Zero-Shot Burst Restoration via Diffusion MAP Inference with Poisson-Gaussian Noise Likelihood

Ziyao Yi* **Luca Savant Aira*** **Diego Valsesia** **Tiziano Bianchi** **Enrico Magli**
Politecnico di Torino
{name.surname}@polito.it

Abstract

Burst image restoration aims to reconstruct a high-quality image from a sequence of low-quality frames. While state-of-the-art supervised methods achieve excellent performance, they rely on extensive paired datasets that may be difficult to acquire in practice and often fail to generalize across different camera sensors or real-world noise levels. Zero-shot diffusion-based methods offer a compelling, training-free alternative. However, they typically struggle on burst imaging due to their assumption of known forward operators. In this paper, we introduce ZSBR: Zero-Shot Burst Restoration, a zero-shot diffusion framework for burst restoration that performs MAP-style inference by combining a pretrained diffusion prior with a physically grounded burst formation model. ZSBR accounts for signal-dependent RAW noise by leveraging a continuous Poisson-Gaussian model. Moreover, unlike common diffusion-based methods that assume a fixed operator, ZSBR refines image alignment parameters online by optimizing per-frame affine matrices within the forward model during sampling. To stabilize the resulting optimization, we integrate an adaptive descent scheme into the sampling loop. Moreover, the method is flexible and allows to achieve a desired distortion-perception tradeoff. Experimental results show that our method provides a flexible and robust solution to burst restoration without requiring any paired training example. Code is available at: https://github.com/IPLLab-PoliTO/ZSBR

1 Introduction



LQ DPS [10] ZSBR (Ours) GT

Figure 1: A comparison of zero-shot models. Left to right: A single input frame, outputs from DPS and ZSBR, and the ground truth.

Burst image restoration aims to recover a high-quality image from a short sequence of degraded frames captured in rapid succession. It is important in applications such as mobile photography [12] and satellite imaging [35, 29], where multiple low-quality observations can be combined to improve resolution and reduce noise. Compared with single-image restoration [7, 8, 30, 51], burst restoration

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

can exploit complementary information across frames, but it must also handle inter-frame motion [4], RAW imaging pipelines, and realistic sensor noise [55].

Recent deep learning approaches have delivered impressive progress in burst image restoration, spanning from architectural design [16, 33, 45] to dataset construction [3, 49, 34, 52]. State-of-the-art methods are mostly supervised deep models trained on paired burst datasets. These methods achieve strong performance when the test data matches the training distribution, but they depend on large task-specific datasets and often generalize poorly across sensors, noise levels, or acquisition pipelines. They also offer limited flexibility when one wants to enforce a physically meaningful forward model or a calibrated noise likelihood at test time.

Pretrained diffusion models offer an appealing alternative. They provide strong image priors [25, 31, 13] without requiring task-specific retraining, and their iterative sampling process makes it possible to incorporate explicit measurement constraints during inference, while also being able to favor perception against distortion [5]. Recent works [43] have explored the use of diffusion models for burst super-resolution, but they typically include components requiring supervised training on specific datasets, which may still inherit distribution biases. This is a limitation which motivates us to investigate a new fully zero-shot approach for burst image restoration, as existing zero-shot diffusion methods are not well suited to burst imaging. First, burst restoration involves a complex forward model that includes inverse ISP operations, inter-frame motion, downsampling, and mosaicking. In practice, part of this operator is unknown because the frame-to-frame alignment is not available at test time. Second, RAW burst data is affected by signal-dependent heteroscedastic noise, whereas standard diffusion-based methods usually assume a fixed operator and homoscedastic Gaussian noise. Under these conditions, naive diffusion guidance becomes inaccurate and unstable.

We address these issues with ZSBR, a zero-shot diffusion framework for burst restoration. We formulate inference as a *joint Maximum A Posteriori (MAP) estimation* over both the clean image and the differentiable burst formation model using a pretrained diffusion prior and a Poisson-Gaussian negative log-likelihood (NLL). During sampling, the method jointly refines the image and the per-frame affine alignment parameters and, optionally, noise statistics. To stabilize optimization through this highly non-linear model, we replace fixed guidance scaling with an adaptive descent scheme based on Adam. This yields a flexible training-free method that can handle misalignment, unknown noise statistics, and variable burst length at inference time.

Our main contributions are summarized as follows:

- We propose ZSBR, a zero-shot diffusion method for burst restoration that formulates the task as blind MAP inference with a pretrained diffusion prior and a differentiable physical burst model.
- We develop a joint optimization strategy that refines the restored image, the unknown inter-frame affine transformations, and optionally the sensor noise parameters during sampling, using a Poisson-Gaussian likelihood tailored to RAW burst measurements.
- We expose a simple test-time hyperparameter that exploits ZSBR stochastic nature and enable exploration of the perception-distortion tradeoff [5] by averaging reconstructions from multiple stochastic runs.
- We demonstrate on burst super-resolution and burst denoising that ZSBR is competitive with supervised methods even if it is training-free, while showing cross-domain generalization under the standard color-normalized evaluation protocol, making it attractive when paired data or matched sensor calibration are unavailable.

2 Related Work

Burst restoration includes burst super-resolution and burst denoising. In both cases, the goal is to combine information from multiple frames captured in rapid succession while handling motion, noise, and the RAW imaging pipeline. The problem was first studied in [44], and has since been approached with both model-based methods [17, 2, 41, 37] and learning-based approaches [4, 3, 45, 38].

DBSR [3] is an encoder-decoder architecture with an explicit alignment module and introduced both synthetic and real datasets. Subsequent works differ mainly in how they represent and fuse temporal information: BIPNet [15] aggregates complementary cues into pseudo-burst features, RBSR [50]

uses recurrent fusion over frames. Transformer-based architectures are also widely used. For example, Burstormer [16] introduces a transformer architecture with deformable alignment and cyclic burst sampling to jointly align and fuse multi-frame features for burst super-resolution, denoising, and low-light enhancement; BSRT [33] combines pyramid optical flow-guided deformable alignment with Swin Transformer blocks to robustly align and fuse multi-frame RAW bursts for high-quality super-resolution. More recent works explore Mamba-style sequence models for efficient long-range fusion [45, 14].

Beyond burst super-resolution, burst denoising is also a widely studied task. The method of [32] introduced homography-based alignment followed by fast per-pixel fusion in temporal and spatial domains. The method of [21] proposed a fully convolutional network to process multiple frames jointly. KPN [34] predicted spatially varying kernels to perform both alignment and denoising, and provided a grayscale burst dataset. BPN [52] is a basis prediction network that estimates global basis kernels shared within an image and introduced a synthetic sRGB burst denoising dataset. The method of [39] proposed an end-to-end trainable pipeline that leverages wavelet-derived high-frequency features. EBDNet [36] integrated optical flow estimation with a kernel prediction network in an end-to-end fashion to solve the relative motion issue. Notably, several burst super-resolution models such as BIPNet [15] and Burstormer [16] also achieve strong performance on burst denoising tasks.

Diffusion for burst restoration has been studied by relatively few works. Recent advances like BSRD [43] and its efficient extension E-BSRD [26] have demonstrated significant gains in perceptual quality by combining deterministic burst SR estimates with diffusion-based refinement. Specifically, these methods initialize the diffusion reverse process from an intermediate step using a pre-restored image generated by a deterministic model (e.g. , Burstormer [16] or BIPNet [15]), rather than from pure random noise. These methods require supervised training on paired LR-HR datasets to learn the specific noise scheduling and refinement trajectory. This training dependency, coupled with the need for an external initialization model, can limit adaptability in real-world scenarios where ground truth is unavailable. This motivates our exploration of a zero-shot approach that aims to leverage generative priors without relying on paired training data or external restoration models.

Zero-shot diffusion methods for other inverse problems have advanced rapidly. DDRM [27] derives a closed-form posterior update for linear degradations. ILVR [9] performs low-frequency guidance from a reference image. DDNM [47] restores images by refining the null space projection. The seminal DPS [10] work formulates restoration as a constrained optimization problem and supports non-linear operators. Later DPS-based methods improve performance [40] or efficiency [11], and recent work [53] argues that DPS is better understood as MAP inference than as posterior sampling.

