, $\{B_k(x)\}_{k=1}^{K}$ are basis functions, and $\{\alpha_k\}$ are their corresponding weights. This formulation allows the learning problem to focus on predicting a compact set of coefficients α_k rather than entire dense parameters.

Low-rank parameterization similarly constrains model weights or intermediate representations to lie in a lower-dimensional subspace. A common approach is matrix factorization, where a weight matrix $W \in \mathbb{R}^{m\times n}$ is approximated as

$$W \approx U \cdot V, \quad U \in \mathbb{R}^{m\times r}, \quad V \in \mathbb{R}^{r\times n}, \quad \text{with } r \ll \min(m,n), \tag{4}$$

reducing the number of parameters and operations from $O(mn)$ to $O(r(m+n))$. Such decompositions are widely used for compressing large models in vision transformers and convolutional networks.

Mathematically, basis expansion can be interpreted as a form of low-rank approximation in the functional domain. If we consider the parameter map $P(x)$ as a vector in a high-dimensional space, our formulation restricts this vector to lie within a low-dimensional subspace spanned by the K basis functions $\{B_k(x)\}$. Specifically, by representing the map as $P(x) = \sum_{k=1}^{K} \alpha_k B_k(x)$, we are effectively performing a rank-K approximation of the signal.

While this structural constraint is powerful for general model compression, our framework leverages it in a unique manner. Instead of compressing an entire network, we strategically apply this basis-decomposed parameterization only to the guidance map that controls the classical filter. This targeted approach is what yields the key advantages for on-device deployment: it ensures the neural component remains compact and quantization robust, while also enabling the runtime-adaptive precision that is crucial for building flexible and trustworthy on-device AI systems.

3. Proposed Method

In this section, we present our quantization-robust parameterized hybrid neural filtering framework, that is designed to address the challenges of trustworthy on-device AI. Our method is built upon two core assumptions, which we will explain in the following subsections. First, we introduce the overall hybrid filtering framework, which decouples parameter estimation from image synthesis to enhance efficiency and interpretability. Second, we describe our basis-decomposed parameterization, a novel strategy to provide inherent robustness against quantization and enable runtime-adaptive precision. To demonstrate the effectiveness of the proposed approach, we choose RGB-guided depth map super-resolution as a representative task since it is a well-studied problem [31–33] using linear filtering, the Guided Filter [11], and has practical applications in mobile environments, such as Bokeh effect generation and 3D reconstruction. However, the proposed framework is general and applicable to any task that can be formulated with any differentiable filtering.

3.1. Overall Framework: A Hybrid Neural Filtering Approach

Our proposed method is built upon a hybrid neural filtering framework that decouples the complex task of image synthesis from the data-driven task of parameter estimation. As illustrated in Figure 1, the proposed approach represents distinct responsibilities to a lightweight neural network and a classical, mathematically defined filter. The overall process can be formulated as

$$I_{out} = \mathcal{F}(I_{in}, G, P(x)), \tag{5}$$

where I_{in} is the input image to be processed (e.g., a low-resolution depth map), G is a guidance image (e.g., a high-resolution RGB image), and $\mathcal{F}$ represents a classical differentiable filter. The behavior of this filter is controlled by a spatially varying parameter map $P(x)$, which is predicted and post-processed by a lightweight neural network $\mathcal{N}$:

$$P(x) = \mathcal{N}(I_{in}, G; \theta), \tag{6}$$

where θ denotes the learnable weights of the network.

For the differentiable filter $\mathcal{F}$ using the Guided Filter, the Guided Filter assumes a local linear model between the guidance image G and the output I_{out} within a local window ω_k:

$$I_{out}(i) = a_k G(i) + b_k, \quad \forall i \in \omega_k, \tag{7}$$

where (a_k, b_k) are the linear coefficients assumed to be constant in the window ω_k. These coefficients have a closed-form solution derived from minimizing a regularized least-squares objective:

$$a_k = \frac{\frac{1}{|\omega_k|}\sum_{i\in\omega_k} G(i)I_{in}(i) - \mu_k \bar{I}_{in,k}}{\sigma_k^2 + \epsilon_k} \tag{8}$$

$$b_k = \bar{I}_{in,k} - a_k \mu_k, \tag{9}$$

where μ_k and σ_k^2 are the mean and variance of the guidance image G in ω_k, and $\bar{I}_{in,k}$ is the mean of the input image I_{in} in the same window. Crucially, our framework introduces adaptability by allowing the neural network to control the regularization parameter ϵ_k, which dictates the degree of edge-preserving smoothing. We achieve this by defining ϵ_k as a function of the predicted parameter map $P(x)$. Specifically, the network-predicted map $P(x)$ serves as a spatially varying regularization map, and ϵ_k is computed as the average value of $P(x)$ within the window ω_k. This allows the network to learn to apply strong

smoothing (high ϵ_k) in textured or flat regions while preserving sharp edges (low ϵ_k) where needed, guided by the content of both the input and guidance images.

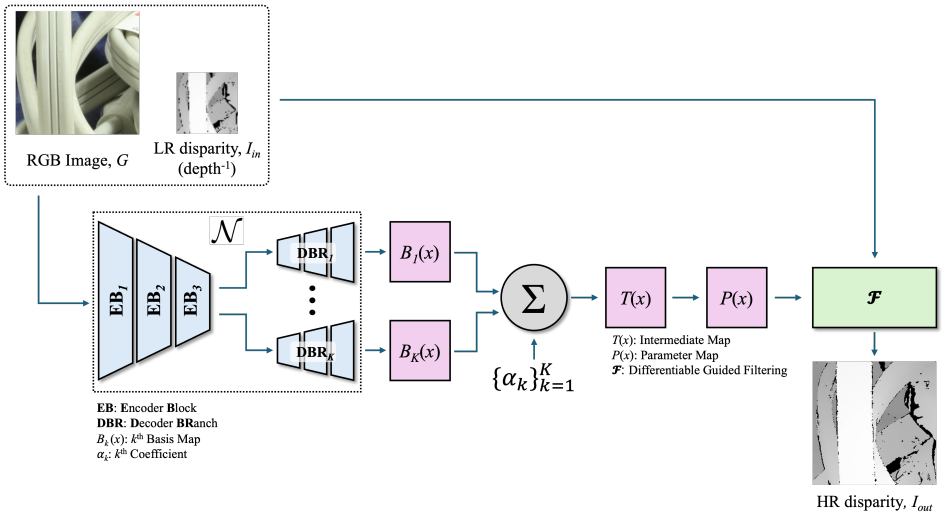


Figure 1. Overview of the proposed hybrid neural filtering framework. Our framework decouples parameter estimation from image synthesis. A lightweight network $\mathcal{N}$ with a shared encoder and K parallel decoders predicts a set of basis maps $\{B_k(x)\}$. These maps are combined with fixed coefficients $\{\alpha_k\}$ to form a parameter map $P(x)$, which guides a classical filter $\mathcal{F}$ to produce the final output I_{out}. Detailed layer configurations and parameter counts for $\mathcal{N}$ are provided in Table 1.

