PickMoment: Continuous-Time Single-Image-to-Video via Learning Deblurring and Blur-to-Video

Junseong Shin[1] **Hyeonsu Jo**[2] **Daehyun Kim**[3] **Tae Hyun Kim**[1,3†]

[1]Department of Artificial Intelligence, Hanyang University
[2]Department of Intelligence Convergence, Hanyang University
[3]Department of Computer Science, Hanyang University
{junsung6140, gustn249, daehyun, taehyunkim}@hanyang.ac.kr
[†]Corresponding author

Abstract

Motion blur arises from the uniform temporal integration of a continuous sharp signal over a finite exposure window, yet existing learning-based methods sidestep this physical model and predict only the sharp signal itself: most single-image deblurring methods recover a single frame at the exposure center, while blur-to-video methods predict a fixed set of frames. We introduce PickMoment, a continuous-time reformulation that directly learns the interval-mean blur over arbitrary sub-intervals of the exposure with a single deterministic model. Drawing an analogy to MeanFlow's average-velocity formulation, we train the model with three supervisions derived from the blur integral: an empirical reconstruction loss from available subframes, an additivity loss that enforces self-consistency across overlapping sub-intervals, and a sharp-frame loss anchored at the zero-interval limit. A single trained model unifies single-image deblurring, blur-to-video generation, and continuous-time pick-a-moment recovery as different queries to the same network, with no separate training for each task. Our PickMoment achieves state-of-the-art performance among generative-prior-based deblurring methods on GoPro and HIDE while competitive against restoration-based methods on RealBlur, and the highest per-frame fidelity on GoPro-7 blur-to-video, all with one forward pass per output frame and no iterative sampling. Our project page is available here.

1 Introduction

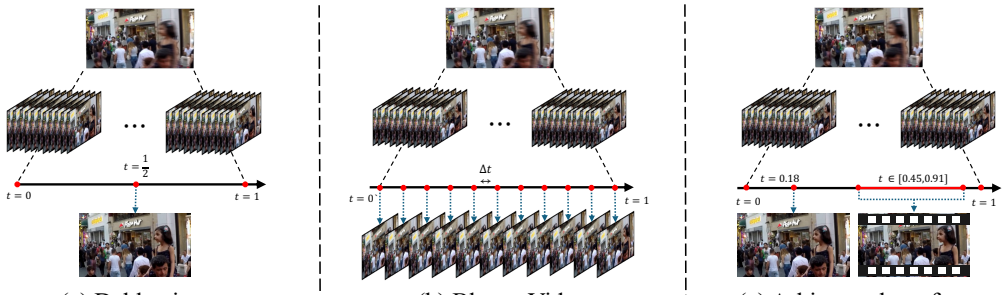
Motion blur is a pervasive degradation in real-world imaging, arising from camera shake or object motion during the exposure process. Beyond deteriorating visual quality, it hinders human perception and limits the applicability of images in downstream tasks such as object detection [36], 3D reconstruction [28], and depth estimation [14]. To address this, prior works have not only advanced learning-based deblurring approaches [30] but also sought to understand the underlying principles of blur formation [9].

Motion blur arises from the temporal integration of sharp frames over a finite exposure window. Following Nah et al. [30], given an exposure interval $[0, T]$ of duration T, we model the blurred image $B(0, T)$ as

$$B(0,T) \;=\; \frac{1}{T}\int_0^T S(u)\,\mathrm{d}u \simeq \frac{1}{M}\sum_{i=0}^{M-1} S_i, \tag{1}$$

where $S(u)$ denotes the sharp frame at instant $u \in [0, T]$ and M is the number of uniformly sampled frames used in discretization. This physically grounded interpretation has been widely adopted not only for realistic blur synthesis, but also as a key principle for learning-based deblurring.

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



(a) Deblurring (b) Blur to Video (c) Arbitrary sharp frames

Figure 1: Overview of the proposed **PickMoment** framework. Given a single motion-blurred image, our method unifies three tasks: **(a)** Image deblurring, predicting the center sharp frame of the blur exposure. **(b)** Blur-to-video generation predicts an n-frame sharp video by uniformly sampling the normalized exposure trajectory at M time points with interval $\Delta t = 1/(M-1)$; the figure illustrates an 11-frame example. **(c)** Arbitrary sharp-frame prediction enables the recovery of sharp frames at arbitrary time points within the blur exposure, thereby enabling continuous-time video generation.

Building on this view, single-image deblurring methods have progressed from CNNs [30, 42] and transformers [51, 19] to diffusion-based approaches [5, 23], achieving strong per-pixel and perceptual fidelity. However, they still recover only a single sharp frame from the sharp sequence $\{S_0, S_1, ..., S_{M-1}\}$ (typically the center frame of the exposure), collapsing the temporal integration into a single moment and discarding the rich temporal structure encoded in the blur.

To exploit this structure, blur-to-video methods aim to recover the entire sharp sequence from a single blurry image. Early work [16] trains with order-invariant losses that accept both forward and reversed sequences as valid targets. Subsequent methods introduce learned motion latents [33], differentiable warping with auxiliary regularizers [2], latent-space ordering classifiers [32], or, most recently, sampling from a pretrained video diffusion prior [43]. Across both single-image deblurring and blur-to-video tasks, however, the dominant learning target has remained the sharp signal itself, either at one timestep or at a fixed set of discrete timesteps. This fails to fully exploit the temporal ordering encoded in the physical formulation of Eq. 1, where the model is used only for synthesizing training data rather than shaping the network predictions themselves. In contrast, rather than learning a direct mapping from blur to a sharp frame or a sequence of video frames, our goal is to directly predict the blur formation integral itself from a single blurred image, allowing the physical model to serve as a core component of the learning objective.

A closely related idea has been explored in generative modeling. MeanFlow [10] directly predicts an average velocity field supervised by identities derived from its integral definition over arbitrary intervals. Since motion blur, as defined in Eq. 1, is itself a continuous-time integral of sharp frames over the exposure interval, it shares the same algebraic structure as MeanFlow's average velocity field, such that integral-derived identities serve as natural self-consistency constraints across all sub-intervals.

In this work, we propose **PickMoment**, which trains a network to predict blur observations over arbitrary sub-intervals of the exposure duration, enabling the learning of continuous temporal representations that generalize to any temporal range from a single blurry input. Specifically, we leverage two structural properties of motion blur: the blurry input itself corresponds to the blur integrated over the full exposure interval, and standard datasets [30, 37] provide subframes with known temporal ordering. Together, these properties, combined with our MeanFlow-inspired training strategy, enable the network to produce deterministic and direction-consistent reconstructions with one forward pass per output frame. As illustrated in Fig. 1, single-image deblurring, blur-to-video generation, and arbitrary-moment recovery naturally emerge as different queries to the same model. Furthermore, sharp frames at arbitrary continuous timesteps arise as the zero-interval limit, enabling continuous-time **"pick-a-moment"** recovery from a single blurred image.

In summary, our contributions are as follows: We introduce **PickMoment**, a physically grounded reformulation of motion deblurring that models a blurry image as a continuous-time integral over the exposure interval and casts deblurring as the prediction of interval-mean sharp content over arbitrary sub-intervals. By learning a continuous temporal representation, PickMoment unifies single-image deblurring, blur-to-video generation, and continuous-time **"pick-a-moment"** recovery

within a single formulation, without requiring separate models or objectives for different tasks. To effectively learn this formulation, we draw a structural analogy between sub-interval blur prediction and the average velocity formulation of MeanFlow. Specifically, we exploit two integral-derived identities — additivity across sub-intervals and the zero-interval limit — as supervision signals for temporally coherent learning. Extensive experiments demonstrate that PickMoment achieves state-of-the-art performance among generative-prior-based deblurring methods on GoPro and HIDE while remaining competitive with restoration-based baselines, and enables 1-step continuous-time blur-to-video generation with significantly reduced computational cost compared to video diffusion approaches.

2 Related Work

Single-image motion deblurring. Single-image motion deblurring aims to recover a sharp image from a single blurry observation, evolving from traditional handcrafted priors [9, 31] to deep learning. End-to-end CNN architectures [30, 42, 52, 20, 50, 4, 8, 29] pioneered direct blur-to-sharp mapping, followed by transformer-based models [51, 45, 19] that achieve state-of-the-art per-pixel fidelity through global self-attention. More recently, diffusion-based approaches have emerged to push perceptual quality. HI-Diff [5] integrates hierarchical latent priors into a diffusion framework for deblurring, and BlurDM [12] embeds the blur formation process into a dual-diffusion scheme (jointly diffusing noise and blur) and serves as a prior for downstream deblurring networks. A parallel line adapts large pretrained generative models for image restoration. DiffBIR [21] restores blind degradations using the Stable Diffusion [35] prior, and Diff-Plugin [25] provides plug-ins for diffusion-based low-level tasks. Most recently, several methods adapt the same prior specifically for deblurring. UID-Diff [6] addresses unpaired deblurring with a decoupled diffusion model on top of a refined VAE [18], and the concurrent FideDiff [23] adapts Stable Diffusion through consistency training for one-step inference. However, all these methods predict only a single sharp frame at the temporal center of the exposure, leaving users unable to pick any other moment encoded in the blur.