Despite strong progress, existing methods still leave a clear gap. Supervised burst restoration methods depend on paired training data and often inherit the statistics of the training pipeline, while existing diffusion-based burst methods still rely on supervised reconstruction modules. Zero-shot diffusion methods, in contrast, are flexible and training-free, but they are usually designed for much simpler inverse problems. Instead, the burst restoration setting is characterized by the following issues. A first issue is the complex partially-unknown forward operator, as the degradation model includes the non-linear camera processing and inter-frame motion, typically not known at test time. We remark that the inter-frame motion must be estimated from degraded observations, even if a fixed and inaccurate alignment can produce visible artifacts. A second issue is the heteroscedastic RAW noise model. Unlike the constant-variance Gaussian assumption used in most zero-shot diffusion methods, real RAW burst measurements exhibit signal-dependent noise. Ignoring this property leads to poorly calibrated data consistency, especially in over- or under-exposed image regions.

Our method addresses both issues by combining zero-shot diffusion inference with joint alignment refinement and a physically grounded likelihood for RAW burst observations.

3 Methodology

In this section, we introduce ZSBR, our zero-shot diffusion method for burst restoration. Given a RAW burst $\{\mathbf{y}^b\}_{b=1}^{B}$ of noisy and misaligned frames, the goal is to recover the most plausible estimate $\hat{\mathbf{x}}$ of the true unknown clean image $\mathbf{x}^*$. We model the b-th burst frame as

$$\mathbf{y}^b = \mathcal{A}^b(\mathbf{x}^*) + \mathbf{n}^b, \quad b = 1, \dots, B\,, \tag{1}$$

where $\mathbf{n}^b$ is the noise term and $\mathcal{A}^b$ denotes the acquisition pipeline, or *forward operator*.

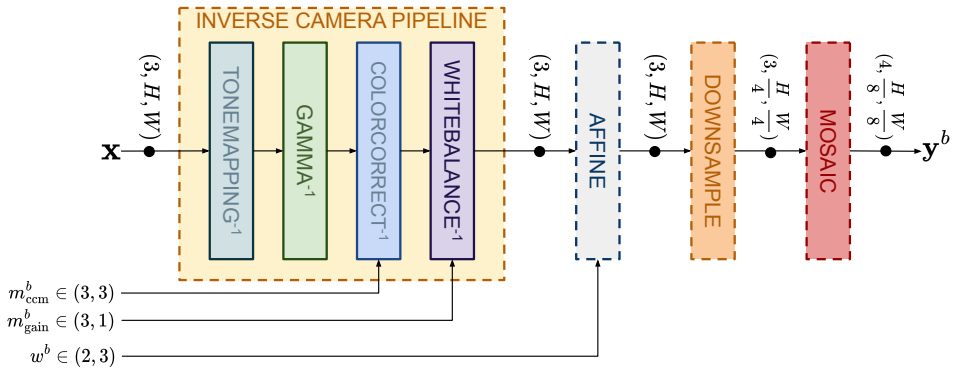


Figure 2: Forward operator for burst formation $\mathcal{A}(\mathbf{x}, w^b, m^b)$. A sRGB image $\mathbf{x}$ is mapped to a RAW burst frame $\mathbf{y}^b$. $\mathbf{x}$ is first processed by the inverse camera pipeline (inverse tone mapping, gamma linearization, inverse color correction, inverse white balancing) using camera-related parameters m^b. The resulting linear image is then warped using the flow induced by w^b, followed by downsampling and mosaicking.

3.1 Forward operator for burst formation

As shown in Fig. 2, the proposed forward operator maps a sRGB image to a burst frame via a composition of (differentiable) sub-operators: the inverse camera pipeline, geometry warping, downsampling and mosaicking. Formally, we factorize the overall forward operator $\mathcal{A}^b$, introduced in Eq. (1), into:

$$\mathcal{A}^b(\mathbf{x}) = \mathcal{A}(\mathbf{x}, w^b, m^b) = \mathcal{M} \circ \mathcal{D} \circ \mathcal{W}_{w^b} \circ \mathcal{C}^{-1}_{m^b}(\mathbf{x}), \tag{2}$$

where we made explicit the dependence of $\mathcal{A}$ on the affine transformation parameters w^b and camera parameters m^b. The sub-operators are defined as follows:

Inverse Camera Pipeline ($\mathcal{C}^{-1}_{m^b}$): this operator inverts the standard camera pipeline, overall mapping sRGB domain into linear domain. It applies inverse tone mapping, gamma linearization, color correction, and inverse white balance. The specific color correction matrices and per-channel white-balance gains provided by the camera parameters m^b are used to ensure color fidelity in the linear sensor space.

Adaptive Affine Estimation ($\mathcal{W}_{w^b}$): handheld burst photography inherently captures misaligned frames. Unlike the camera pipeline parameters m^b, which are known, the exact geometric transformations between burst frames are not known. Luckily, the burst frames are captured in rapid succession so that the misalignment between two given frames of a burst is typically small. Hence we can model it with a global per-frame affine transformation w^b with reasonable accuracy.

Since the performance of a restoration method critically depends on the accuracy of the estimated geometric transformation, we propose an optimization scheme which updates the affine transformations $\{w^b\}$ on the fly during the reverse sampling process. By treating $\{w^b\}$ as an optimization variable, we can iteratively refine the alignment based on the current image estimate, mitigating the accumulation of errors, even if starting from an inaccurate initial estimate. This initialization procedure for w^b is described in detail in the Appendix A and runs as follows. We first independently convert each burst frame from RAW to sRGB by demosaicking, color-correcting, gamma-correcting and tone-mapping. Then, to reduce the effects of false coloring[22], we convert sRGB to grayscale images. Finally we apply PWCNet [42], an available off-the-shelf optical flow estimator, to estimate the camera movements between the reference frame and the rest of the burst. Given the estimated dense optical flow $\mathbf{f}^b$, its affine approximation w^b is computed, in a least-square sense. Without loss of generality, the first burst frame ($b = 1$) is considered as a reference and all other frames are aligned to that.

Downsampling ($\mathcal{D}$) and mosaicking ($\mathcal{M}$): The operator $\mathcal{D}$ simulates the spatial resolution degradation via bilinear downsampling, while $\mathcal{M}$ applies the Bayer pattern to produce RAW sensor data. In the task of burst denoising, the forward operator is simplified by removing mosaicking and downsampling.

Algorithm 1 ZSBR

Input: RAW burst $\{\mathbf{y}^b\}_{b=1}^B$, repetitions N, diffusion steps T, gradient steps K, NLL model $\mathcal{L}_{\text{NLL}}$.
Output: Perception-distortion aware reconstructed sample

for $n = 1$ **to** N **do**
  $\{w^b\}_{b=1}^B \leftarrow$ EstimateAffineTransforms$(\{\mathbf{y}^b\}_{b=1}^B)$
  $\mathbf{x}_T^n \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
  for $t = T$ **to** 1 **do**
    $\mathbf{x} \leftarrow$ SamplerStep$(\mathbf{x}_t^n, s_\theta(\mathbf{x}_t^n, t))$
    Initialize all Adam optimizers.
    for $j = 1$ **to** K **do**
      $\hat{\mathbf{x}} \leftarrow$ Decode (Tweedie $(\mathbf{x}, s_\theta(\mathbf{x}, t), t)$)
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{NLL}}\left(\hat{\mathbf{x}}, \{\mathbf{y}^b, w^b\}_{b=1}^B\right)$
      for $\Theta \in \{\mathbf{x}, \sigma_{\text{read}}, \sigma_{\text{shot}}, w^2, \ldots, w^B\}$ **do**
        $\Theta \leftarrow$ AdamUpdate$(\nabla_\Theta \mathcal{L})$
    $\mathbf{x}_{t-1}^n \leftarrow \mathbf{x}$
return $\frac{1}{N}\sum_{n=1}^N$ Decode $(\mathbf{x}_0^n)$

3.2 Physically-grounded data consistency via NLL

Many solvers for inverse problems [47, 20, 10, 53] enforce data consistency by minimizing an ℓ^2 residual: $\|\mathbf{y} - \mathcal{A}(\mathbf{x})\|_2^2$. This formulation implicitly assumes an i.i.d. Gaussian likelihood with constant variance, $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathcal{A}(\mathbf{x}), \varsigma^2\mathbb{I})$, which rarely holds in real burst RAW measurements where noise is strongly signal-dependent.