Table 1. Detailed parameter distribution of our proposed network, $\mathcal{N}$. The network consists of a shared encoder, a shared upsampling path in the decoder, and four parallel, lightweight decoder branches for predicting the $B_k(x)$.

Our Network, $\mathcal{N}$		
Component	**Sub-Modules**	**Parameters**
Encoder Path	Encoder Block 1 (16 channels)	9872
	Encoder Block 2 (32 channels)	41,632
	Encoder Block 3 (64 channels)	166,208
	Subtotal	**217,712**
Decoder Path	Shared Skip-Connection Blocks	2608
	Decoder Branch 1 (B_1)	138,145
	Decoder Branch 2 (B_2)	138,145
	Decoder Branch 3 (B_3)	138,145
	Decoder Branch 4 (B_4)	138,145
	Subtotal	**555,188**
Total Trainable Parameters		**772,900**
Total mult-adds (G)		**10.26**

3.2. Quantization Robustness via Basis-Decomposed Parameterization

A primary challenge in on-device deployment is that the output of a neural network is sensitive to quantization. It is important to distinguish our proposed solution, basis decomposition, from traditional model compression techniques. Rather than being a method to reduce model size post hoc, our basis decomposition is an architectural design choice made a priori to ensure the network is inherently robust to the subsequent bit-precision reduction of quantization. To achieve this, we introduce a novel parameterization where, instead of predicting a single high-precision parameter map directly, our framework

predicts a set of K basis maps. These maps are then combined using pre-defined, fixed coefficients to form the final parameter map as formulated below:

$$P(x) = \sum_{k=1}^{K} \alpha_k \cdot B_k(x), \tag{10}$$

where $\{B_k(x)\}_{k=1}^{K}$ are the basis maps predicted by a neural network $\mathcal{N}$, and $\{\alpha_k\}_{k=1}^{K}$ is a set of fixed scalar coefficients.

3.2.1. Network Architecture for Basis Prediction

The core of our parameter estimation stage is the lightweight neural network, $\mathcal{N}$, which is designed for efficient on-device inference. As illustrated in Figure 1, our network $\mathcal{N}$ adopts a one-encoder, multi-decoder architecture. A single, shared encoder, designed with residual convolutional blocks, takes the concatenated input image I_{in} and guidance image G to extract a compact feature representation. This shared representation is then fed into K parallel, lightweight decoders. Each decoder, Decoder_k, is specialized to predict a single corresponding basis map, $B_k(x)$. This design allows the network to learn a disentangled representation for each basis component while maintaining high computational efficiency by sharing the feature extraction backbone.

3.2.2. Parameter Map Composition

The K basis maps predicted by the network are combined to form the final, spatially varying regularization map $P(x)$ for the Guided Filter. This composition involves two key steps. First, the basis maps are combined via a weighted summation using a set of fixed coefficients $\{\alpha_k\}$, which are structured as a geometric progression to create a hierarchical representation:

$$\alpha_k = \beta^{-(k-1)}, \quad \text{for } k = 1, \ldots, K, \tag{11}$$

where β is the base of the progression (e.g., $\beta = 2$ in our experiments). This choice is inspired by the principle of positional notation used in number systems. Our framework adapts this concept to the functional domain: the fixed coefficients $\{\alpha_k\}$ act as pre-defined place-value weights, and the neural network learns to predict the corresponding maps $\{B_k(x)\}$. This formulation transforms the unstructured task of regressing a single, continuous-valued map into a more constrained problem of predicting a set of basis maps that function like bit-planes. This structured approach provides a more stable learning target and encourages a coarse-to-fine representation, as the first few basis maps naturally learn to capture the most critical, large-scale information. The weighted sum is then normalized to produce an intermediate map $T(x) \in [0, 1]$:

$$T(x) = \frac{\sum_{k=1}^{K} \alpha_k B_k(x)}{\sum_{k=1}^{K} \alpha_k}. \tag{12}$$

Second, to address the non-linear sensitivity of the Guided Filter's regularization parameter ϵ, we map the intermediate linear map $T(x)$ to the final parameter map $P(x)$ using a log-uniform transformation:

$$P(x) = \epsilon_{\min} \left(\frac{\epsilon_{\max}}{\epsilon_{\min}} \right)^{T(x)}, \tag{13}$$

where $[\epsilon_{\min}, \epsilon_{\max}]$ defines the target range for the regularization parameter (e.g., $[10^{-6}, 10^{-2}]$).

3.2.3. Analysis of Quantization Effects

For on-device deployment, the neural network $\mathcal{N}$ is quantized, which results in the prediction of quantized basis maps, $\hat{B}_k(x)$. The final parameter map is then reconstructed at runtime following the same composition process. First, an intermediate quantized map $\hat{T}(x)$ is formed:

$$\hat{T}(x) = \frac{\sum_{k=1}^{K} \alpha_k \hat{B}_k(x)}{\sum_{k=1}^{K} \alpha_k}. \tag{14}$$

This is then transformed to the final quantized parameter map:

$$\hat{P}(x) = \epsilon_{\min} \left(\frac{\epsilon_{\max}}{\epsilon_{\min}} \right)^{\hat{T}(x)}. \tag{15}$$

The resulting quantization error in the parameter map domain is $\Delta P(x) = P(x) - \hat{P}(x)$. This formulation allows us to bound the error as stated in the following proposition.