Blur-to-video. Beyond single-frame deblurring, blur-to-video recovers an entire sharp sequence from a single blurry image, inverting the temporal integration of the exposure. [16] first formulated this task with an ordering-invariant loss that accepts both forward and reversed sequences as valid targets, declaring the directional ambiguity unresolvable. Subsequent methods reduce the ambiguity via handcrafted motion priors [33, 2] with limited out-of-distribution robustness. MotionETR [54] estimates per-pixel motion offsets and warps a predicted sharp image to produce the sequence. Animation-from-Blur [55] models motion ambiguity as multi-modal and conditions decomposition on a quantized motion-direction guidance (four discrete optical-flow directions), generating multiple plausible decompositions. HyperCUT [32] learns a hyperplane in latent space that separates forward from reversed sequences, using the vector side to assign temporal order. Most recently, Blur2Vid [43] fine-tunes a video diffusion transformer and treats directional ambiguity by sampling, producing plausible draws rather than a single committed reconstruction. Despite these advances, all methods predict sharp content directly, rather than modeling the integral structure that defines motion blur.

Average-velocity Flow Matching. Flow Matching [22, 24, 1] is a generative modeling framework that learns to transform a simple prior distribution into a complex data distribution. This is achieved by training a network to predict the instantaneous velocity field along a continuous transport path between the two distributions. MeanFlow [10] instead trains a network on the *average velocity* over a time interval, which encodes the trajectory of the entire interval in a single prediction and lets one model handle arbitrary-interval queries. Subsequently, SplitMeanFlow [11] introduces an algebraic additivity identity as a training objective, and pixel MeanFlow (pMF) [27] predicts an image-like quantity for one-step generation without a latent space. Unlike these generative models, we adopt only the interval-averaging structure, predicting the temporal average of sharp frames over a sub-interval of the exposure in the latent space of a frozen VAE.

3 Method

3.1 Preliminary: MeanFlow

MeanFlow [10] is a generative modeling framework that introduces the notion of *average velocity* to characterize flow fields, in contrast to the instantaneous velocity modeled by Flow Matching. Given a flow trajectory with state z_t at time t, the average velocity over $[s, t]$ is the time-average of the

instantaneous velocity v:

$$u(z_t, s, t) \triangleq \frac{1}{t-s} \int_s^t v(z_\tau, \tau)\, \mathrm{d}\tau. \tag{2}$$

Multiplying Eq. 2 by $(t-s)$ and differentiating with respect to t (with fixed s), we obtain the following relation via the fundamental theorem of calculus, referred to as the *MeanFlow Identity*:

$$u(z_t, s, t) = v(z_t, t) - (t-s)\, \tfrac{d}{dt} u(z_t, s, t). \tag{3}$$

Since the true average u is unobserved, MeanFlow trains a network u_θ by regressing it to a target derived from the MeanFlow Identity (a *consistent estimator* in the sense that it equals u whenever u_θ does):

$$\mathcal{L}_{\text{MeanFlow}} = \mathbb{E}\Big\| u_\theta(z_t, s, t) - \mathrm{sg}\big(\hat{u}_{\text{id}}\big) \Big\|_2^2, \quad \hat{u}_{\text{id}} = v(z_t, t) - (t-s)\, \tfrac{d}{dt} u_\theta(z_t, s, t), \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient and the instantaneous prediction emerges as the zero-interval limit: $\lim_{s \to t} u(z_t, s, t) = v(z_t, t)$.

While effective, computing the target $\hat{u}_{\text{id}}$ in Eq. 4 requires a Jacobian-vector product (JVP) through the network to evaluate $\frac{d}{dt} u_\theta$, which increases per-step compute and memory. SplitMeanFlow [11] addresses this by deriving an *algebraic* counterpart of the MeanFlow Identity: splitting the integration domain at any $m \in [s, t]$ in Eq. 2 yields,

$$u(z_t, s, t) = \omega\, u(z_m, s, m) + (1{-}\omega)\, u(z_t, m, t), \quad \omega = \tfrac{m-s}{t-s}, \tag{5}$$

which follows directly from $\int_s^t = \int_s^m + \int_m^t$ and recovers the differential MeanFlow Identity in the limit $m \to t$. SplitMeanFlow uses Eq. 5 as a JVP-free training target, reducing per-step compute and memory.

3.2 Continuous-Time Blur Reformulation

Conventional deblurring methods predict only a single sharp frame at the temporal center of the exposure, collapsing the exposure to one moment. In contrast, to enable queries over arbitrary sub-intervals of the exposure, we first generalize the blur formation in Eq. 1 from the full exposure interval $[0, T]$ to an arbitrary sub-interval $[s, t]$. Formally, for any $0 \le s < t \le T$, the interval-mean blur is expressed as:

$$B(s, t) \triangleq \frac{1}{t-s} \int_s^t S(\tau)\, \mathrm{d}\tau, \tag{6}$$

where $S(\tau)$ is the sharp frame at time τ and $B(s, t)$ is the average sharp content over $[s, t]$. Throughout, we assume uniform temporal integration over a known exposure window, as in standard blur synthesis [30]; Appendix I discusses other regimes. This approach unifies the time-continuous representation of average sharp content $B(s, t)$, thereby generalizing conventional methods that recover a mid-exposure sharp image from a blurred observation $B(0, T)$, by interpreting the sharp frame $S(T/2)$ as the zero-interval limit $\lim_{t \to T/2} B(T/2, t)$.

Analogy with MeanFlow. We focus on the formal analogy between the interval-mean formulation of blur in Eq. 6 and the average velocity in MeanFlow (Eq. 2): both are time-averages of an underlying quantity over a temporal interval, with the correspondence $B \leftrightarrow u$ and $S \leftrightarrow v$. The analogy holds at the level of the averaging operator rather than the integrand: MeanFlow averages a velocity, whereas $B(s, t)$ averages sharp images, so the identities below carry over while MeanFlow's stochastic noise-to-data sampling does not. Applying the same derivation as the *MeanFlow Identity*, we obtain the analogous *blur differential identity*:

$$B(s, t) = S(t) - (t-s)\, \tfrac{d}{dt} B(s, t). \tag{7}$$

Likewise, the splitting argument behind Eq. 5 carries over to the blur integral, yielding the *blur additivity identity*: for any $m \in [s, t]$,

$$B(s, t) = \omega\, B(s, m) + (1-\omega)\, B(m, t), \quad \omega = \tfrac{m-s}{t-s}, \tag{8}$$

which admits a clean physical reading: a blur over an interval is the convex combination of two adjacent sub-exposure blurs. The blur differential identity also implies a useful boundary case: as the interval shrinks to zero, the blur converges to the sharp frame at that instant:

$$\lim_{t \to \tau} B(\tau, t) = S(\tau), \tag{9}$$

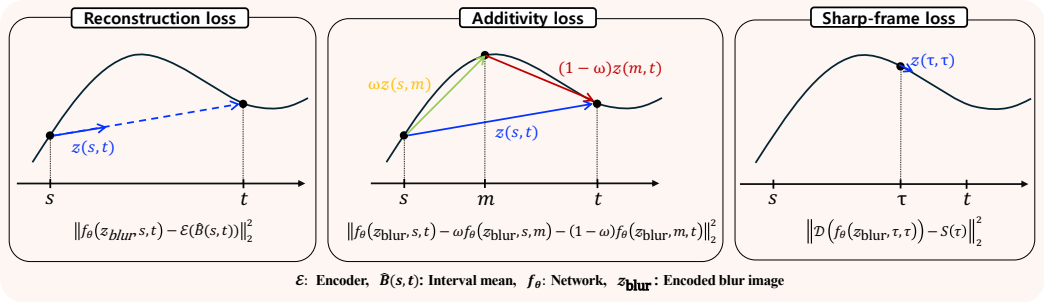


Figure 2: Conceptual illustration of the training strategy of PickMoment.

making single-image deblurring at an arbitrary continuous time τ the zero-interval limit of the same formulation, and enabling continuous-time **"pick-a-moment"** recovery from a single blurred image, as well as continuous-time sharp video generation.

Architecture and conditioning. We instantiate f_θ as a diffusion transformer built on the FLUX.2-klein backbone [3], which consists of a frozen VAE with encoder $\mathcal{E}$ and decoder $\mathcal{D}$, along with a transformer f_θ. The input blurry image x_{blur} is encoded into the latent space as $z_{\text{blur}} = \mathcal{E}(x_{\text{blur}})$, and f_θ is parameterized as a residual prediction: the transformer takes (z_{blur}, s, t) as input and outputs a latent displacement that is added to z_{blur} to produce $f_\theta(z_{\text{blur}}, s, t)$. To inject interval information, we replace FLUX's original time conditioning with an interval conditioner: the triple $(s, t, t-s)$ is sinusoidally embedded [47] and projected by a shared MLP, then injected through the same conditioning pathway as the original timestep.