In real-world burst scenarios, noise is a composite of photon shot noise and electronic read noise [6, 19, 23], governed by the specific sensor characteristics and exposure settings. To faithfully represent this, the observed frames $\{\mathbf{y}^b\}_{b=1}^B$ are modeled with B independent continuous Poisson-Gaussian distributions:

$$\begin{cases} \mathbf{y}^b = \mathcal{A}^b(\mathbf{x}^*) + \mathbf{n}^b \\ \mathbf{n}^b \sim \mathcal{N}\left(\mathbf{0}, \Sigma^b\right) \\ \Sigma^b = \text{diag}\left(\sigma_{\text{read}}^2 + \sigma_{\text{shot}}^2 \mathcal{A}^b(\mathbf{x}^*)\right) \end{cases} \qquad \forall b = 1, \ldots, B. \tag{3}$$

where $\mathbf{x}^*$ is the unknown clean image. In this way, the likelihood $p(\mathbf{y}|\mathbf{x})$ is:

$$\mathcal{L}_{\text{NLL}}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^B \left[\left(\mathcal{A}^b(\mathbf{x}) - \mathbf{y}^b\right)^T \left(\Sigma^b\right)^{-1} \left(\mathcal{A}^b(\mathbf{x}) - \mathbf{y}^b\right) + \log\det\left(\Sigma^b\right)\right]. \tag{4}$$

where $\Sigma^b = \text{diag}\left(\sigma_{\text{read}}^2 + \sigma_{\text{shot}}^2\, \mathcal{A}^b(\mathbf{x})\right)$ is evaluated at the current estimate $\mathbf{x}$.

In the circumstance where the exact noise parameters σ_{read}^2 and σ_{shot}^2 are unknown, we can treat them as variables and optimize them (in logarithmic space to ensure positivity) jointly with the image estimate. While a fully Bayesian treatment might assign a conjugate prior such as an Inverse-Gamma distribution to these variance components to sample from their posterior, such an approach does not reflect the physical reality of burst photography. For a given camera sensor and exposure setting, the read and shot noise coefficients are fixed, deterministic scalars. Because we are solely interested in discovering their true physical values rather than capturing their statistical uncertainty, full posterior sampling is unnecessary. Instead, seeking their MAP point estimates via direct gradient optimization is mathematically sufficient and physically grounded. The detailed estimation and assessment of the physical recovery procedure is shown in Appendix D.3. This joint optimization formulation allows the model to implicitly adapt to unknown noise statistics, providing a robust guidance signal even in entirely blind settings.

3.3 ZSBR: algorithm and optimization

We now describe the full ZSBR algorithm, our diffusion-based solver for burst restoration under the forward operator of Section 3.1 and the physically-grounded data consistency likelihood of Section 3.2. The overall ZSBR algorithm is summarized in Algorithm 1.

Given an observed burst $\{\mathbf{y}^b\}_{b=1}^B$, we aim to recover a clean image $\hat{\mathbf{x}}$ that matches the natural image distribution and explains the observations through the burst formation process.

A standard way to formalize this goal is via MAP estimation:

$$(\hat{\mathbf{x}}, \{\hat{w}^b\}) = \arg\max_{\mathbf{x}, \{w^b\}} \Big(\log p(\mathbf{x}) + \log p(\{\mathbf{y}^b\} \mid \mathbf{x}, \{w^b\}) \Big), \tag{5}$$

where $\log p(\mathbf{x})$ is an image prior and the likelihood term enforces data consistency. If we were to optimize Eq. (5) directly with gradient ascent on $\mathbf{x}$, a generic update would read:

$$\mathbf{x}_{(k+1)} \leftarrow \mathbf{x}_{(k)} + \eta_k \Big(\nabla_{\mathbf{x}} \log p(\mathbf{x}_{(k)}) + \nabla_{\mathbf{x}} \log p(\{\mathbf{y}^b\} \mid \mathbf{x}_{(k)}, \{w^b\}) \Big), \tag{6}$$

where the prior gradient pushes toward the natural-image manifold, and the likelihood gradient pushes for consistency with observations. There are two difficulties with this approach: the prior gradient is not available in closed form for natural images; the likelihood gradient is not trivial as it involves the iterate $\mathbf{x}_{(k)}$ and not directly the cleanest estimate we could have at iteration k.

Diffusion models address the first issue by learning $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, i.e. , the score function of a sequence of progressively noised distributions $\{p_t\}$. At diffusion step t, a pre-trained network provides an estimate $s_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, which plays the role of the prior gradient term in Eq. (6). Moreover, diffusion models also yield a point estimate of the clean sample at step t, denoted as $\hat{\mathbf{x}}_{0|t}$ addressing the second issue. This estimate is crucial as it allows us to evaluate our physically-grounded loss at an almost clean image, as maximizing the likelihood is equivalent to minimizing $\mathcal{L}_{\mathrm{NLL}}(\mathbf{x}, \{w^b\})$ from Eq. (4):

$$\log p(\{\mathbf{y}^b\} \mid \mathbf{x}_t, \{w^b\}) \;\approx\; \log p(\{\mathbf{y}^b\} \mid \hat{\mathbf{x}}_{0|t}, \{w^b\}). \propto -\mathcal{L}_{\mathrm{NLL}}\big(\hat{\mathbf{x}}_{0|t}, \{w^b\}\big)\,. \tag{7}$$

Combining the diffusion prior (score) with the likelihood approximation above yields a guided reverse process that can be interpreted as performing MAP-style ascent along the diffusion trajectory:

$$\mathbf{x}_{t-1} \;\leftarrow\; \textsc{SamplerStep}(\mathbf{x}_t, s_\theta(\mathbf{x}_t, t)) \;+\; \lambda_t\, \mathbf{g}_t, \qquad \mathbf{g}_t \triangleq -\nabla_{\mathbf{x}_t} \mathcal{L}_{\mathrm{NLL}}\big(\hat{\mathbf{x}}_{0|t}, \{w^b\}\big), \tag{8}$$

where SAMPLERSTEP denotes one step of a standard sampler (e.g., DDIM / DPM-Solver), and the second term injects the data-consistency gradient backpropagated through $\hat{\mathbf{x}}_{0|t}$, the burst operator, and all its sub-operators (warping, inverse ISP, downsampling, mosaicking). Importantly, the burst length B appears only inside the NLL term; therefore, the same procedure supports a variable number of frames at inference time.

In burst restoration, $\mathcal{A}$ is highly non-linear and depends on alignment parameters; consequently, the magnitude and conditioning of $\mathbf{g}_t$ vary drastically across diffusion steps and across spatial/color dimensions. Using a fixed guidance scale λ_t is therefore brittle. To stabilize optimization, we replace the plain scaled gradient step with an adaptive descent rule based on Adam. In particular, we perform $K = 10$ Adam steps per diffusion step (for a total of $T = 50$ diffusion steps) to amortize Adam's internal state updates (momentum and variance estimates), yielding more reliable and well-scaled gradient updates than a single step. This makes the effective guidance step size self-tuning, improves robustness to gradient spikes, and reduces sensitivity to manual hyperparameter selection.

As discussed in Section 3.1, inaccurate alignment causes biased likelihood gradients and can derail the reverse process. We therefore treat the affine transforms $\{w^b\}$, initialized as described in Section 3.1, as optimization variables and update them on the fly during sampling by descending the same NLL.

ZSBR operates in a latent space for efficiency. This choice reflects a trade-off: while pixel-space diffusion models may be better suited for low-level vision tasks, latent diffusion models benefit from training on much larger datasets, resulting in a stronger prior in the zero-shot setting. To keep notation consistent throughout the paper, we continue to write the diffusion variable as $\mathbf{x}_t$ and its clean prediction as $\hat{\mathbf{x}}_{0|t}$. Implementation-wise, $\mathbf{x}_t$ lives in the latent space and $\hat{\mathbf{x}}_{0|t}$ denotes the decoded image used for $\mathcal{A}$ and $\mathcal{L}_{\mathrm{NLL}}$. This does not change the optimization logic above; it only changes where the score model is evaluated. It alternates, at each diffusion step, a standard reverse diffusion update driven by the learned score prior and an adaptive, NLL-based data-consistency descent step, while jointly refining the burst alignments $\{w^b\}$ (and optionally unknown noise parameters in the NLL). Note that the inner optimization loop operates at a fixed diffusion step t: it updates the current latent and auxiliary variables to improve data consistency, but it does not modify the diffusion noise schedule itself. Finally, the NLL gradients over burst frames factorize across b and can be evaluated efficiently via standard backpropagation with batched, parallel computation. This yields a blind, physics-grounded, zero-shot solver that is robust to operator blindness and heteroscedastic noise, and supports variable burst length B at inference time.

Perception–Distortion tradeoff. We exploit the stochasticity of ZSBR to expose a simple test-time control over the perception–distortion tradeoff. Loosely speaking, averaging stochastic reconstructions suppresses sample-specific variations and can favor lower-distortion estimates, providing an empirical perception–distortion control. Concretely, we run ZSBR independently N times with different random seeds and average the resulting reconstructions:

$$\underbrace{\arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y})}_{\text{perception}} \longrightarrow \underbrace{\frac{1}{N}\sum_{i=1}^{N} \hat{\mathbf{x}}_i}_{\text{tradeoff}} \longrightarrow \underbrace{\mathbb{E}_p[\mathbf{x} \mid \mathbf{y}]}_{\text{distortion}} . \tag{9}$$

As N increases, averaging suppresses stochastic details, common in sharper and more perceptual outputs, and shifts the reconstruction toward a more distortion-oriented estimate, which typically improves PSNR.