Proposition 1 (Bounded reconstruction error)**.** *The ℓ_2 norm of the parameter map error, $\|\Delta P\|_2$, is bounded by a scaled, weighted sum of the quantization errors of the individual basis maps:*

$$\|\Delta P\|_2 \leq C \cdot \sum_{k=1}^{K} \alpha_k \cdot \|B_k - \hat{B}_k\|_2, \tag{16}$$

where C is a constant determined by the range $[\epsilon_{\min}, \epsilon_{\max}]$.

Proof. Let $f(t) = \epsilon_{\min}(\epsilon_{\max}/\epsilon_{\min})^t$. The error is $\Delta P(x) = f(T(x)) - f(\hat{T}(x))$. The function $f(t)$ is differentiable and its derivative is bounded over the domain $t \in [0, 1]$. By the Mean Value Theorem, there exists a function $T_c(x)$ with values between $T(x)$ and $\hat{T}(x)$ such that $\Delta P(x) = f'(T_c(x))(T(x) - \hat{T}(x))$. Let $L = \sup_{t\in[0,1]} |f'(t)|$ be the Lipschitz constant of f. Then, $|\Delta P(x)| \leq L|T(x) - \hat{T}(x)|$. Integrating over the spatial domain and applying the triangle inequality yields the following derivation:

$$\begin{aligned}
\|\Delta P\|_2 &\leq L \cdot \|T - \hat{T}\|_2 \\
&= L \cdot \left\| \frac{\sum_{k=1}^{K} \alpha_k (B_k - \hat{B}_k)}{\sum_{k=1}^{K} \alpha_k} \right\|_2 \\
&\leq \frac{L}{\sum_{k=1}^{K} \alpha_k} \sum_{k=1}^{K} \|\alpha_k (B_k - \hat{B}_k)\|_2 \\
&= \frac{L}{\sum_{k=1}^{K} \alpha_k} \sum_{k=1}^{K} |\alpha_k| \cdot \|B_k - \hat{B}_k\|_2.
\end{aligned}$$

Letting $C = L/(\sum_{k=1}^{K} \alpha_k)$, the proposition holds. □

This proposition reveals a crucial insight: the total reconstruction error is controlled by the fixed, pre-defined coefficients $|\alpha_k|$. This mathematical foundation provides two significant benefits for trustworthy on-device AI. First, it enhances quantization robustness. The network can learn to distribute information across the basis maps according to the importance dictated by the magnitude of their corresponding α_k. As shown by Proposition 1, this structure actively suppresses the impact of quantization errors from the less significant basis maps on the final output. Second, as illustrated in Figure 2, this structure enables runtime-adaptive precision. By reconstructing the map using only a subset of the first $K' < K$ basis maps, a faster, lower-fidelity approximation can be obtained without any need for retraining.

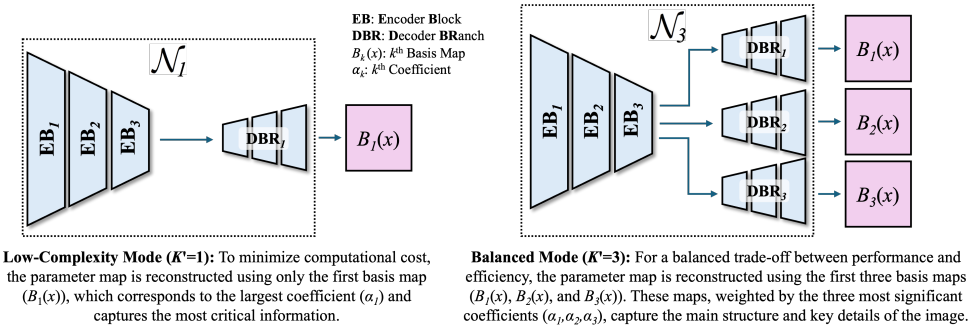


Figure 2. Illustration of our runtime-adaptive precision mechanism. By varying the number of active parallel decoders (K'), a single trained model can operate at different complexity levels to flexibly trade off performance and computational cost without retraining.

4. Experiments

In this section, we conduct a series of experiments to evaluate the effectiveness of the proposed framework. We first describe the experimental setup, including datasets, loss functions, and implementation details. We then present quantitative and qualitative comparisons against conventional methods on the task of RGB-guided depth map super-resolution.

4.1. Datasets Setup

For training the proposed framework, Middlebury stereo datasets [34–38] were used. These datasets are widely adopted for disparity-related tasks (disparity being the inverse of depth), providing high-quality RGB–disparity pairs captured under controlled indoor conditions. The data includes a variety of scenes with diverse textures, lighting conditions, and geometric structures, making them suitable for evaluating the performance of stereo estimation and disparity refinement.

Unlike RGB images, disparity (or depth) maps often contain large textureless regions, which can make the task of disparity super-resolution artificially easy. To mitigate this and ensure our model is trained on challenging examples, we employ a stratified sampling strategy based on depth variation. Specifically, we first generate a pool of 30,000 random 256×256 patches from all available RGB–disparity pairs in the Middlebury Stereo collection. For each patch, we compute the standard deviation of its disparity values and cluster the patches into three groups: high, medium, and low variance.

To construct our dataset, we adopt a different sampling strategy for training versus validation and testing. The training set is created by randomly sampling exclusively from the high- and medium-variance groups, forcing the model to learn from more complex and structured regions. In contrast, the validation and test sets are sampled randomly from all three clusters to ensure that our evaluation reflects a realistic distribution of scenes, including simpler, textureless areas. For each selected patch, the original cropped disparity map is used as the ground truth (D_{HR}). The corresponding low-resolution input (D_{LR}) is generated by applying bicubic downsampling to the ground truth, followed by bicubic upsampling back to the original 256×256 resolution. The entire set of patches is divided into training (70%), validation (15%), and testing (15%) sets.

4.2. Loss Functions

The proposed hybrid framework is trained end-to-end by optimizing a composite loss function designed to enforce reconstruction fidelity, structural similarity, and perceptual quality. The total loss $\mathcal{L}$ is a weighted sum of multiple components:

$$\mathcal{L}(\theta) = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{grad}\mathcal{L}_{grad} + \lambda_{lap}\mathcal{L}_{lap}, \quad (17)$$

where λ_{rec}, λ_{grad}, and λ_{lap} are weighting hyperparameters for each loss term.