3.3 Training Strategy

We train f_θ by regressing it toward target estimators of the true interval mean $B(s,t)$, following the training principle of MeanFlow in Eq. 4. As illustrated in Fig. 2, we employ three complementary supervision signals, each derived from the integral formulation in Eq. 6: an empirical mean computed from available subframes, an additivity-based self-consistency across sub-intervals, and a zero-interval boundary condition motivated by the MeanFlow analogy that anchors sharp-frame predictions.

Reconstruction loss (empirical supervision from subframes). Conventional motion-blur datasets directly provide the sharp subframes $\{S(\tau_i)\}$ that correspond to each blurry observation. This gives a direct ground-truth signal not available in MeanFlow's setting: we compute the empirical interval mean by averaging the subframes that fall within $[s,t]$,

$$\hat{B}(s,t) = \frac{1}{M_{s,t}} \sum_{\tau_i=s}^{t} S(\tau_i), \tag{10}$$

where $\hat{B}$ denotes the interval mean, and $M_{s,t}$ is the number of subframes within the interval $[s,t]$. Since $\hat{B}$ provides an empirical estimate of $B(s,t)$ from available subframes, we follow the MeanFlow analogy as discussed in Sec. 3.2. Accordingly, regressing f_θ to its encoded counterpart mirrors the form of $\mathcal{L}_{\text{MeanFlow}}$.

$$\mathcal{L}_{\text{recon}} = \mathbb{E}\left\| f_\theta(z_{\text{blur}}, s, t) - \text{sg}\big(\mathcal{E}(\hat{B}(s,t))\big) \right\|_2^2, \tag{11}$$

with (s,t) sampled uniformly from the set of subframe-aligned pairs $(\tau_i, \tau_j) : 0 \le i \le j \le M-1$ (Appendix A).

Additivity loss (self-consistency across sub-intervals). While $\mathcal{L}_{\text{recon}}$ supervises sampled interval queries, it provides no structural guarantee that f_θ behaves as a true interval mean function. We therefore enforce the blur additivity identity (Eq. 8) as a self-consistency constraint. Substituting f_θ into the right-hand side of Eq. 8 yields the additivity-form training target (the blur-domain analog of SplitMeanFlow's self-consistency target [11]), evaluable with two forward passes through f_θ without any time-derivative computation:

$$\mathcal{E}(\hat{B}(s,t)) = \omega\, f_\theta(z_{\text{blur}}, s, m) + (1-\omega)\, f_\theta(z_{\text{blur}}, m, t). \tag{12}$$

Regressing $f_\theta(z_{\text{blur}}, s, t)$ to this target yields the additivity loss:

$$\mathcal{L}_{\text{add}} = \mathbb{E}\Big\| f_\theta(z_{\text{blur}}, s, t) - \omega f_\theta(z_{\text{blur}}, s, m) - (1-\omega) f_\theta(z_{\text{blur}}, m, t) \Big\|_2^2, \tag{13}$$

which enforces consistency across overlapping intervals by sampling m uniformly from the subframe timestamps strictly between s and t, $\{\tau_k : s < \tau_k < t\}$, thereby propagating the point-wise supervision of $\mathcal{L}_{\text{recon}}$ into a coherent function over the full $[s, t]$ domain.[1]

Sharp-frame loss (the deblurring boundary case). Single-image deblurring at an arbitrary timestamp τ corresponds to the zero-interval query $s = t = \tau$ of f_θ, with target $S(\tau)$ as defined in Eq. 9. We supervise this case directly using the GT subframe at τ, with a pixel-space ℓ_1 distance and a perceptual EA-LPIPS distance [23] computed after decoding, to encourage sharp, photo-realistic outputs:

$$\mathcal{L}_{\text{sharp}} = \mathbb{E}_\tau \Big[\big\| \mathcal{D}(f_\theta(z_{\text{blur}}, \tau, \tau)) - S(\tau) \big\|_1 + \lambda_{\text{perc}} \, \text{EA-LPIPS}\big(\mathcal{D}(f_\theta(z_{\text{blur}}, \tau, \tau)),\, S(\tau)\big) \Big], \tag{14}$$

with τ sampled uniformly from the subframe timestamps $\{\tau_i\}_{i=0}^{M-1}$. This anchors the deblurring output directly with GT supervision, complementing the interval-level signals from $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{add}}$.

Total objective. The complete loss combines the three terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{add}} \mathcal{L}_{\text{add}} + \lambda_{\text{sharp}} \mathcal{L}_{\text{sharp}}. \tag{15}$$

The auxiliary weights are set to $(\lambda_{\text{add}}, \lambda_{\text{sharp}}, \lambda_{\text{perc}}) = (0.1, 1.0, 2.0)$, prioritizing the empirical reconstruction signal $\mathcal{L}_{\text{recon}}$; the full warm-up schedule is provided in Appendix A.

3.4 Inference

A single trained f_θ unifies three inference modes by varying the interval query, each evaluated in one forward pass without retraining or iterative sampling. For *single-image deblurring*, we query at the zero-interval limit $(s, t) = (T/2 - \epsilon,\, T/2 + \epsilon)$ to recover the center sharp frame, where ϵ denotes a small constant. For *blur-to-video* with N output frames, we query at N uniformly spaced zero-intervals $\{(\tau_i - \epsilon,\, \tau_i + \epsilon)\}_{i=0}^{N-1}$ with $\tau_i = i/(N-1)$, recovering the sharp sequence over the exposure. For *pick-a-moment* recovery, we query at $(\tau - \epsilon,\, \tau + \epsilon)$ for any user-specified $\tau \in [0, T]$. All three modes share the same parameters and produce deterministic outputs given the same blurry input. Here, deterministic refers to the inference procedure, which samples no noise as in [49, 23]; although built on a pretrained generative prior, PickMoment does not retain the sampling-based generative capability of stochastic inference. We use $\epsilon = 0.01$ as the default zero-interval width.

4 Experiments

4.1 Experimental Setup

Datasets. We train on the GoPro dataset [30]. Each blurry image is synthesized by averaging 7–13 consecutive sharp frames with a known temporal order over the exposure window, and we use all of these subframes as supervision targets during training. We train only on the GoPro training split and apply the same model to all evaluation datasets without any per-dataset fine-tuning. We conduct experiments on GoPro [30], HIDE [37], RealBlur-J, and RealBlur-R [34] for single-frame evaluation. For blur-to-video evaluation, we use GoPro-7, the 7-frame variant of the GoPro test split established by Jin et al. [16] and adopted by subsequent work [43], where each blurred image is paired with seven sharp frames spanning the exposure.

Metrics. For single-image deblurring, we report PSNR, SSIM [48], LPIPS [53], and DISTS [7] computed against the center sharp frame of each blur exposure, following the FideDiff [23] setup. For blur-to-video, we additionally report the bidirectional patch-based variants PSNR_p, SSIM_p, and LPIPS_p according to Blur2Vid [43], which take a per-patch minimum between the prediction and its temporally-reversed counterpart against the ground-truth sequence to account for the directional ambiguity of motion-blur inversion, together with FVD [46] and the end-point error (EPE) of optical flow computed via RAFT [44]. All metrics are measured at 640×360 resolution to ensure a fair comparison with prior blur-to-video methods [16, 54, 43].

[1] Strictly, the additivity in Eq. 8 holds in pixel space; applying it to the latent outputs of f_θ (Eq. 13) relies on the approximate linearity of the VAE encoder over perceptually similar inputs, and acts as an effective structural prior in practice (validated in Appendix B). A pixel-space variant would require additional decoder evaluations at each training step, incurring substantial computational overhead and increased training time.

Table 1: Quantitative comparison for single-image deblurring. **Bold** indicates the best result and underline indicates the second-best result among generative-prior-based methods.