4 Experiments

4.1 Experiments settings

We instantiate the diffusion prior using Stable Diffusion 2.1, and we use the `EulerAncestralDiscreteScheduler` sampler. All the experiments are conducted on A6000 Nvidia GPUs. Restoring a single burst with $B = 14$ takes 154 seconds and about 12.66GiB peak vRAM. Sequential N = 8 averaging costs about 20.5 min at a similar peak memory; full cost comparisons are in Appendix D.7.

Datasets. The *SyntheticBurst Dataset*[3] validation set is used in our experiments for burst super-resolution. It contains 300 scenes, each composed of 14 RAW burst frames and a corresponding ground truth image. All the burst frames are generated from sRGB images from the pipeline introduced in [4]. It is constructed by first converting sRGB images to linear sensor space using the inverse camera pipeline from [6]. Synthetic bursts are then generated by applying random translations and random rotations to simulate misalignment. Each burst frame is downsampled by a factor of 4 via bilinear interpolation and subsequently mosaicked with a Bayer pattern to approximate real RAW sensor data. Finally, the images are corrupted by adding noise following a Poisson-Gaussian model.

The *BurstSR* validation set[4] is employed in the experiment of domain shifting. It contains 882 burst–ground truth pairs, where low-quality RAW bursts are captured with mobile devices and corresponding high-quality images are obtained using DSLR cameras.

In terms of burst denoising, we evaluate our algorithm on the *color burst denoising dataset* [52], which includes 100 scenes, each composed of 8 linear burst frames and a corresponding ground truth image. In this setting, there is no downsampling and no mosaicking. Following [52], we report results only for the most noisy setting, where the sensor gain is the highest.

Evaluation Metrics. To provide a comprehensive evaluation of the restoration quality, we report both full-reference and no-reference metrics. PSNR and SSIM are traditional full-reference metrics for fidelity evaluation [48], while LPIPS [56] is a perceptual full-reference metric. The no-reference evaluation metrics include CLIPIQA [46], MUSIQ [28], and MANIQA [54]. Notably, no-reference metrics serve as complementary perceptual indicators alongside full-reference and measurement-consistency metrics because of the possible hallucination by the prior.

4.2 Comparison with SOTAs

Table 1 summarizes the quantitative results on the SyntheticBurst dataset. To the best of our knowledge, there are currently no established *zero-shot* methods specifically designed for burst restoration in the literature: existing high-performing approaches are supervised, i.e. , trained end-to-end on paired burst datasets, thereby inheriting the corresponding acquisition pipeline and noise statistics. Importantly, we do not claim that zero-shot methods should outperform in-distribution supervised SOTAs; rather, our aim is to provide a robust training-free alternative that remains applicable when paired data and sensor-specific calibration are unavailable or when the test acquisition deviates from the training distribution. Because our goal is to study burst restoration in the genuinely training-free setting, we therefore report the results for DPS [10]. We also introduce a simple zero-shot reference baseline (SimpleVAE) to provide a meaningful point of comparison within this regime.

Table 1: SyntheticBurst[3] results. Methods with † are blind. If not specified, assume $N = 1$.

	Method	Full-reference metrics			No-reference metrics↑		
		LPIPS	PSNR	SSIM	CLIP-IQA	MUSIQ	MANIQA
Supervised	Burstormer[16]	0.030	43.21	0.9709	0.432	37.766	0.334
	BIPNet[15]	0.032	42.61	0.9682	0.428	35.883	0.323
	RBSR[50]	0.041	42.31	0.9639	0.414	35.477	0.324
	BSRT[33]	0.028	43.54	0.9713	0.418	38.585	0.334
	DBSR[4]	0.072	40.05	0.9482	0.385	31.866	0.292
	BSRD+Random	0.067	33.86	0.8730	N.A	N.A	N.A
	BSRD+Burstormer	0.027	40.74	0.9524	0.383	38.727	0.322
	BSRD+BIPNet[43]	0.029	40.53	0.9509	0.376	37.897	0.319
Zero-shot	Bicubic	0.290	31.04	0.8350	0.244	17.924	0.173
	DPS[10]	0.110	34.17	0.8922	0.418	36.572	0.334
	SimpleVAE	0.156	34.97	0.8503	0.230	36.801	0.257
	Ours	0.067	39.34	0.9293	0.441	41.518	0.373
	Ours†	0.067	39.18	0.9247	0.442	41.751	0.376
	Ours $N = 8$	0.055	41.04	0.9496	0.412	38.619	0.349

Table 2: Denoising[52] results. Methods with † are blind. If not specified, assume $N = 1$.

	Method	Full-reference metrics			No-reference metrics↑		
		LPIPS	PSNR	SSIM	CLIP-IQA	MUSIQ	MANIQA
Supervised	BPN[52]	0.118	31.18	0.8595	0.432	50.933	0.386
	BIPNet[15]	0.063	32.87	0.8988	0.510	54.385	0.445
	EBDNet[36]	0.059	33.36	0.9070	0.518	54.103	0.450
	Burstormer[16]	0.060	33.55	0.9080	0.529	55.358	0.474
Zero-shot	Bicubic	0.172	24.89	0.6135	0.392	36.962	0.237
	DPS[10]	0.139	26.22	0.7732	0.458	44.294	0.332
	SimpleVAE	0.141	28.65	0.7769	0.390	48.011	0.362
	Ours	0.076	30.75	0.8420	0.539	55.364	0.477
	Ours†	0.081	29.95	0.8289	0.529	55.354	0.474
	Ours $N = 8$	0.079	32.20	0.8776	0.486	54.489	0.451



LQ Burstormer BIPNet BSRT BSRD+Bur BSRD+BIP Ours GT

Figure 3: Qualitative comparison on a SyntheticBurst example (ID 259).

This is a lightweight baseline that uses the Stable Diffusion 3.5 VAE [18], keeping the rest of the ZSBR pipeline the same, but instead of using the pretrained diffusion model as the source of natural image prior, it imposes a standard normal distribution prior on the latent space.

Moreover, when the noise parameters are treated as unknown and estimated during inference, performance degrades only mildly, as shown in the non-blind vs blind rows in Table 1, suggesting that the likelihood can recover reasonable noise statistics even in absence of explicit calibration.

For completeness, we also report representative supervised state-of-the-art burst SR methods, including DBSR [4], RBSR [50], BSRT [33], BIPNet [15], and Burstormer [16]. We further include recent diffusion-based burst pipelines such as BSRD [43] and its variants, which use (finetuned) diffusion as a refinement stage but still rely on a strong pretrained supervised reconstructor to produce an initial super-resolved estimate. For clarity, we group these approaches under supervised baselines, since their performance and generalization are bound to their training distributions.

Similarly, regarding the burst denoising task, Table 2 reports a comparison against representative supervised SOTAs, including BPN [52], BIPNet [15], Burstormer [16], and EBDNet [36]. As in burst super-resolution, we include the same SimpleVAE zero-shot baseline to establish a training-free reference point under our forward model and evaluation protocol.

Overall, the supervised methods achieve the strongest full-reference fidelity on both benchmarks, as expected given that they are trained on paired data with matched acquisition statistics. As shown in Table 1, the single-run variant ($N = 1$) of ZSBR already provide competitive restoration quality while achieving strong no-reference scores. When evaluated at a more distortion-oriented operating point via test-time averaging ($N = 8$), ZSBR further improves its full-reference metrics and narrows the gap to supervised methods. This behavior is consistent with our goal: ZSBR is designed as a robust training-free alternative rather than a replacement for supervised SOTAs under matched conditions. The denoising benchmark (Table 2) shows similar behaviors.

To evaluate ZSBR's generalization under domain shift, we perform a cross-dataset experiment where methods trained on the SyntheticBurst dataset [3] are directly tested on the real-world BurstSR dataset [4]. We compare ZSBR against the supervised baselines in Table 1. In this evaluation, ZSBR operates both in the blind and non-blind settings. We follow DBSR[4]'s BurstSR evaluation protocol during evaluation, which uses a separate 3×3 color transformation fitted to the ground-truth reference for each method and scene before computing full-reference metrics. More details on the protocol are in Appendix D.8 . Table 3 shows that ZSBR outperforms the competing methods in color-normalized

Table 3: Domain shift comparison. The evaluations are calculated after color alignment.