4.2.1. Reconstruction Loss

For the primary data fidelity term, we employ the Charbonnier loss, a smooth variant of the L1 loss that is less sensitive to outliers:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\|I_{out}^{(i)} - I_{gt}^{(i)}\|_2^2 + \epsilon^2}. \tag{18}$$

4.2.2. Structural and Perceptual Losses

To better preserve fine details and sharp edges, we incorporate a gradient consistency loss, $\mathcal{L}_{grad}$, and a high-frequency loss, $\mathcal{L}_{lap}$. These are defined as

$$\mathcal{L}_{grad} = \|\nabla I_{out} - \nabla I_{gt}\|_1, \tag{19}$$

$$\mathcal{L}_{lap} = \|\Delta I_{out} - \Delta I_{gt}\|_1, \tag{20}$$

where ∇ and Δ denote the Sobel gradient magnitude and Laplacian operators, respectively.

4.3. Implementation Details

Our proposed framework is implemented in PyTorch 2.8.0 and trained on the Middlebury dataset as described above . The detailed architecture of our network is summarized in Table 1. To maximize parameter efficiency, we employ a shared-encoder and parallel-decoder structure. The shared encoder is designed to be relatively deep to extract a rich feature representation that is common to all parallel decoder branches. Conversely, each of the four parallel decoders is intentionally designed to be lightweight. For our main experiments, the number of decoders, K, is empirically set to 4, as this provides the best trade-off between performance and complexity on our validation set. Furthermore, to enhance generalization and reduce the number of parameters, the upsampling convolution operations within the decoder path are also shared across all branches. This design provides an additional advantage for runtime flexibility; when operating on highly resource-constrained devices, a subset of the decoders can be activated, further reducing the total number of active parameters.

The network predicts a total of $K = 4$ basis maps, which is adjustable. A β for α_k is set to 2. The model is trained for 500 epochs using the Adam optimizer with an initial learning rate of 10^{-4}, which is decayed according to a cosine annealing schedule. The loss weighting parameters are set to $\lambda_{rec} = 1.0$, $\lambda_{grad} = 0.5$, and $\lambda_{lap} = 0.5$. Due to the inherent robustness of our parameterized approach, which results in a negligible performance gap between FP32 and INT8 precision, our model is trained natively in an 8-bit environment to reflect realistic on-device conditions. All experiments are conducted on a single NVIDIA RTX 5090 GPU.

As a baseline for comparison, we utilize a standard U-Net that achieves comparable performance to our proposed framework in a vanilla end-to-end setting. This network consists of a conventional single-encoder, single-decoder architecture. The detailed parameter distribution is presented in Table 2.

Table 2. Parameter distribution of the baseline U-Net.

Baseline, BaseNet		
Component	**Sub-Modules**	**Parameters**
Encoder	Encoder Block 1 (32 channels)	19,680
	Encoder Block 2 (64 channels)	92,352
	Encoder Block 3 (128 channels)	369,024
	Subtotal	**481,056**
Decoder	Decoder Block 1 (128→64 ch)	426,304
	Decoder Block 2 (64→32 ch)	106,656
	Final Blocks (32→1 ch)	29,121
	Subtotal	**562,081**
Total Trainable Parameters		**1,043,137**
Total mult-adds (G)		**12.53**

4.4. Ablation Study on Runtime-Adaptive Precision

To validate the flexibility of our framework, we first evaluate its performance by varying the number of active decoder branches (K) from 1 to 4. This corresponds to adjusting the number of basis maps used for reconstruction at runtime. The results are summarized in Table 3.

The results clearly demonstrate the effectiveness of our runtime-adaptive precision mechanism. As the number of active decoders (K) increases from 1 to 4, the PSNR shows a corresponding monotonic improvement from 27.17 dB to 27.22 dB, at a modest and predictable increase in parameters and computational cost. This confirms that a single trained model can be flexibly deployed to meet different performance and efficiency requirements, a key feature for on-device applications.

Table 3. Quantitative comparison of performance and complexity on the Middlebury dataset. The best results are highlighted in **bold** and the second-best results are underlined.

Method	**# Decoders (K)**	**Parameters**	**MACs (G)**	**PSNR (dB)**	**MSE**
BaseNet	N/A	1.04 M	12.53	**27.40**	0.013941
Ours, $\mathcal{N}$	1	0.43 M	8.40	27.17	0.004979
Ours, $\mathcal{N}$	2	0.52 M	9.02	27.18	0.004968
Ours, $\mathcal{N}$	3	0.62 M	9.64	27.20	0.004950
Ours, $\mathcal{N}$	4	0.77 M	10.26	27.22	**0.004928**

4.5. Stability Analysis: Hybrid vs. End-to-End Approach

A more crucial insight is revealed by comparing the stability of our hybrid approach against the end-to-end *BaseNet*, using the PSNR and MSE metrics from Table 3. While the *BaseNet* achieves the highest PSNR (27.40 dB), its MSE (0.013941) is approximately 2.8 times higher than that of our full model (0.004928).

Since PSNR reflects average perceptual quality while MSE is highly sensitive to large, localized errors (outliers), this discrepancy suggests that the *BaseNet*, despite its high average performance, is prone to producing significant errors in challenging regions. In contrast, our framework's consistently low MSE across all configurations indicates superior stability and reliability. This ability to avoid catastrophic failures is a cornerstone of a Trustworthy AI system, making our hybrid approach a more robust and predictable solution for practical on-device applications.

4.6. Ablation Study on Architectural Efficiency

To investigate the generality of our framework and address the feedback on mobile-optimized networks, we conducted an additional ablation study. We replaced the standard convolutions in our backbone architecture with depthwise separable convolutions (DSC) [8], a key technique used in efficient architectures like MobileNet.

DSC factorizes a standard convolution into a depthwise and a pointwise operation, which can significantly reduce parameters and computational cost. The results of this modification, which we term $\mathcal{N}'$, are presented in Table 4. This experiment demonstrates that our proposed hybrid parameterization is not tied to a specific backbone architecture and can be flexibly combined with other efficiency-enhancing techniques for further optimization.

Table 4. Actual parameter distribution of the network after applying depthwise separable convolutions. The table compares the new lightweight model ($\mathcal{N}'$) against the original standard model ($\mathcal{N}$).