Dataset	Metric	Restoration-based				Generative-prior-based				
		Restormer [51]	HI-Diff [5]	UFPNet [8]	AdaRevD [29]	DiffBIR [21]	OSEDiff [49]	Diff-Plugin [25]	FideDiff [23]	Ours
GoPro	PSNR↑	32.92	33.33	34.09	34.60	26.15	24.34	22.88	28.79	**29.43**
	SSIM↑	0.9611	0.9642	0.9686	0.9716	0.8377	0.8228	0.7798	0.9148	**0.9253**
	LPIPS↓	0.0841	0.0799	0.0764	0.0712	0.2366	0.1738	0.2332	0.0831	**0.0802**
	DISTS↓	0.0724	0.0710	0.0666	0.0672	0.1460	0.0834	0.1166	0.0525	**0.0429**
HIDE	PSNR↑	31.22	31.46	31.74	32.35	25.94	23.20	21.94	27.28	**27.52**
	SSIM↑	0.9423	0.9446	0.9471	0.9525	0.8216	0.7611	0.7272	0.8775	**0.8891**
	LPIPS↓	0.1082	0.1055	0.0931	0.0872	0.2091	0.2208	0.2732	0.1068	**0.1046**
	DISTS↓	0.0730	0.0725	0.0676	0.0666	0.1246	0.1001	0.1411	0.0647	**0.0553**
RealBlur-J	PSNR↑	28.96	29.15	29.87	30.12	26.92	26.83	25.77	**28.96**	28.94
	SSIM↑	0.8786	0.8898	0.8840	0.8945	0.7450	0.8004	0.7711	0.8695	**0.8757**
	LPIPS↓	0.1561	0.1470	0.1441	0.1408	0.2587	0.1793	0.2055	**0.1142**	0.1184
	DISTS↓	0.1112	0.1050	0.1101	0.1037	0.1599	0.1198	0.1355	**0.0800**	0.0841
RealBlur-R	PSNR↑	36.19	36.28	36.25	36.53	32.60	33.54	32.64	36.01	**36.08**
	SSIM↑	0.9572	0.9583	0.9528	0.9570	0.8493	0.9056	0.8496	0.9424	**0.9510**
	LPIPS↓	0.0608	0.0602	0.0615	0.0621	0.3388	0.1057	0.1422	**0.0584**	0.0612
	DISTS↓	0.0833	0.0814	0.0854	0.0846	0.2576	0.1322	0.1673	0.0862	**0.0833**



Figure 3: Qualitative comparison for single-image deblurring on GoPro, HIDE, and RealBlur-J (top to bottom). Best viewed zoomed in.

Training details. We optimize with Adam [17] for a total of 100K iterations. The learning rate is linearly warmed up to 10^{-4} over the first 2K steps and then cosine-annealed [26] to 10^{-5}. We use bf16 mixed precision with gradient accumulation of 2 steps. Following Restormer [51], we adopt progressive learning where the (patch size, batch size) pairs are updated to $[(256^2, 8), (384^2, 4), (512^2, 2)]$ at iterations $[30\text{K}, 60\text{K}, 100\text{K}]$. Training is performed on a single NVIDIA RTX A6000 GPU and takes around 4 days. Full hyperparameters are given in Appendix A.

4.2 Single-Image Deblurring Results

We compare PickMoment against representative single-image deblurring methods spanning two families: *restoration-based* approaches (Restormer [51], HI-Diff [5], UFPNet [8], AdaRevD [29]) and *generative-prior-based* ones (DiffBIR [21], OSEDiff [49], Diff-Plugin [25], FideDiff [23]). For deblurring, PickMoment is evaluated using the zero-interval limit $(s, t) = (T/2 - \epsilon, T/2 + \epsilon)$.

Quantitative results. Within the family of generative-prior-based methods, Table 1 shows that PickMoment achieves state-of-the-art performance on GoPro and HIDE across all four metrics, surpassing FideDiff by $+0.64$ dB PSNR on GoPro and $+0.24$ dB on HIDE. Results on RealBlur-J/R are competitive with the closely matched FideDiff, while PickMoment achieves better performance on 12 out of 16 evaluation metrics overall. Transformer-based baselines such as AdaRevD [29] achieve higher per-pixel PSNR but predict only a single sharp frame and do not natively support blur-to-video.

Qualitative results. Fig. 3 compares deblurred outputs on representative samples from GoPro, HIDE, and RealBlur. PickMoment restores sharper local structures than existing diffusion-based baselines, especially around vehicle boundaries, facial details, and fine texture patterns. The results show

Table 2: Quantitative comparison for blur-to-video on GoPro-7.

Method	$\text{PSNR}_p\uparrow$	$\text{SSIM}_p\uparrow$	$\text{LPIPS}_p\downarrow$	FVD↓	EPE↓
Jin et al. [16]	25.23	0.8190	0.084	235.53	3.38
MotionETR [54]	26.54	0.8825	0.015	94.90	1.46
Blur2Vid [43]	<u>30.01</u>	<u>0.9359</u>	<u>0.012</u>	**21.46**	**0.39**
Ours	**31.24**	**0.9388**	**0.0086**	<u>27.40</u>	<u>0.86</u>

Table 3: Generalization ablation on unseen timestamps and $\mathbf{B} \rightarrow \mathbf{B}$ denotes our proposed method.

Method	Metric	$\tau = 2$	$\tau = 4$	$\tau = 6$	$\tau = 8$	$\tau = 10$	Mean
$\mathbf{B} \rightarrow \mathbf{S}$	PSNR↑	26.51	27.47	28.42	27.41	26.52	27.27
	SSIM↑	0.8273	0.8450	0.8626	0.8450	0.8288	0.8418
$\mathbf{B} \rightarrow \mathbf{B}$	PSNR↑	**27.34**	**28.34**	**29.44**	**28.30**	**27.55**	**28.19**
	SSIM↑	**0.8447**	**0.8682**	**0.8860**	**0.8681**	**0.8560**	**0.8646**

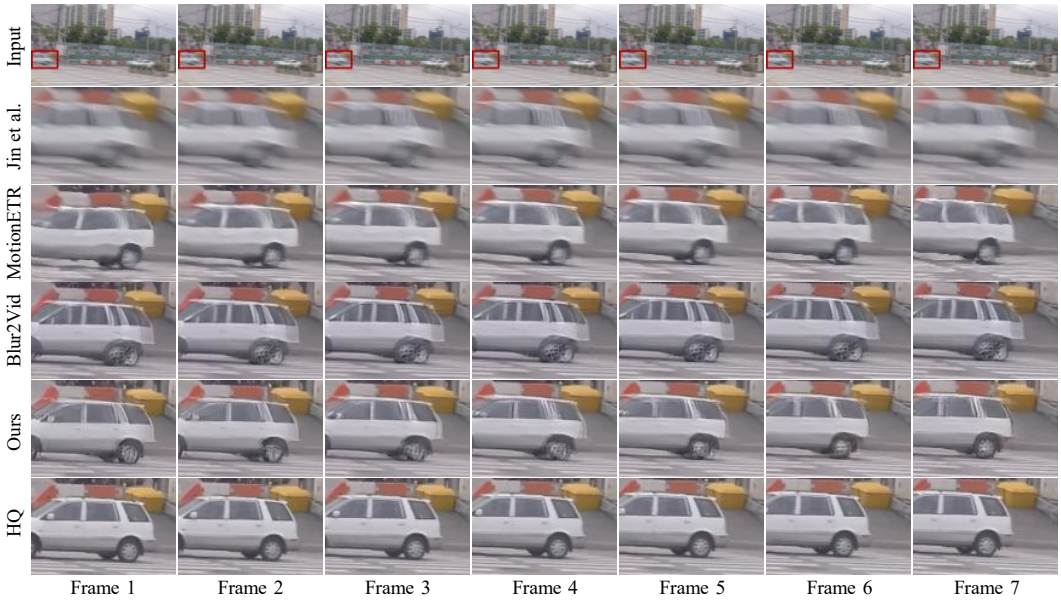


Figure 4: Qualitative comparison for blur-to-video on GoPro-7. Best viewed zoomed in.

clearer edges, reduced residual blur, and fewer structural distortions. Additional visual comparisons are provided in Appendix H.

4.3 Blur-to-Video Results

We compare PickMoment against prior blur-to-video methods that recover an in-exposure sharp sequence from a single blurred input: Jin et al. [16], MotionETR [54], and Blur2Vid [43], which is the state-of-the-art video-diffusion-based approach. Given a blurred input, PickMoment reconstructs the full exposure-time sequence through predictions of f_θ at N uniformly spaced intervals (Sec. 3.4).

Quantitative results. Table 2 shows that PickMoment achieves the highest per-frame fidelity, surpassing Blur2Vid by $+1.23$ dB in PSNR_p and reducing LPIPS_p by a relative margin of 28%. Blur2Vid retains an advantage on temporal smoothness (FVD, EPE), reflecting an architectural trade-off: deterministic per-interval prediction maximizes per-frame quality, whereas video-diffusion sampling provides implicit temporal regularization. PickMoment needs one forward pass per output frame, issued as a single batch ($7.90\,\text{s}$ for a 7-frame sequence), in contrast to Blur2Vid's iterative sampling, yielding roughly a $25\times$ inference speedup (Appendix G). A warping-based check (Appendix F) shows that our frame-to-frame change is close to that of the ground truth (warp L1 0.0089 vs. 0.0096), whereas Blur2Vid is smoother than the ground truth (0.0072). Across all 1,744 GoPro-7 clips, PickMoment also commits to one motion direction rather than blending both (median forward/reverse PSNR gap 9.2 dB), which we report as an empirical characterization rather than a guarantee.

Qualitative results. Fig. 4 compares recovered in-exposure sequences against Blur2Vid [43] on GoPro-7. Jin et al. [16] leaves substantial blur and recovers little sharp structure, while MotionETR noticeably distorts the vehicle geometry. Blur2Vid produces more plausible frames, but residual blur and softened details remain around the vehicle body and wheels. PickMoment better restores sharp object boundaries and local textures, yielding visually clearer and more structurally faithful in-exposure frames. Additional visual comparisons are provided in Appendix H.

Table 4: Advantage of interval prediction on GoPro. $\mathbf{B} \rightarrow \mathbf{B}$ denotes our proposed method.