Method	aLPIPS	aPSNR (dB)	aSSIM
Ours†	**0.060**	45.95	0.9684
Ours† $N = 8$	0.063	**46.19**	**0.9695**
Ours	0.061	45.96	0.9686
BSRD+Burstormer	0.064	44.14	0.9565
BSRD+BIPNet	0.063	44.06	0.9535
DBSR (Syn.)	0.068	44.26	0.9604
RBSR (Syn.)	0.069	44.10	0.9590
BSRT-S (Syn.)	0.070	44.00	0.9582
Burstormer (Syn.)	0.080	43.31	0.9495
BIPNet (Syn.)	0.080	43.06	0.9444

Table 4: Ablation of ZSBR components.

Norm	Opt	Flow	LPIPS	PSNR	SSIM
ℓ^2	SGD	Frozen	0.105	36.42	0.9081
ℓ^2	Adam	Learnable	0.074	39.11	0.9231
NLL	SGD	Learnable	0.347	19.81	0.6762
NLL	Adam	Frozen	0.090	37.10	0.9092
NLL	Adam	Learnable	0.067	39.34	0.9293

Table 5: Initialization sensitivity; perturbations and schedules in Appendix D.2.

Initialization	PSNR↑	SSIM↑	LPIPS↓
PWCNet, frozen	37.10	0.9092	0.090
PWCNet, refined	39.34	0.9293	0.067
PWCNet + small	39.21	0.9290	0.062
PWCNet + medium	39.14	0.9273	0.065
PWCNet + large	36.60	0.8967	0.081
Identity (start 0%)	31.97	0.8281	0.086

structural restoration under cross-sensor domain shift. Fig. 4 qualitatively shows that competing methods exhibit noticeable artifacts while ZSBR produces sharper and more faithful reconstructions.

In addition, we evaluate 30 SIDD-Full scenes [1] from five smartphone cameras, using 14 RAW captures per scene. Table 6 indicates that calibrated and blind NLL ZSBR achieve 42.62 and 42.37 dB RAW PSNR, respectively, compared with 32.92 dB for the noisy reference. The static multi-frame setup and noise-metadata sources are specified in Appendix D.1.

To quantify the perception–distortion effect, for each burst we run ZSBR N times with different random seeds and average the reconstructed outputs before evaluation. We report PSNR and FID[24] on SyntheticBurst, with FID computed between the restored images and the ground-truth image distribution of the training split, i.e. unpaired. Fig. 5 shows that the PSNR monotonically improves as N increases, confirming that averaging suppresses seed-specific variance and leads to lower-distortion reconstructions. In contrast, FID shows a non-monotonic trend: it improves sharply from $N = 1$ to $N = 2, 3$, but gradually worsens as N becomes larger. We hypothesize that the initial gain comes from averaging out occasional outliers or failure cases, while excessive averaging suppresses high-frequency stochastic details and shifts the output distribution away from perceptually realistic samples.

Table 6: Real RAW denoising on 30 SIDD-Full scenes. PSNR is in dB. LPIPS is evaluated in sRGB.

	RAW		sRGB		
Condition	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
Noisy reference	32.92	0.7052	22.61	0.3586	0.453
ZSBR + calibrated NLL	**42.62**	**0.9625**	**33.88**	**0.8643**	**0.114**
ZSBR + blind NLL	42.37	0.9612	33.72	0.8624	**0.114**
ZSBR + ℓ^2	42.10	0.9579	33.16	0.8542	0.128

4.3 Ablation study

To validate the effectiveness of the key components in ZSBR, we perform an ablation study that isolates the roles of (i) the data-consistency term, (ii) the inner-loop optimizer, and (iii) online refinement of the registration parameters. Results are reported in Table 4. Starting from a standard homoscedastic ℓ^2 consistency term with Adam and affine finetuning, replacing ℓ^2 with our Poisson-Gaussian NLL consistently improves all full-reference metrics, supporting our claim that signal-dependent reweighting yields a better-calibrated guidance signal in RAW space. The choice of optimizer is critical: although SGD is a natural baseline, combining NLL guidance with joint optimization over the latent and affine parameters causes severe instability and failure to converge

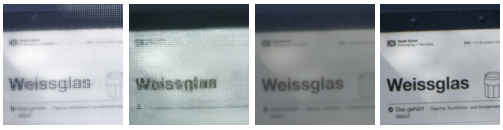


Figure 4: From top to bottom and left to right: BSRD+BIPNet, BSRD+Burstormer, Ours, and GT.

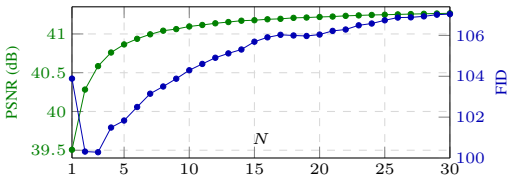


Figure 5: PSNR and FID against N.

, whereas Adam's adaptive scaling and momentum stabilize updates through the highly non-linear forward operator. Finally, freezing the alignment to the initial PWCNet-derived estimate leads to a marked degradation , indicating that even modest misalignment biases the likelihood gradients and forces the sampler to satisfy an incorrect consistency constraint; refining the affine transforms online mitigates this effect. Overall, the best configuration combines NLL guidance, Adam updates, and affine finetuning, confirming that ZSBR benefits from jointly addressing heteroscedastic noise and operator uncertainty during inference.

The influence of initialization quality is also evaluated. Table 5 investigates this sensitivity on all 300 SyntheticBurst Dataset scenes. Small and medium perturbations reduce PSNR by only 0.13 and 0.20 dB, respectively, whereas large perturbations and identity initialization lead to drops of 2.74 and 7.37 dB. Thus, ZSBR tolerates moderate initialization errors but relies on a reasonable starting estimate. Details of the experimental setting can be found in Appendix D.2.

Beyond the component ablations, we further investigate ZSBR's noise and affine matrix parameter recovery, sensitivity to modeling assumptions, optimization behavior, and failure modes. Online refinement reduces affine EPE by 33.9%, improving alignment in 49/50 scenes, while the recovered shot-noise coefficient achieves $R^2 = 0.7491$ (Appendix D.3). ISP perturbations reveal increasing sensitivity to calibration errors (Appendix D.4), while backbone and prompt experiments demonstrate applicability to a pixel-space prior (Appendix D.5) and limited sensitivity to prompt changes (Appendix B). The NLL reduction indicates an improvement in measurement consistency during sampling(Appendix D.6), while representative failure cases highlight limitations under weak textures and severe misalignment (Appendix D.10).

5 Conclusion and Limitations

We introduced ZSBR, a zero-shot diffusion framework for burst image restoration that performs MAP-style inference under a physically-grounded burst formation model. ZSBR explicitly models heteroscedastic RAW noise via a Poisson-Gaussian likelihood and jointly refines inter-frame alignment by optimizing per-frame affine transforms during sampling. To stabilize optimization through the resulting non-linear operator, we integrate an adaptive Adam-based descent scheme into the diffusion loop and optionally estimate unknown sensor noise parameters online. Meanwhile, we provide a test-time hyperparameter to navigate the perception-distortion tradeoff, enabling users to favor either perceptual sharpness or more distortion-oriented reconstructions.

Experiments on burst super-resolution and burst denoising demonstrate that ZSBR provides a flexible training-free alternative that remains robust when forward operators and noise statistics are uncertain. At the same time, zero-shot burst restoration remains a challenging setting: when large matched training sets are available, supervised methods still achieve stronger in-distribution performance. In addition, ZSBR is computationally more demanding at inference time due to iterative diffusion sampling and inner-loop optimization. We nevertheless believe it provides a useful and robust option for burst restoration under distribution shift and model uncertainty. ZSBR remains limited in handling local motion, parallax, occlusion, and rolling shutter, as it assumes predominantly static scenes with global-affine motion and a reasonable PWCNet initialization.

Funding and Competing Interests

The authors received no third-party funding or support for this work and declare no competing financial interests.

References

[1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *European conference on computer vision*, pages 571–582. Springer, 1996.

[3] G. Bhat, M. Danelljan, and R. Timofte. Ntire 2021 challenge on burst super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 613–626, June 2021.

[4] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9209–9218, 2021.

[5] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[6] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11036–11045, 2019.

[7] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[8] L. Chen, X. Chu, X. Zhang, and J. Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022.

[9] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

[10] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[11] H. Chung, S. Lee, and J. C. Ye. Decomposed diffusion sampler for accelerating large-scale inverse problems. *arXiv preprint arXiv:2303.05754*, 2023.