Standard Conv ($\mathcal{N}$) vs. Depthwise Separable Conv ($\mathcal{N}'$)			
Component	**Sub-Modules**	**Parameters**	**Reduction**
Encoder Path	Encoder Block 1 (16 ch)	1848	−81.3%
	Encoder Block 2 (32 ch)	6208	−85.1%
	Encoder Block 3 (64 ch)	21,632	−87.0%
	Subtotal	**29,688**	**−86.4%**
Decoder Path	Shared Skip-Connection Blocks	2800	+7.4%
	Decoder Branches (4×) (17,633 per branch)	70,532	−87.2%
	Subtotal	**73,332**	**−86.8%**
Total Trainable Parameters		**103,020**	**−86.7%**
Total mult-adds (G)		**1.71 G**	**−83.3%**

To examine the real-world feasibility of our proposed efficiency, the optimized $\mathcal{N}'$ model was deployed on a mobile AP processor (Qualcomm Snapdragon 865). Table 5 presents the measured average inference latency and peak memory footprint, obtained under the same experimental conditions as the PC-based experiments.

Table 5. On-device inference latency and memory footprint measured on Qualcomm Snapdragon 865.

Precision	**Activations (MB)**	**Params (MB)**	**Estimated Peak (MB)**	**Latency (ms)**
FP32	28.1	0.41	39.8	35.2
INT8	8.4	0.10	14.3	14.8

4.7. Analysis of Quantization Robustness

A core claim of our work is that the proposed framework is inherently robust to quantized inference. To verify this, we compare the performance of our models ($\mathcal{N}$ and the DSC-variant $\mathcal{N}'$) against the *BaseNet* under post-training quantization. We quantize all models from full precision (FP32) to 8-bit integers (INT8) and measure the performance degradation (ΔPSNR). To evaluate generalizability, the experiments are conducted on both the Middlebury and, the more challenging, real-world NYU Depth V2 datasets.

The results, presented in Table 6, strongly validate our approach. On the Middlebury dataset, the standard *BaseNet* suffers a severe performance drop of 1.5603 dB after quantization, highlighting the well-known sensitivity of end-to-end models. In stark contrast, our proposed model, $\mathcal{N}$, exhibits a negligible degradation of only 0.0044 dB. This provides powerful empirical evidence for the effectiveness of our basis-decomposed parameterization. Furthermore, our DSC-variant, $\mathcal{N}'$, also shows a minimal drop of just 0.0836 dB,

confirming that the robustness stems from our framework's architecture, not a specific convolutional backbone.

The experiments on the NYU Depth V2 dataset further confirm these findings. While the performance degradation is slightly higher across all models on this real-world dataset, the relative trend holds. The performance of *BaseNet* drops by a significant 0.81 dB, whereas our models, $\mathcal{N}$ and $\mathcal{N}'$, degrade by only 0.14 dB and 0.13 dB, respectively. These results demonstrate that our framework's superior quantization robustness is not limited to a single dataset but generalizes effectively, making it a reliable solution for practical on-device deployment.

Table 6. Performance Quantitative comparison for 8-bit inference. The best results are highlighted in **bold** and the second-best results are underlined.

Middlebury Datasets				NYU Depth V2 Datasets			
Method	**Precision**	**PSNR**	**ΔPSNR**	**Method**	**Precision**	**PSNR**	**ΔPSNR**
BaseNet	FP32 INT8	**27.4031** 25.8428	1.5603	*BaseNet*	FP32 INT8	31.6097 30.7997	0.8100
$\mathcal{N}$ $(K = 4)$	FP32 INT8	27.2245 27.2201	**0.0044**	$\mathcal{N}$ $(K = 4)$	FP32 INT8	32.1278 31.9839	0.1439
$\mathcal{N}'$ $(K = 4)$	FP32 INT8	26.1132 26.0296	0.0836	$\mathcal{N}'$ $(K = 4)$	FP32 INT8	31.4593 31.3288	**0.1305**

4.8. Qualitative Analysis

To visually validate the stability and quantization robustness of our framework, we provide qualitative comparisons in Figures 3 and 4.

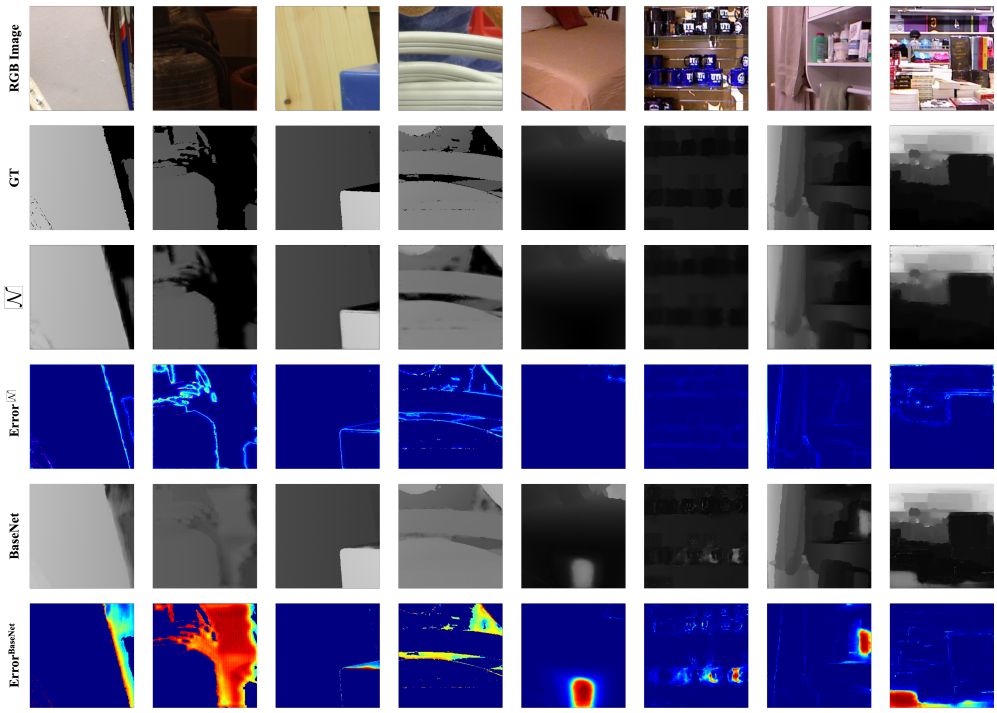


Figure 3. Qualitative comparison of depth map super-resolution on sample data selected from the Middlebury and NYU Depth V2 datasets. For each example, we show the RGB image, Ground Truth (GT), the result of our method ($\mathcal{N}$), its corresponding error map (Error$^{\mathcal{N}}$), the result of the baseline (BaseNet), and its error map (Error$^{\text{BaseNet}}$).