Method	Deblurring				Blur-to-video				
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	PSNR_p↑	SSIM_p↑	LPIPS_p↓	FVD↓	EPE↓
$\mathbf{B} \rightarrow \mathbf{S}$	27.72	0.8944	0.1595	0.1149	29.16	0.8894	0.0292	59.51	1.15
$\mathbf{B} \rightarrow \{\mathbf{S}(\tau_\mathbf{i})\}_{\mathbf{i=0}}^{\mathbf{N-1}}$	27.76	0.8945	0.1604	0.1166	29.26	0.8933	0.0278	**58.76**	1.13
$\mathbf{B} \rightarrow \mathbf{B}$ (ours)	**28.45**	**0.9080**	**0.1455**	**0.1043**	**29.78**	**0.9034**	**0.0271**	59.29	**1.02**

Table 5: Loss-term ablation on GoPro. ✓ indicates the loss is enabled.

Losses			Deblurring				Blur-to-video				
$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{add}}$	$\mathcal{L}_{\text{sharp}}$	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	PSNR_p↑	SSIM_p↑	LPIPS_p↓	FVD↓	EPE↓
		✓	29.06	0.9208	0.0812	0.0438	30.17	0.9186	0.0104	38.07	1.33
✓			28.45	0.9080	0.1455	0.1043	29.78	0.9034	0.0271	59.29	1.02
✓	✓		28.56	0.9096	0.1422	0.1020	29.95	0.9058	0.0260	58.91	0.92
✓		✓	29.24	0.9236	**0.0787**	**0.0420**	30.68	0.9255	0.0146	67.10	0.99
✓	✓	✓	**29.43**	**0.9253**	0.0802	0.0429	**31.24**	**0.9388**	**0.0086**	**27.40**	**0.86**

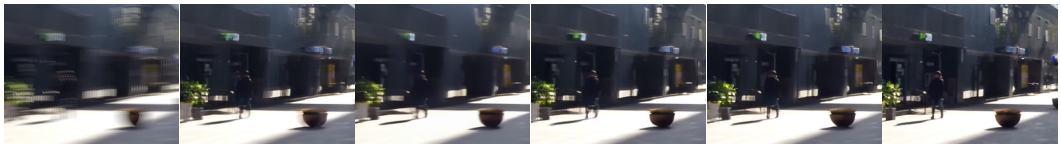


Figure 5: Loss-term ablation visualization on GoPro-7 (frame 6 of 7). Best viewed zoomed in.

4.4 Ablation Studies

We conduct three ablations to study (i) generalization to unseen timestamps, (ii) the choice of supervision target, and (iii) the contribution of each loss term. All variants share the same backbone, training data, and training budget, and are trained using only $\mathcal{L}_{\text{recon}}$ in the latent space unless stated otherwise. This isolates the effect of the supervision target, while the impact of the loss design is analyzed separately in the loss ablation study. Three baselines are used: $(\mathbf{B} \rightarrow \mathbf{S})$, which learns a blur-to-single-sharp-frame mapping by predicting a single sharp frame $S(\tau)$ at a given discrete timestamp τ; $(\mathbf{B} \rightarrow \{\mathbf{S}(\tau_\mathbf{i})\}_{\mathbf{i=0}}^{\mathbf{N-1}})$, which learns a blur-to-multi-sharp-frame mapping by predicting N sharp frames simultaneously through a multi-channel output at predefined timestamps from a single blurred image, following [16]; and **Ours ($\mathbf{B} \rightarrow \mathbf{B}$)**, which predicts arbitrary interval-mean blurs from a blurred image. Because $\mathbf{B} \rightarrow \mathbf{S}$ is a conventional sharp-frame predictor on the same pretrained backbone, Table 4 also serves as a capacity-matched comparison, and a large pretrained prior is standard among the generative-prior-based baselines [21, 49, 25, 23].

Generalization to unseen timestamps. In Table 3, we evaluate whether interval supervision generalizes to timestamps unseen during training, as our model learns continuous-time blur representations rather than timestamp-specific predictions which are inaccessible to previous blur-to-video models. Both $\mathbf{B} \rightarrow \mathbf{S}$ and Ours are trained on only the odd-indexed subframes, and we evaluate at the unseen even-indexed timestamps $\tau \in \{\frac{2}{11}, \frac{4}{11}, \frac{6}{11}, \frac{8}{11}, \frac{10}{11}\}$. Ours outperforms $\mathbf{B} \rightarrow \mathbf{S}$ at every timestamp, with $+0.92$ dB Mean PSNR and $+0.023$ Mean SSIM values, indicating that interval supervision generalizes more smoothly across continuous time than direct sharp-frame prediction. Beyond held-out GoPro timestamps, we evaluate zero-shot on X4K1000FPS [41], where averaging 33 consecutive 1000-fps frames leaves ground truth at 33 timestamps per exposure: PickMoment outperforms Blur2Vid at every timestamp, with a 1.35 dB higher mean PSNR (Appendix E). Querying the endpoint of a synthesized half-exposure blur, an input length unseen in training, also stays within 1 dB of center-frame deblurring on the full blur (28.49 vs. 29.43 dB), which we view as supporting evidence for a continuous interval-mean field rather than proof.

Advantage of continuous-time interval prediction. Table 4 compares the three supervision targets on the standard evaluation. Interval-mean supervision dominates on every metric except FVD, where $\mathbf{B} \rightarrow \{\mathbf{S}(\tau_\mathbf{i})\}$ is marginally better, indicating that learning the integral yields stronger representations than learning direct sharp-frame prediction.

Effectiveness of loss functions. We analyze the contribution of each loss term in the full objective (Table 5, Fig. 5). Using $\mathcal{L}_{\text{recon}}$ alone produces coarse outputs (LPIPS 0.1455), while $\mathcal{L}_{\text{sharp}}$ nearly halves the LPIPS value, accounting for most of the perceptual improvement. $\mathcal{L}_{\text{add}}$ alone yields only marginal gains, but combined with $\mathcal{L}_{\text{sharp}}$ it reduces FVD from 67.10 to 27.40 and EPE from 0.99 to 0.86, showing that the two losses are complementary: $\mathcal{L}_{\text{sharp}}$ enforces per-frame sharpness while $\mathcal{L}_{\text{add}}$ propagates temporal coherence across the recovered sequence (Fig. 5). In particular, $\mathcal{L}_{\text{add}}$ is especially effective for blur-to-video task, where temporal consistency is critical. With $\mathcal{L}_{\text{sharp}}$ alone, deblurring stays close to the full objective while blur-to-video degrades (FVD 38.07 vs. 27.40), so the interval-mean formulation mainly improves temporal coherence.

5 Conclusion

We introduced PickMoment, a continuous-time reformulation of motion deblurring that predicts the interval-mean blur over arbitrary sub-intervals of the exposure. A single deterministic model unifies single-image deblurring, blur-to-video, and pick-a-moment recovery as different queries, trained with three supervisions derived from the integral definition: an empirical reconstruction loss, an additivity loss for self-consistency across sub-intervals, and a sharp-frame loss at the zero-interval limit. PickMoment achieves state-of-the-art single-image deblurring among generative-prior-based methods on GoPro and HIDE, competitive performance on RealBlur, and the highest per-frame fidelity on GoPro-7 blur-to-video, all with one forward pass per output frame. The main remaining gap is temporal smoothness, where iterative video-diffusion sampling still provides stronger implicit regularization; combining our deterministic interval-mean prediction with iterative sampling is a natural next step. We discuss further limitations in Appendix I.

Acknowledgments and Disclosure of Funding

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2026 (Project Name: Development of AI-Based Animation Production Technology to Ensure Character and Scene Consistency and Continuity for Enhanced Efficiency and Quality, Project Number: RS-2026-25525207, Contribution Rate: 30%). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their 3D metaverse transformation), and IITP grant funded by the Korea government (MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)).

References

[1] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, Chaoning Zhang, and In So Kweon. Restoration of video frames from a single blurred image with motion understanding. In *CVPR Workshops (NTIRE)*, 2021.

[3] Black Forest Labs. FLUX.2. `https://github.com/black-forest-labs/flux2`, 2025.

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, 2022.

[5] Zheng Chen, Yulun Zhang, Ding Liu, Bin Xia, Jinjin Gu, Linghe Kong, and Xin Yuan. Hierarchical integration diffusion model for realistic image deblurring. In *NeurIPS*, 2023.

[6] Junhao Cheng, Wei-Ting Chen, Xi Lu, and Ming-Hsuan Yang. Unpaired deblurring via decoupled diffusion model. *arXiv preprint arXiv:2502.01522*, 2025.

[7] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[8] Zhenxuan Fang, Fangfang Wu, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *CVPR*, 2023.

[9] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (SIGGRAPH)*, 2006.

[10] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[11] Yi Guo, Wei Wang, Zhihang Yuan, Rong Cao, Kuan Chen, Zhengyang Chen, Yuanyuan Huo, Yang Zhang, Yuping Wang, Shouda Liu, and Yuxuan Wang. SplitMeanFlow: Interval splitting consistency in few-step generative modeling. *arXiv preprint arXiv:2507.16884*, 2025.