[12] A. N. Cotrim, G. Barbosa, C. A. N. Santos, and H. Pedrini. Simple base frame guided residual network for raw burst image super-resolution. *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 3:77 – 87, 2024. doi: 10.5220/0012348300003660. Cited by: 2; All Open Access, Gold Open Access.

[13] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[14] X. Di, L. Peng, P. Xia, W. Li, R. Pei, Y. Cao, Y. Wang, and Z.-J. Zha. Qmambabsr: Burst image super-resolution with query state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23080–23090, 2025.

[15] A. Dudhane, S. W. Zamir, S. Khan, F. S. Khan, and M.-H. Yang. Burst image restoration and enhancement. In *Proceedings of the ieee/cvf Conference on Computer Vision and Pattern Recognition*, pages 5759–5768, 2022.

[16] A. Dudhane, S. W. Zamir, S. Khan, F. S. Khan, and M.-H. Yang. Burstormer: Burst image restoration and enhancement transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5703–5712. IEEE, 2023.

[17] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6(12): 1646–1658, 1997. doi: 10.1109/83.650118.

[18] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[19] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE transactions on image processing*, 17(10): 1737–1754, 2008.

[20] T. Garber and T. Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2398–2407, 2025.

[21] C. Godard, K. Matzen, and M. Uyttendaele. Deep burst denoising. In *Proceedings of the European conference on computer vision (ECCV)*, pages 538–554, 2018.

[22] M. Guarnera, G. Messina, and V. Tomaselli. Adaptive color demosaicing and false color removal. *Journal of Electronic Imaging*, 19(2):021105–021105, 2010.

[23] S. W. Hasinoff. Photon, poisson noise. In *Computer vision: a reference guide*, pages 980–982. Springer, 2021.

[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] K. Kawai, T. Oba, K. Tokoro, K. Akita, and N. Ukita. Efficient burst super-resolution with one-step diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 864–873, 2025.

[27] B. Kawar, M. Elad, S. Ermon, and J. Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

[28] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[29] J. Lafenetre, N. L. Nguyen, G. Facciolo, and T. Eboli. Handheld burst super-resolution meets multi-exposure satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2056–2064, 2023.

[30] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[31] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[32] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6):1–9, 2014.

[33] Z. Luo, Y. Li, S. Cheng, L. Yu, Q. Wu, Z. Wen, H. Fan, J. Sun, and S. Liu. Bsrt: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 998–1008, 2022.

[34] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2502–2510, 2018.

[35] A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli. Deepsum: Deep neural network for super-resolution of unregistered multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3644–3656, 2019.

[36] S. Pan and Y. Li. Ebdnet: Integrating optical flow with kernel prediction for burst denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[37] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern recognition letters*, 5(3):223–226, 1987.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[39] X. Rong, D. Demandolx, K. Matzen, P. Chatterjee, and Y. Tian. Burst denoising via temporally shifted wavelet transforms. In *European Conference on Computer Vision*, pages 240–256. Springer, 2020.

[40] F. Rozet, G. Andry, F. Lanusse, and G. Louppe. Learning diffusion priors from observations by expectation maximization. *Advances in Neural Information Processing Systems*, 37:87647–87682, 2024.

[41] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America A*, 6(11):1715–1726, 1989.

[42] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[43] K. Tokoro, K. Akita, and N. Ukita. Burst super-resolution with diffusion models for improving perceptual quality. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[44] R. Y. Tsai and T. S. Huang. Multiframe image restoration and registration. *Multiframe image restoration and registration*, 1:317–339, 1984.

[45] O. Unal, S. Marty, and D. Dai. Burst image super-resolution with mamba. *arXiv preprint arXiv:2503.19634*, 2025.

[46] J. Wang, K. C. Chan, and C. C. Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023.

[47] Y. Wang, J. Yu, and J. Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[49] P. Wei, Y. Sun, X. Guo, C. Liu, G. Li, J. Chen, X. Ji, and L. Lin. Towards real-world burst image super-resolution: Benchmark and method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13233–13242, 2023.

[50] R. Wu, Z. Zhang, S. Zhang, H. Zhang, and W. Zuo. Rbsr: Efficient and flexible recurrent network for burst super-resolution. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 65–78. Springer, 2023.

[51] R. Wu, L. Sun, Z. Ma, and L. Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37:92529–92553, 2024.

[52] Z. Xia, F. Perazzi, M. Gharbi, K. Sunkavalli, and A. Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11844–11853, 2020.

[53] T. Xu, X. Cai, X. Zhang, X. Ge, D. He, M. Sun, J. Liu, Y.-Q. Zhang, J. Li, and Y. Wang. Rethinking diffusion posterior sampling: From conditional score estimator to maximizing a posterior. *arXiv preprint arXiv:2501.18913*, 2025.

[54] S. Yang, T. Wu, S. Shi, S. Lao, Y. Gong, M. Cao, J. Wang, and Y. Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1191–1200, 2022.

[55] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2696–2705, 2020.

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

APPENDIX

A Estimate affine transformations algorithm

Algorithm 2 computes the initial estimate of the affine transformations between each burst frame and the reference one. These alignments are then used in the proposed forward operator, as described in Section 3.1.

Algorithm 2 Estimate affine transformations

Input: RAW burst frames $\{\mathbf{y}^b\}_{b=1}^{B}$

Output: Affine transformation matrices $\{w^b\}_{b=1}^{B}$

$\tilde{\mathbf{y}}^1 = \text{RawToGrayscale}(\mathbf{y}^1)$
$w^1 \leftarrow \mathbb{I}$
for $b = 2$ **to** B **do**
    $\tilde{\mathbf{y}}^b \leftarrow \text{RawToGrayscale}(\mathbf{y}^b)$
    $\mathbf{f}^b \leftarrow \text{PWCNet}(\tilde{\mathbf{y}}^1, \tilde{\mathbf{y}}^b)$
    $w^b \leftarrow \text{FlowToAffineOLS}(\mathbf{f}^b)$
return $\{w^b\}_{b=1}^{B}$

B Stable Diffusion Prompts and Sensitivity

The **positive prompt** used to condition the stable diffusion backbone in all the ZSBR experiments is "realistic photo, natural contrast, natural colors, clean image".

The **negative prompt** is "overtextured, exaggerated details, oversharpened, halos, ringing, moire, aliasing, crunchy, HDR artifacts, plastic, waxy, smeared, watercolor, painterly, illustration, cartoon, CGI, 3d render, text, watermark, logo".

Prompt swap. We stress test the chosen prompt's sensitivity by inverting the negative prompts and positive prompts, the result is shown in Table 7

Swapping the prompts changes PSNR by only 0.2 dB, SSIM by 0.0017, and LPIPS by 0.003, The changes are small and mixed across distortion and perceptual metrics: distortion slightly decreases, whereas LPIPS slightly improves. The relevant observation is that even this deliberately adverse prompt change does not materially affect reconstruction quality. This suggests that the RAW likelihood and multi-frame data consistency dominate the text conditioning in ZSBR.

Table 7: Prompt sensitivity for Stable Diffusion 2.1. The swapped condition exchanges the positive and negative prompts.

Prompt setting	PSNR (dB)↑	SSIM↑	LPIPS↓
Default prompts	39.34	0.9293	0.067
Swapped prompts	39.14	0.9276	0.064

C Visual Result

Fig. 6 shows examples for the SyntheticBurst dataset. Similarly, Fig. 7 shows examples for the Color Denoising dataset. We can see that, while DPS shows artifacts and inaccuracies, our zero-shot method, ZSBR, is comparable with the supervised SOTAs.