Figure 3 compares final outputs on challenging cases. The end-to-end *BaseNet* fails to preserve sharp boundaries and often introduces blurry artifacts as shown by its bright and spatially widespread error map. In contrast, our approach reconstructs fine details with fewer localized errors, demonstrating much cleaner and more stable results. This supports the higher MSE performance in Table 3, showing that our approach achieves more stable results.

Figure 4 illustrates the effect of 8-bit quantization. Our approach produces nearly identical outputs before and after quantization ($\mathcal{N}_{\text{FP32}}$ vs. $\mathcal{N}_{\text{INT8}}$), yielding an almost black quantization error map that confirms its robustness. In contrast, *BaseNet* exhibits visible degradation and large structured errors, indicating that our hybrid, parameter-decoupled design ensures stable performance under quantization.

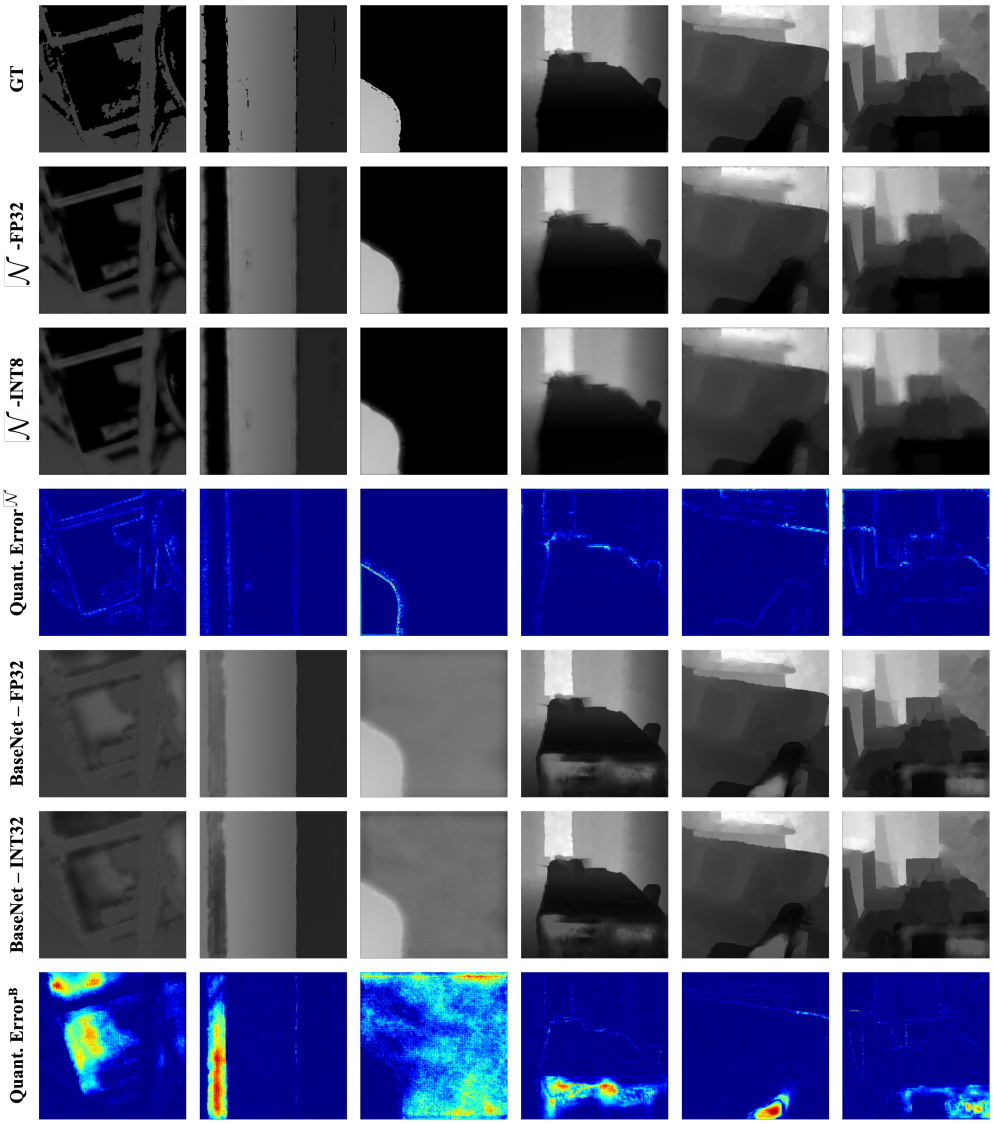


Figure 4. Visual comparison of quantization robustness on sample data from the Middlebury and NYU Depth V2 datasets. We show the results before (FP32) and after (INT8) quantization, along with the quantization error map (the absolute difference between the two).

4.9. Application to Other Tasks: Matting Mask Refinement

To verify the generalizability of our framework, we conduct an additional experiment on image matting mask refinement. Using the MODNet dataset [39], we task our model with refining a low-quality alpha matte, using the corresponding RGB image as guidance. The network, using the same efficient $\mathcal{N}'$ architecture, predicts the optimal epsilon map

for the Guided Filter. As shown in the qualitative results in Figure 5, our framework successfully estimates the context-aware filter parameters to produce a high-quality alpha matte with fine details. This experiment confirms that our proposed parameterization scheme is not limited to depth super-resolution but can be effectively applied to other image processing tasks.