[12] Jin-Ting He, Fu-Jen Tsai, Yan-Tsung Peng, Min-Hung Chen, Chia-Wen Lin, and Yen-Yu Lin. BlurDM: A blur diffusion model for image deblurring. In *NeurIPS*, 2025.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[14] Zhe Hu, Li Xu, and Ming-Hsuan Yang. Joint depth estimation and camera shake removal from single blurry image. In *CVPR*, 2014.

[15] Eun Woo Im, Junseong Shin, Sungyong Baik, and Tae Hyun Kim. Deep variational bayesian modeling of haze degradation process. In *CIKM*, 2023.

[16] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *CVPR*, 2018.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[18] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[19] Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *CVPR*, 2023.

[20] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019.

[21] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Wanli Ouyang, Yu Qiao, and Chao Dong. Diffbir: Towards blind image restoration with generative diffusion prior. In *ECCV*, 2024.

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

[23] Xiaoyang Liu, Zhengyan Zhou, Zihang Xu, Jiezhang Cao, Zheng Chen, and Yulun Zhang. Fidediff: Efficient diffusion model for high-fidelity image motion deblurring. In *ICLR*, 2026.

[24] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[25] Yuhao Liu, Zhanghan Ke, Fang Liu, Nanxuan Zhao, and Rynson W. H. Lau. Diff-plugin: Revitalizing details for diffusion-based low-level tasks. In *CVPR*, 2024.

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[27] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

[28] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V. Sander. Deblur-NeRF: Neural radiance fields from blurry images. In *CVPR*, 2022.

[29] Xintian Mao, Qingli Li, and Yan Wang. Adarevd: Adaptive patch exiting reversible decoder pushes the limit of image deblurring. In *CVPR*, 2024.

[30] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017.

[31] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016.

[32] Bang-Dang Pham, Phong Tran, Anh Tran, Cuong Pham, Rang Nguyen, and Minh Hoai. Hypercut: Video sequence from a single blurry image using unsupervised ordering. In *CVPR*, 2023.

[33] Kuldeep Purohit, Anshul Shah, and A. N. Rajagopalan. Bringing alive blurred moments. In *CVPR*, 2019.

[34] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[36] Mohamed Sayed and Gabriel Brostow. Improved handling of motion blur in online object detection. In *CVPR*, 2021.

[37] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019.

[38] Junseong Shin and Tae Hyun Kim. PRISM: Latent composition consistency for single-image reflection removal. In *ECCV*, 2026.

[39] Junseong Shin, Seungwoo Chung, Yunjeong Yang, and Tae Hyun Kim. HazeFlow: Revisit haze physical model as ode and non-homogeneous haze generation for real-world dehazing. In *ICCV*, 2025.

[40] Junseong Shin, Kijun Kim, Minseong Kim, Dongjin Kim, and Tae Hyun Kim. ODDR: One-step deshadow diffusion via reward guidance. In *NeurIPS*, 2026.

[41] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: extreme video frame interpolation. In *ICCV*, 2021.

[42] Xin Tao, Hongyun Gao, Yi Wang, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

[43] SaiKiran Tedla, Kelly Zhu, Trevor Canham, Felix Taubner, Michael S. Brown, Kiriakos N. Kutulakos, and David B. Lindell. Generating the past, present and future from a motion-blurred image. *ACM Transactions on Graphics (Blur2Vid)*, 44(6), 2025.

[44] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[45] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *ECCV*, 2022.

[46] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. *ICLR Workshop*, 2019.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[49] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. In *NeurIPS*, 2024.

[50] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

[51] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[52] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, 2019.

[53] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[54] Youjian Zhang, Chaoyue Wang, Stephen J. Maybank, and Dacheng Tao. Exposure trajectory recovery from motion blur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[55] Zhihang Zhong, Xiao Sun, Zhirong Wu, Yinqiang Zheng, Stephen Lin, and Imari Sato. Animation from blur: Multi-modal blur decomposition with motion guidance. In *ECCV*, 2022.

A Training Details

Architecture and trainable parameters. We instantiate f_θ on FLUX.2-klein-4B [3], a $\sim$4B-parameter diffusion transformer paired with a pretrained VAE $(\mathcal{E}, \mathcal{D})$. We perform full fine-tuning of the transformer only (no LoRA [13] or partial freezing); the VAE encoder and decoder are kept frozen throughout training. f_θ is parameterized as a residual displacement from the encoded blur input, $f_\theta(z_{\text{blur}}, s, t) := z_{\text{blur}} + h_\theta(z_{\text{blur}}, s, t)$, where h_θ is the transformer output. Interval conditioning replaces FLUX's original timestep conditioning: the triple $(s, t, t-s)$ is sinusoidally embedded, projected by a shared MLP, and injected through the existing AdaLN pathway.

Data pipeline. We use the GoPro training split of 2,103 blurry/sharp pairs, where each blurry image is paired with 7–13 consecutive subframes that compose it. At each training step we apply a random crop (with the same crop coordinates shared across the blurry image and all of its paired subframes to preserve pixel correspondence) and a random horizontal flip; no color jitter or other augmentations are used. Pixel values are normalized to $[0, 1]$ before VAE encoding, matching the input convention used by FLUX.2. The DataLoader uses 4 workers, and the random seed is fixed for reproducibility.

Loss sampling. All three loss terms (Eqs. 11, 13, 14) are computed in the same batch and averaged with mean reduction:

- $\mathcal{L}_{\text{recon}}$: (s, t) is sampled uniformly from the set of subframe-aligned pairs $\{(\tau_i, \tau_j) : 0 \le i \le j \le M-1\}$. With probability 0.25, we constrain to zero-interval pairs $i = j$ (so $s = t = \tau_i$ and $\hat{B}(s, t) = S(\tau_i)$), ensuring that the inference regime $s \approx t$ used for deblurring and pick-a-moment recovery is densely covered during training. While inference uses a small but nonzero ϵ (e.g., $\epsilon = 0.01$), the $s = t$ branch and $\mathcal{L}_{\text{sharp}}$ at τ both regress to the same sharp-frame target $S(\tau)$, so the predictor at $|t - s| \approx \epsilon$ inherits this supervision by continuity.
- $\mathcal{L}_{\text{add}}$: the splitting point m is sampled uniformly from the subframe timestamps strictly between s and t, $\{\tau_k : s < \tau_k < t\}$.
- $\mathcal{L}_{\text{sharp}}$: since ground-truth sharp frames are only available at subframe timestamps, τ is sampled uniformly from the discrete set $\{i/(M-1) : i = 0, \ldots, M-1\}$, where M is the number of subframes paired with the blurry input. The EA-LPIPS perceptual loss reproduces the implementation of FideDiff [23].

Optimization and progressive learning. We optimize with Adam [17] for a total of 100K iterations. The learning rate is linearly warmed up to 10^{-4} over the first 2K steps and then cosine-annealed [26] to 10^{-5}. We use bf16 mixed precision with gradient accumulation of 2 steps. Following Restormer [51], we adopt progressive learning, updating (patch size, batch size) pairs to $[(256^2, 8), (384^2, 4), (512^2, 2)]$ at iterations $[30\text{K}, 60\text{K}, 100\text{K}]$. The auxiliary loss weights $(\lambda_{\text{add}}, \lambda_{\text{sharp}}, \lambda_{\text{perc}}) = (0.1, 1.0, 2.0)$ are linearly warmed up from zero to their target values over the first patch stage (30K iterations), allowing $\mathcal{L}_{\text{recon}}$ to establish a stable baseline before the structural and boundary constraints take effect. We do not use Exponential Moving Average (EMA).

Hardware and wall-clock time. All training runs on a single NVIDIA RTX A6000 GPU and takes approximately 4 days to complete.

Table 6: **Training hyperparameters.**

Backbone	FLUX.2-klein-4B (transformer only fine-tuned, VAE frozen)
Trainable params	$\sim$4B (full transformer)
Optimizer	Adam
Total iterations	100K
LR schedule	Linear warmup to 10^{-4} over 2K, then cosine-annealing to 10^{-5}
Mixed precision	bf16
Gradient accumulation	2 steps
Progressive learning	$(256^2, 8) \to (384^2, 4) \to (512^2, 2)$ at 30K/60K/100K
$(\lambda_{\text{add}}, \lambda_{\text{sharp}}, \lambda_{\text{perc}})$	$(0.1, 1.0, 2.0)$, linear warmup from 0 over first 30K iterations
EMA	None
Augmentation	Random crop (shared across paired subframes) + horizontal flip
Pixel range	$[0, 1]$ before VAE
DataLoader workers	4
Hardware	$1 \times$ NVIDIA RTX A6000
Training time	$\sim$4 days

B Empirical Validation of Latent-Space Additivity

The additivity loss $\mathcal{L}_{\text{add}}$ (Eq. 13) enforces the identity $z_{s,t} \approx \omega\, z_{s,m} + (1-\omega)\, z_{m,t}$ in the VAE latent space, where $\omega = (m-s)/(t-s)$ and $z_{x,y} = \mathcal{E}(B(x,y))$. While the identity holds exactly in pixel space (by Eq. 8), its validity in latent space depends on the encoder being approximately linear over the manifold of blurry images. We empirically verify this on the FLUX.2-klein-4B VAE that we use throughout the paper. Approximate compositionality in a pretrained VAE latent space has also been exploited for reflection removal [38].