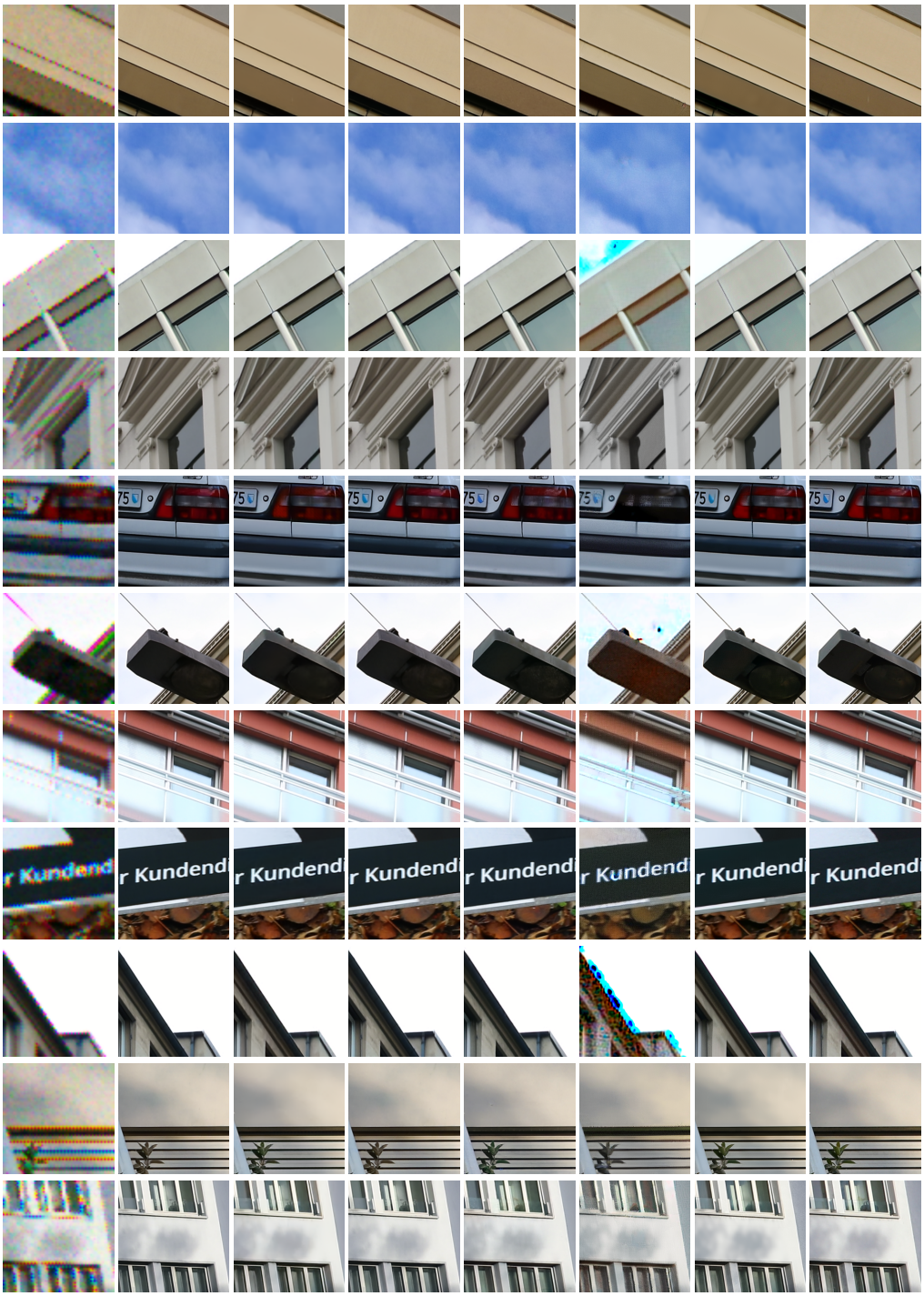


Figure 6: Qualitative comparison on SyntheticBurst dataset.

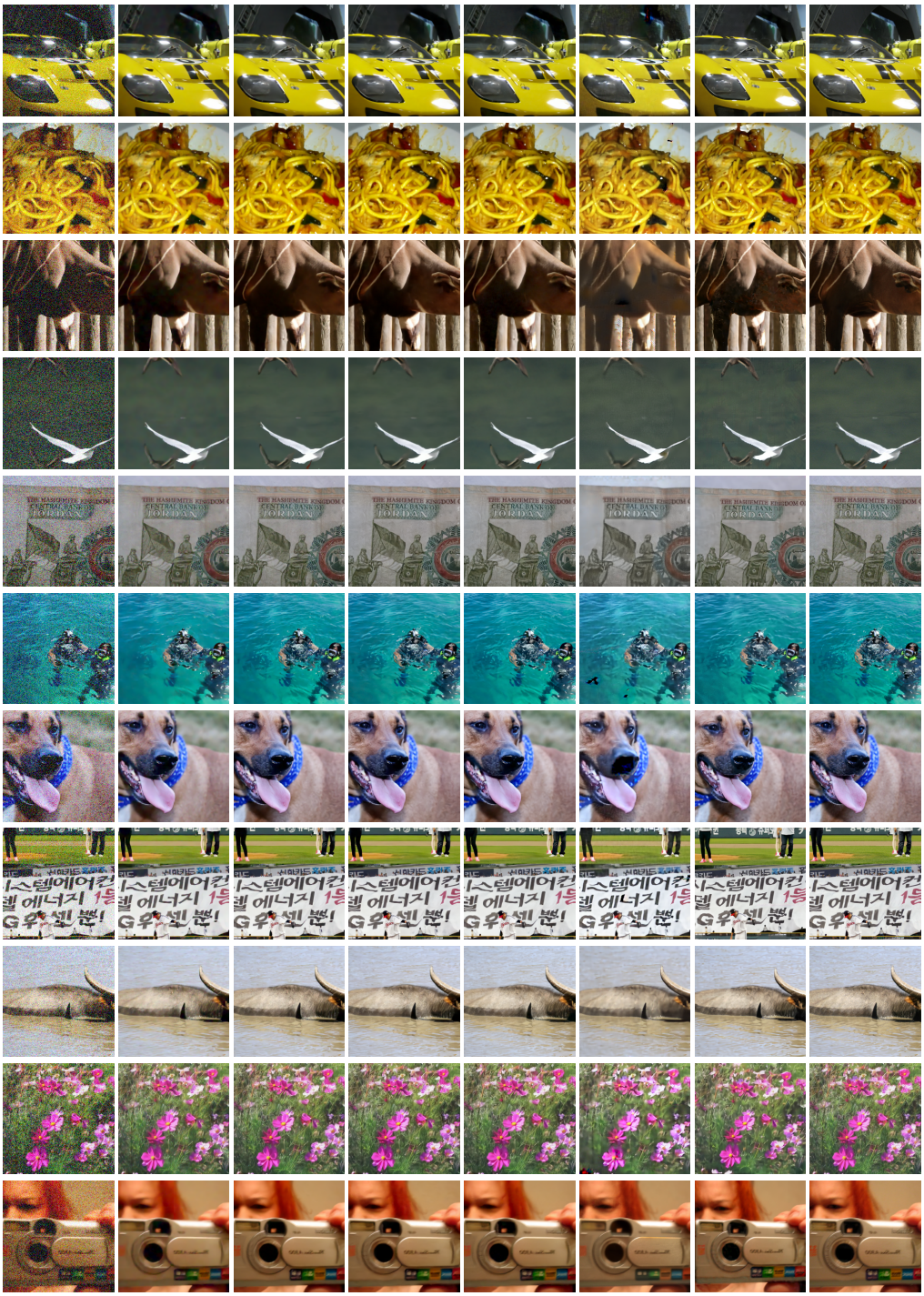


Figure 7: Qualitative comparison on Color Denoising dataset.

D Additional Experiments and Implementation Details

D.1 Real RAW Denoising Protocol

We selected 30 Full-scenes in SIDD dataset. Each scene contains 14 frames, and The first frame is used as the reference, and the other 13 are additional input frames. Calibrated NLL uses SIDD noise metadata, whereas blind NLL estimates shot/read coefficients during inference. RAW domain PSNR/SSIM and sRGB PSNR/SSIM/LPIPS are reported in Table 6. Blind NLL remains within 0.25 dB RAW and 0.16 dB sRGB PSNR of calibrated NLL. This supports the effectiveness of the Poisson-Gaussian likelihood under real sensor noise and multiple camera pipelines.

D.2 Alignment Initialization

We evaluate initialization sensitivity on all 300 SyntheticBurst dataset scenes. The default affine transforms are initialized using frozen PWCNet and refined during inference. We compare frozen and refined transforms, apply controlled rigid perturbations after PWCNet initialization, and test identity initialization. Small, medium, and large perturbations use rotation/translation magnitudes of $\pm 0.25^\circ$/0.25 LR pixel, $\pm 0.75^\circ$/0.75 LR pixel, and $\pm 5^\circ$/5 LR pixels, respectively. The perturbed configurations use the standard refinement schedule, starting at 25% of the diffusion trajectory; the identity configuration enables refinement from the beginning.

D.3 Recovery Capability

Affine Recovery.

To further prove that the motion estimate is actually optimizing the affine matrix instead of solely improving performance, we evaluate alignment on a separate controlled 50-scene synthetic benchmark with known transformations. Each affine transform is converted to a dense displacement field on the low-resolution image grid. Endpoint error (EPE) is measured against the ground-truth field and averaged over evaluated pixels, non-reference frames, and scenes.

Table 8: Alignment accuracy on the controlled 50-scene benchmark. EPE is in LR pixels. This benchmark is separate from the 300-scene initialization study.