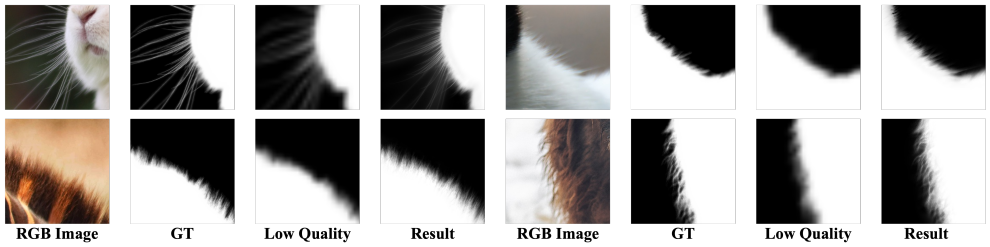


Figure 5. Qualitative examples of matting mask refinement on the MODNet dataset.

5. Discussion

The experimental results demonstrate the effectiveness of the proposed framework. In this section, we discuss the broader implications of our findings, acknowledge the limitations of our current approach, and suggest potential directions for future research.

5.1. Implications for Trustworthy AI

Our framework's contributions to Trustworthy AI extend beyond the core achievement of quantization robustness, offering inherent advantages in system stability and explainability. Unlike monolithic end-to-end networks, which often function as black boxes, the behavior of our hybrid model can be partially interpreted by visualizing the parameter map $P(x)$ predicted by the neural network. This map provides clear insight into the model's decision-making process, showing which regions are designated for detail preservation (low ϵ_k) versus those requiring smoothing (high ϵ_k). This degree of transparency is a significant step toward building more interpretable and, therefore, trustworthy AI systems.

Furthermore, this explainability is complemented by the superior stability demonstrated in our experiments. The consistently low Mean Squared Error (MSE) of our framework, in contrast to the baseline's high score, indicates an ability to avoid the large, localized errors and catastrophic failures common in end-to-end models. This reliability, especially under the hardware constraints of on-device environments, is a cornerstone of a trustworthy system. By successfully combining exceptional quantization robustness, predictable stability, and inherent explainability, our work provides a practical and mathematically grounded foundation for developing the next generation of trustworthy on-device AI applications.

5.2. Limitations and Scope

While our framework demonstrates significant advantages, we acknowledge several limitations and areas for further investigation. A primary challenge lies in the training process due to the indirect nature of the supervision signal. The network predicts a parameter map whose values have a non-linear effect on the final output, and the long backpropagation path from the final image loss can lead to instability. To mitigate this, we employed key strategies, including a composite loss function with structural regularizers ($\mathcal{L}_{grad}$, $\mathcal{L}_{lap}$) and a robust cosine annealing learning rate scheduler, which proved sufficient to ensure stable convergence.

From a theoretical standpoint, our mathematical analysis is centered on the deterministic error bound provided in Proposition 1. We acknowledge that this analysis could be extended. For instance, deriving tighter error bounds by considering the statistical properties of the basis maps, or developing a probabilistic analysis of the quantization

error distribution, could provide deeper insights and further strengthen the framework's theoretical foundations.

At the application level, our framework is subject to the inherent limitations of RGB-guided methods, most notably the potential for texture copying artifacts. This issue arises when a geometrically flat region in the depth map corresponds to a high-frequency texture in the RGB guidance image, causing the model to misinterpret texture edges as depth discontinuities. While less critical for applications like synthetic bokeh, such artifacts can be detrimental to tasks requiring high geometric fidelity, like 3D reconstruction, and this remains an open challenge in the field.

Finally, the scope of our method is intentionally focused on lightweight, efficient performance in resource-constrained environments. Its advantages in parameter efficiency and quantization robustness are most pronounced on mobile and embedded systems. Consequently, in high-resource settings with ample computational capacity, the relative benefits of our design may diminish when compared to large-scale, full-precision models.

5.3. Future Directions

Our framework establishes a solid foundation and opens up several exciting avenues for future research. A promising direction is to address the inherent limitation of texture copying by integrating attention mechanisms. This could enable the model to dynamically modulate the influence of the guidance image, for instance by down-weighting the RGB guide in texture-rich but geometrically flat regions. The framework could be further enhanced through more advanced adaptive basis learning. For instance, the coefficients $\{\alpha_k\}$, which are currently fixed as a geometric progression, could be learned during training or even predicted dynamically by the network for each input. Exploring constraints such as basis orthogonality during learning could also enforce a more disentangled and efficient representation. Another avenue could involve learning a set of universal, fixed basis functions from the entire dataset, allowing the network to predict only a compact set of spatially varying coefficients. This could lead to a more powerful and compact representation.

Theoretically, extending our analysis beyond the current deterministic error bound is an important next step. Developing a probabilistic analysis of the quantization error distribution could provide deeper insights and stronger guarantees for the framework's robustness. Furthermore, to broaden the scope of trustworthiness, the framework could be extended to quantify uncertainty, perhaps by applying techniques like Monte Carlo Dropout to the basis map prediction, which would significantly enhance system reliability in critical applications. Additionally, future work could investigate other facets of trustworthiness, such as fairness, by analyzing the model's performance across diverse demographic or environmental conditions.

Finally, to demonstrate the true on-device viability and generalizability of our approach, future work must involve broader experimental validation. This includes applying the framework to a wider range of image processing tasks, such as denoising or dehazing, and complementing these experiments with the detailed profiling of latency and energy consumption on actual mobile hardware. We believe these future explorations will build upon our work to further advance the development of reliable and resource-efficient on-device AI.

6. Conclusions

In this work, we proposed a novel parameterized hybrid neural filtering framework designed to address the critical challenges of building trustworthy and efficient AI systems for on-device image processing. Our core contributions are twofold. First, by decoupling parameter estimation from image synthesis, our framework significantly enhances parameter

efficiency and stability compared to standard end-to-end models. Second, we introduced a basis-decomposed parameterization, a mathematically grounded approach that provides exceptional robustness against quantization, a mandatory step for on-device deployment. Our experimental results on RGB-guided depth map super-resolution demonstrated that this design not only prevents performance degradation under 8-bit quantization but also enables runtime-adaptive precision, allowing a single model to flexibly adapt to diverse computational constraints. By providing a practical and theoretically sound solution, this work lays a solid foundation for the future development of reliable and resource-efficient AI applications.

Author Contributions: Conceptualization, S.H. and S.J.; methodology, S.H., S.-W.K., S.M. and S.J.; writing—original draft preparation, S.H. and S.J.; writing—review and editing, S.H., S.-W.K., S.M. and S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in the Middlebury Stereo Datasets at https://vision.middlebury.edu/stereo/data/, accessed on 20 August 2025.