Protocol. We use the GoPro test split's high-frame-rate subframes (11 sequences, $\sim$1,100 frames each). For each sequence we sample 30 random 11-frame windows, with a shared 512×512 random crop per window. For each window we sample 10 random triples (s,t,m) with $0 \le s < m < t \le 1$, $|t-s| \ge 0.2$, and $|m-s|, |t-m| \ge 0.05$, yielding 2,983 triples in total. We deliberately exclude very small intervals from this protocol because the additivity identity governs the regime where multiple subframes are averaged into each side of Eq. 8; the zero-interval limit $s \approx t$ is supervised separately by $\mathcal{L}_{\text{sharp}}$ (and by the $s=t$ branch of $\mathcal{L}_{\text{recon}}$) directly against the ground-truth sharp frame, so it is not governed by the additivity assumption being validated here. For each triple we compute $B_{s,t}$, $B_{s,m}$, $B_{m,t}$ as the average of the subframes whose normalized timestamp falls in the corresponding interval, encode each through the frozen VAE to obtain $z_{s,t}$, $z_{s,m}$, $z_{m,t}$, and then form the additive prediction $z_{\text{pred}} = \omega z_{s,m} + (1-\omega)\, z_{m,t}$. We decode both $z_{s,t}$ and z_{pred} back to pixel space and report PSNR against the corresponding pixel-space references.

Metrics. Three pixel-space PSNR quantities isolate the source of error:

- $\mathsf{PSNR}\big(\mathcal{D}(z_{s,t}), B_{s,t}\big)$ measures the VAE round-trip error alone (the upper bound achievable without any additivity assumption);
- $\mathsf{PSNR}\big(\mathcal{D}(z_{\text{pred}}), \mathcal{D}(z_{s,t})\big)$ isolates the pure additivity error in pixel space, factoring out the round-trip degradation;
- $\mathsf{PSNR}\big(\mathcal{D}(z_{\text{pred}}), B_{s,t}\big)$ measures the combined error a downstream consumer would observe.

Results. Table 7 summarizes the three PSNR distributions over 2,983 triples. The VAE round-trip alone yields a mean PSNR of 42.70 dB, while the additive prediction $\mathcal{D}(z_{\text{pred}})$ matches the round-trip output at 39.82 dB on average. The additivity assumption thus introduces an additional $\sim$3 dB error on top of the VAE's intrinsic reconstruction floor, roughly twice its MSE. This is of the same order as the round-trip error itself, so it is not a new dominant source of error, and it enters only as a training signal rather than as a constraint at inference. A pixel-space variant would avoid this approximation but require additional decoder evaluations at each training step. The combined error against the true pixel target $B_{s,t}$ remains at 37.40 dB, comfortably above the noise floor at which deblurring metrics saturate. We further verified that the additivity error is essentially uncorrelated with both the interval width $|t-s|$ and the splitting position ω, indicating that the encoder's near-linearity is uniform over the parameter range used during training rather than concentrated near a degenerate corner. These observations justify applying $\mathcal{L}_{\text{add}}$ directly in the latent space without an explicit decode step.

Table 7: **Latent-space additivity validity.** PSNR distributions over $N = 2{,}983$ random (s,t,m) triples drawn from 11 GoPro test sequences (512×512 crops). The pure additivity error $\mathcal{D}(z_{\text{pred}})$ vs. $\mathcal{D}(z_{s,t})$ is comparable to the VAE round-trip floor, indicating that $\mathcal{E}$ is approximately linear over the blur-image manifold.

PSNR (dB)	Description	Mean	Median	Std.	10th pct.	90th pct.
$\mathcal{D}(z_{s,t})$ vs. $B_{s,t}$	(VAE round-trip only)	42.70	43.12	2.32	39.27	45.24
$\mathcal{D}(z_{\text{pred}})$ vs. $\mathcal{D}(z_{s,t})$	(pure additivity)	39.82	39.38	4.49	34.36	45.94
$\mathcal{D}(z_{\text{pred}})$ vs. $B_{s,t}$	(combined)	37.40	37.26	2.80	33.83	41.03

C Center Frame Quality on GoPro-7

We additionally report the quality of the center sharp frame produced by each blur-to-video method on the GoPro-7 split (1,744 clips at 640×360 resolution, the same evaluation protocol as Table 2; frame index 3 of the 7-frame sequence is taken as the center). This complements Table 2's sequence-level metrics by isolating per-frame deblurring quality at the exposure midpoint within each method's recovered sequence. Table 8 reports full-image PSNR, SSIM, LPIPS, and DISTS against the corresponding ground-truth sharp frame. *Note: this evaluation is not directly comparable to Table 1, which uses the standard GoPro test split with 11-frame averaging at full resolution; the protocols differ in averaging length, resolution, and test composition.*

Table 8: **Center frame quality on the GoPro-7 split.** 1,744 clips at 640×360. Frame index 3 of the recovered 7-frame sequence is evaluated against the corresponding ground-truth sharp frame using full-image metrics.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓
Jin et al. [16]	29.44	0.9237	0.1345	0.0982
MotionETR [54]	28.60	0.9415	0.0509	0.0495
Blur2Vid [43]	30.66	0.9528	0.0363	0.0380
Ours	**33.22**	**0.9664**	**0.0291**	**0.0298**

D Additional Ablation Studies

This section extends the ablations of Sec. 4.4 with cross-dataset quantitative results and qualitative visualizations.

Cross-dataset loss ablation. Table. 9 extends the loss-term ablation of Table. 5 from GoPro to the full evaluation suite (GoPro, HIDE, RealBlur-J, RealBlur-R), measured on single-image deblurring metrics. The relative ordering established on GoPro holds across all four datasets: $\mathcal{L}_{\text{add}}$ alone gives marginal gains over $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{sharp}}$ provides the bulk of the perceptual improvement (LPIPS, DISTS), and the full objective achieves the best PSNR/SSIM on every dataset. This consistency across both synthetic (GoPro, HIDE) and real (RealBlur-J/R) blur indicates that the contribution of each identity-derived loss generalizes beyond the GoPro training distribution.

Table 9: Cross-dataset loss-term ablation on single-image deblurring metrics. Extension of Table. 5.

Dataset	Metric	$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{add}}$	$\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{sharp}}$	Full
GoPro	PSNR↑	28.45	28.56	29.24	**29.43**
	SSIM↑	0.9080	0.9096	0.9236	**0.9253**
	LPIPS↓	0.1455	0.1422	**0.0787**	0.0802
	DISTS↓	0.1043	0.1020	**0.0420**	0.0429
HIDE	PSNR↑	26.73	26.79	27.49	**27.52**
	SSIM↑	0.8668	0.8687	**0.8894**	0.8891
	LPIPS↓	0.1691	0.1682	**0.1015**	0.1046
	DISTS↓	0.1284	0.1257	**0.0539**	0.0553
RealBlur-J	PSNR↑	28.48	28.53	28.89	**28.94**
	SSIM↑	0.8502	0.8518	0.8738	**0.8757**
	LPIPS↓	0.1484	0.1525	**0.1184**	**0.1184**
	DISTS↓	0.1121	0.1138	0.0847	**0.0841**
RealBlur-R	PSNR↑	35.69	35.67	36.01	**36.08**
	SSIM↑	0.9461	0.9444	0.9500	**0.9510**
	LPIPS↓	0.1032	0.1082	**0.0612**	**0.0612**
	DISTS↓	0.1314	0.1361	0.0847	**0.0833**

Advantage of interval prediction. Fig. 6 provides additional qualitative comparisons for the supervision-target ablation in Table. 4, contrasting the three baselines on a representative GoPro sequence. The $\mathbf{B} \to \mathbf{S}$ and $(\mathbf{B} \to \{\mathbf{S}(\tau_i)\}_{i=0}^{N-1})$ baselines recover blurrier sequences with weaker per-frame texture, while interval-mean supervision yields sharper content with motion that better aligns with the ground-truth subframes, supporting the quantitative gains in Table. 4.

Generalization to unseen timestamps. Fig. 7 provides additional qualitative results for the unseen-timestamp ablation in Table. 3, where both PickMoment and the $\mathbf{B} \to \mathbf{S}$ baseline are trained only on odd-indexed subframes and evaluated at the held-out even-indexed timestamps. PickMoment recovers visibly sharper content at the held-out timestamps, indicating that interval-mean supervision generalizes more smoothly across continuous time than direct sharp-frame prediction.