Condition	Initial EPE↓	Final EPE↓
Ground-truth affine, fixed	0.000014	0.000014
PWC affine, fixed	0.831364	0.831364
PWC affine, optimized	0.831363	0.549844

Online refinement reduces mean EPE by 33.9%, from approximately 0.8314 to 0.5498 LR pixels. Alignment improves in 49 of 50 scenes (98%). The direct geometric evaluation supports improvement of the estimated forward model, in addition to the reconstruction gains. These results demonstrate that the likelihood-guided updates improve the estimated forward model itself, rather than merely improving the final reconstruction through the image prior.

Noise-Parameter Recovery.

On synthetic data with known noise parameters, the recovered shot-noise coefficient agrees with ground truth with $R^2 = 0.7491$. The reported read-noise analysis also shows recovery when the true values are above the very small-noise regime. Very small read-noise estimates are limited by the numerical floor reported in this evaluation, approximately 10^{-6}.

D.4 Inverse-ISP Sensitivity

We evaluate inverse-ISP robustness by perturbing two metadata-based components: white-balance gains and the color-correction matrix (CCM). Gains are perturbed multiplicatively in log space, while CCM perturbations preserve each row sum. We use nested $1\times$, $2\times$, and $5\times$ perturbations. Table 9 reports the median gain and CCM errors together with reconstruction metrics.

PSNR decreases progressively as these errors grow, by 0.64, 1.58, and 4.40 dB relative to the unperturbed setting. SSIM also decreases, whereas LPIPS is not monotonic across the tested

Table 9: Sensitivity to white-balance and CCM errors. Gain and CCM errors are the median percentage errors reported in this evaluation.

ISP condition	Gain error	CCM error	PSNR (dB)↑	SSIM↑	LPIPS↓
Unperturbed	0%	0%	39.34	0.9293	0.067
1×	1.56%	1.64%	38.70	0.9280	0.063
2×	3.12%	3.28%	37.76	0.9273	0.064
5×	7.87%	8.19%	34.94	0.9232	0.069

conditions. These results show that the quality of the ZSBR solutions degrades smoothly with bigger and bigger ISP perturbations.

D.5 Diffusion Prior Sensitivity

We also apply the inference procedure to the ImageNet-pretrained pixel-space Guided Diffusion model [13], in place of Stable Diffusion 2.1. Table 10 reports the resulting reconstruction quality. The alternative prior yields a PSNR reduction of 0.82 dB, together with lower SSIM and higher LPIPS.

Table 10: Sensitivity to the diffusion backbone. Guided Diffusion is the ImageNet-pretrained pixel-space model reported in this evaluation.

Diffusion prior	PSNR (dB)↑	SSIM↑	LPIPS↓
Stable Diffusion 2.1	39.34	0.9293	0.0670
Guided Diffusion	38.52	0.8803	0.0828

The fact that the procedure remains effective after replacing SD2.1 with a substantially different pixel-space diffusion prior indicates that it is not intrinsically tied to the SD2.1 latent representation. At the same time, the performance gap confirms that the choice and quality of the pretrained prior have a measurable impact.

D.6 Measurement-NLL Progression

We record the measurement negative log-likelihood (NLL) throughout the standard 50-step diffusion trajectory. The same optimization settings are used for every scene, without selecting runs according to their final performance. The values in Table 11 are burst-level summed NLLs averaged over the evaluated scenes, using the step indices reported here.

Table 11: Mean measurement NLL at the reported sampling checkpoints.

Sampling progress	Diffusion step	Mean NLL↓
20%	9	758,861.08
40%	19	110,363.73
60%	29	−176,876.46
80%	39	−294,972.14
100%	49	−334,385.69

The mean NLL decreases at each of these reported checkpoints, supporting improved measurement consistency along the observed trajectories.

D.7 Matched Runtime, Computation, and Memory

We profile the methods under matched hardware, input, and precision settings. Table 12 reports median runtime, peak GPU memory over the complete restoration procedure, and the total computation reported here. ZSBR takes 154.06 s per restoration, compared with 14.98 s for DPS, approximately 10 s for the two BSRD pipelines, and substantially less than one second for the listed supervised feed-forward methods.

Compared with DPS, ZSBR requires approximately $10.3\times$ longer runtime and $10.0\times$ more computation because each diffusion step includes multiple Adam-guided NLL updates. Peak VRAM increases only marginally, from 12.51 to 12.66 GiB, since the inner updates are executed sequentially.

Table 12: Cost comparison for the evaluated methods. Single-run methods use matched profiling: runtime is the median, memory is peak VRAM over the complete procedure, and computation is in TFLOPs. For sequential $N = 8$, runtime and computation are estimated as $8\times$ the $N = 1$ costs; peak VRAM is approximately unchanged.

Method	Runtime (s)	VRAM (GiB)	Compute (TFLOPs)
DBSR	0.019	0.48	0.22
Burstormer	0.024	0.34	0.12
BIPNet	0.039	0.36	0.52
BSRT-S	0.080	0.18	0.61
RBSR	0.096	0.19	0.31
BSRD + Burstormer	10.03	3.07	231.59
BSRD + BIPNet	10.07	3.08	231.99
DPS	14.98	12.51	175.96
ZSBR ($N = 1$)	154.06	12.66	1750.36
ZSBR ($N = 8$, sequential)	≈ 1232.5	$\approx$12.66	≈ 14002.9

For sequential N-run averaging, runtime and computation scale approximately linearly with N, while peak VRAM stays near the single-run requirement. The current procedure is intended for offline restoration and is not suitable for real-time deployment.

D.8 Color Alignment

The low-resolution bursts and high-resolution references are captured with different cameras. For each scene and each method output, evaluation fits a separate global 3×3 linear color transform to the ground-truth reference. Let $\hat{x}(u)$ and $x^\star(u)$ denote reconstructed and ground-truth linear-RGB values at a pixel u in the evaluated set U. The transform and aligned PSNR are

$$C^\star = \underset{C\in\mathbb{R}^{3\times 3}}{\operatorname{argmin}} \frac{1}{3|U|} \sum_{u\in U} \|C\hat{x}(u) - x^\star(u)\|_2^2 , \tag{10}$$

$$\mathrm{aPSNR}(\hat{x}, x^\star) = -10 \log_{10} \left[\frac{1}{3|U|} \sum_{u\in U} \|C^\star\hat{x}(u) - x^\star(u)\|_2^2 \right] , \tag{11}$$

with a peak value of one. The transform is obtained by linear least squares. Ground truth is used to fit the evaluation transform and compute metrics; it is not used by ZSBR during reconstruction or optimization.

The same fitting procedure is applied to every compared method, table 3 provides a color-normalized structural evaluation and does not directly establish robustness to an unknown camera color pipeline or ISP.

D.9 Optimizer Details

The reported inference schedule uses 50 diffusion steps with 10 inner updates per diffusion step. Latent variables are updated at every diffusion step. Affine refinement and blind shot/read noise estimation in blind setting begin at 25% of sampling.

The optimizer hyperparameters are detailed in Table 13.

Table 13: Optimizer settings reported in this evaluation. The latent update is a manual Adam-style update; affine and blind-noise parameters use PyTorch Adam.

Variable / optimizer	Learning rate	Update schedule
Diffusion latent / manual Adam-style	0.048	All 50 diffusion steps; 10 inner updates per step; moments reset at each diffusion step.
Affine parameters / PyTorch Adam	10^{-3}	Begins at 25% of sampling; 37 diffusion steps $\times$ 10 inner updates.
Blind shot/read parameters / Adam parameter group	0.05	Same 370-update schedule, only when blind-noise estimation is enabled.

D.10 Failure Cases and Limitations

In textureless regions, such as smooth skies or uniform walls, unreliable correspondences can produce poor PWCNet initialization. When the likelihood provides weak constraints, the prior may introduce plausible but incorrect textures. Severe misalignment can also cause ghosting and locally inconsistent details. Online refinement corrects errors within the global-affine model; it does not establish recovery of dynamic objects, parallax, occlusions, or rolling-shutter distortions under non-ideal burst conditions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims in the abstract match the method explanation and the experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion and Limitation section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: No theoretical results

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details and hyperparameter are reported in the Experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is availble at: https://github.com/IPLLab-PoliTO/ZSBR

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting and details are provided in the Experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the standard presentation of the literature and present only one result. This could be considered the outcome of an average run, as no selection over multiple runs has been performed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The resources is reported in experiment setting subsection of the Experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed. ZSBR belongs to image quality improvement algorithms, which do not have much societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the assets are cited properly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new asset release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects, crowdsourcing, or user studies. All experiments are conducted on existing benchmark datasets, and no data collection involving human participants was performed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.