Acknowledgments: This paper was supported by Konkuk University in 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Moor, M.; Banerjee, O.; Abad, Z.S.H.; Krumholz, H.M.; Leskovec, J.; Topol, E.J.; Rajpurkar, P. Foundation models for generalist medical artificial intelligence. *Nature* **2023**, *616*, 259–265. [CrossRef] [PubMed]
2. Lu, Y.; Lin, Y.; Wu, H.; Luo, Y.; Zheng, X.; Xiong, H.; Wang, L. Priors in deep image restoration and enhancement: A survey. *arXiv* **2022**, arXiv:2206.02070.
3. Ji, S.W.; Lee, J.; Kim, S.W.; Hong, J.P.; Baek, S.J.; Jung, S.W.; Ko, S.J. XYDeblur: Divide and Conquer for Single Image Deblurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17421–17430.
4. Li, B.; Qi, P.; Liu, B.; Di, S.; Liu, J.; Pei, J.; Yi, J.; Zhou, B. Trustworthy AI: From principles to practices. *ACM Comput. Surv.* **2023**, *55*, 1–46. [CrossRef]
5. Fu, Z.; Wang, W.; Huang, Y.; Ding, X.; Ma, K.K. Uncertainty Inspired Underwater Image Enhancement. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 465–482.
6. Kaur, D.; Uslu, S.; Rittichier, K.J.; Durresi, A. Trustworthy artificial intelligence: A review. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–38. [CrossRef]
7. Liang, W.; Tadesse, G.A.; Ho, D.; Fei-Fei, L.; Zaharia, M.; Zhang, C.; Zou, J. Advances, challenges and opportunities in creating data for trustworthy AI. *Nat. Mach. Intell.* **2022**, *4*, 669–677. [CrossRef]
8. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
9. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
10. Tan, M.; Le, Q. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
11. He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1397–1409. [CrossRef] [PubMed]
12. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), Bombay, India, 7 January 1998; pp. 839–846.
13. Jing, J.; Liu, S.; Wang, G.; Zhang, W.; Sun, C. Recent advances on image edge detection: A comprehensive review. *Neurocomputing* **2022**, *503*, 259–271. [CrossRef]
14. Zhang, Q.; Shen, X.; Xu, L.; Jia, J. Rolling Guidance Filter. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 815–830.
15. Wu, L.; Fang, L.; Yue, J.; Zhang, B.; Ghamisi, P.; He, M. Deep bilateral filtering network for point-supervised semantic segmentation in remote sensing images. *IEEE Trans. Image Process.* **2022**, *31*, 7419–7434. [CrossRef] [PubMed]

16. Wu, H.; Zheng, S.; Zhang, J.; Huang, K. Fast End-to-End Trainable Guided Filter. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1838–1847.
17. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
18. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
19. Liao, W.; Zhu, Y.; Wang, X.; Pan, C.; Wang, Y.; Ma, L. Lightm-unet: Mamba assists in lightweight unet for medical image segmentation. *arXiv* **2024**, arXiv:2403.05246.
20. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking Vision Transformers for Mobilenet Size and Speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 16889–16900.
21. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
22. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2022; pp. 291–326.
23. Weng, O. Neural network quantization for efficient inference: A survey. *arXiv* **2021**, arXiv:2112.06126.
24. Seo, Y.; Kim, I.; Lee, J.; Choi, W.; Song, S. On quantization of convolutional neural networks for image restoration. *Electron. Imaging* **2022**, *34*, 1–5. [CrossRef]
25. Chen, Y.; Qin, H.; Zhang, Z.; Magno, M.; Benini, L.; Li, Y. Q-mambair: Accurate quantized mamba for efficient image restoration. *arXiv* **2025**, arXiv:2503.21970.
26. Park, E.; Yoo, S.; Vajda, P. Value-Aware Quantization for Training and Inference of Neural Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 580–595.
27. Tailor, S.A.; Fernandez-Marques, J.; Lane, N.D. Degree-quant: Quantization-aware training for graph neural networks. *arXiv* **2020**, arXiv:2008.05000.
28. Liu, S.; Ha, D.S.; Shen, F.; Yi, Y. Efficient neural networks for edge devices. *Comput. Electr. Eng.* **2021**, *92*, 107121. [CrossRef]
29. Uss, M.; Yermolenko, R.; Shashko, O.; Kolodiazhna, O.; Safonov, I.; Savin, V.; Yeo, Y.; Ji, S.; Jeong, J. Predicting High-Precision Depth on Low-Precision Devices Using 2D Hilbert Curves. In Proceedings of the Forty-Second International Conference on Machine Learning, Vancouver, BC, Canada, 13–19 July 2025.
30. Zhang, J.; Zhao, F. Decomposition ascribed synergistic learning for unified image restoration. *arXiv* **2023**, arXiv:2308.00759.
31. Li, Y.; Li, Z.; Zheng, C.; Wu, S. Adaptive weighted guided image filtering for depth enhancement in shape-from-focus. *Pattern Recognit.* **2022**, *131*, 108900. [CrossRef]
32. Ali, U.; Mahmood, M.T. Depth enhancement through correlation-based guided filtering in shape from focus. *Microsc. Res. Tech.* **2021**, *84*, 1368–1374. [CrossRef] [PubMed]
33. Zhong, Z.; Liu, X.; Jiang, J.; Zhao, D.; Ji, X. Guided depth map super-resolution: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–36. [CrossRef]
34. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [CrossRef]
35. Scharstein, D.; Szeliski, R. High-Accuracy Stereo Depth Maps Using Structured Light. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; Volume 1, p. I.
36. Scharstein, D.; Pal, C. Learning Conditional Random Fields for Stereo. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
37. Hirschmuller, H.; Scharstein, D. Evaluation of Cost Functions for Stereo Matching. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
38. Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nešić, N.; Wang, X.; Westling, P. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In Proceedings of the German Conference on Pattern Recognition, Münster, Germany, 2–5 September 2014; pp. 31–42.
39. Ke, Z.; Sun, J.; Li, K.; Yan, Q.; Lau, R.W. Modnet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36; pp. 1140–1147.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.