Effect of each loss term. Figs. 8 and 9 visualize the complementary roles of $\mathcal{L}_{\text{add}}$ and $\mathcal{L}_{\text{sharp}}$ in our full objective (cf. Table. 5), from two perspectives. Fig. 8 shows the recovered 7-frame blur-to-video sequence and reveals the temporal-coherence axis, while Fig. 9 shows the center-frame deblurring output and reveals the

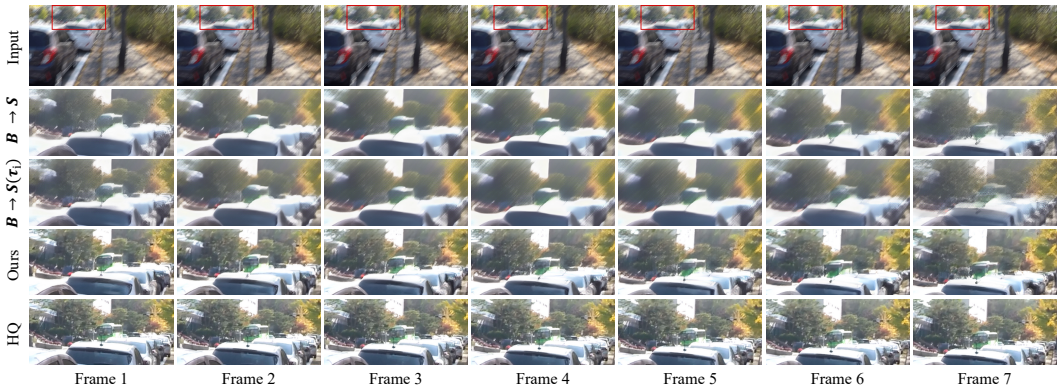


Figure 6: Qualitative comparison for the supervision-target ablation on GoPro-7. Best viewed zoomed in.



Figure 7: Additional blur-to-video comparisons on GoPro, trained on odd frames only.

per-frame sharpness axis. Adding $\mathcal{L}_{\text{add}}$ on top of $\mathcal{L}_{\text{recon}}$ enforces direction-consistent motion across frames but leaves the center frame blurry, since no loss yet anchors per-frame sharpness. Adding $\mathcal{L}_{\text{sharp}}$ instead yields a sharp center frame but with visible inter-frame jitter in the recovered sequence, since each frame is supervised independently with no cross-frame consistency. Only the full objective combining both losses yields outputs that are sharp *and* direction-consistent simultaneously, demonstrating that the two identity-derived losses address orthogonal failure modes.

E Continuous-Time Evidence

Denser timestamps on unseen data. GoPro provides ground truth only at its 7 to 13 subframe timestamps. We therefore use X4K1000FPS [41]: averaging 33 consecutive frames of the 1000-fps source synthesizes the blur and leaves ground truth at 33 timestamps per exposure, over the 15-clip X-TEST split (495 frames per model). Neither model has seen this data; both are applied zero-shot from their GoPro-trained weights, using the released Blur2Vid checkpoint. PickMoment outperforms Blur2Vid at every one of the 33 timestamps (Table 10).

Table 10: **Zero-shot PSNR on X4K1000FPS (X-TEST).** Nine of the 33 timestamps per exposure are shown; the last column is the mean over all 33.

t/T	0.000	0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000	33-pt mean
Blur2Vid [43]	21.31	20.30	19.99	22.92	21.46	22.47	21.95	21.53	21.30	21.60
Ours	**22.09**	**22.39**	**23.29**	**23.65**	**24.07**	**23.61**	**22.89**	**21.82**	**21.60**	**22.95**

Temporal super-resolution. If the model represents a continuous interval-mean field, two queries that name the same physical instant should return the same image even when the input exposure differs. During training the exposure length of the input blur never changes; only the query interval does. On the GoPro test set, we synthesize half-exposure blurs by averaging the first half of the subframes. Route A queries the mid-exposure of the full blur, which is standard center-frame deblurring under the protocol of Table 1; Route B queries the endpoint of the half blur. Both name the same instant, but the model sees an input length it was never trained on in Route B.

Route B stays within 1 dB of Route A, and the two routes agree more closely (27.91 dB) than independent errors at their measured accuracies would predict (25.9 dB). This agreement could also arise from shared model

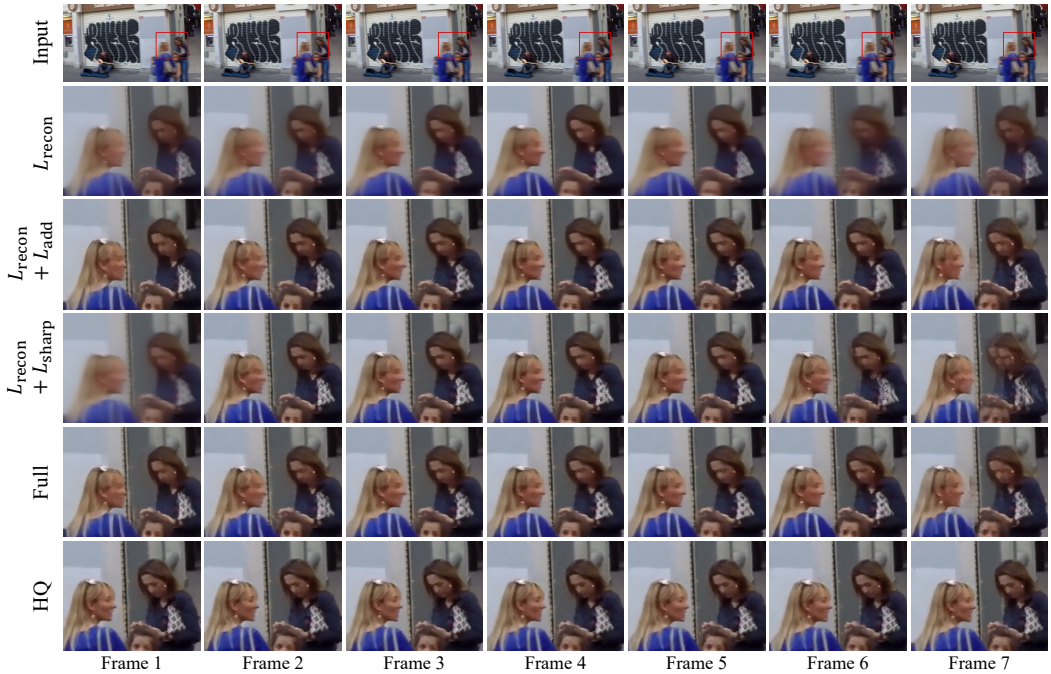


Figure 8: Qualitative comparison for the loss-term ablation on GoPro-7 (sequence view). Each row shows a recovered 7-frame sequence under a different combination of training losses. Best viewed zoomed in.

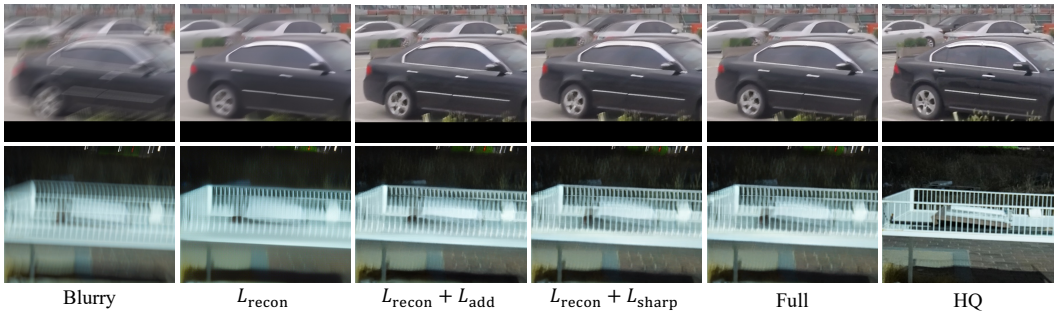


Figure 9: Qualitative comparison for the loss-term ablation on GoPro (center-frame deblurring view). Each column shows the recovered center sharp frame under a different combination of training losses. Best viewed zoomed in.

biases, the shared image prior, or the same scene structure, so we treat it as supporting evidence for a continuous interval-mean field rather than proof.

F Temporal Consistency and Direction

Warping-based temporal consistency. To complement FVD and EPE, we warp each recovered frame onto the next along RAFT [44] flow and measure the photometric residual, on the same GoPro-7 clips as Table 2. The ground-truth sequences give the residual of natural motion; a method closer to it neither flickers (rougher than GT) nor over-smooths (smoother than GT).

Ours is close to the ground-truth sequences, so there is no measurable flicker, while Blur2Vid is smoother than the ground truth by a wide margin, consistent with a prior that samples the whole sequence jointly rather than committing per interval. The per-transition profile for ours is mildly U-shaped (0.0080 at mid-exposure, 0.0094 and 0.0099 at the two ends), while that of the ground truth is flat.

Table 11: **Temporal super-resolution on GoPro.** PSNR of two routes that name the same instant, and their mutual agreement.

	PSNR
Route A (full blur, queried at mid-exposure) vs. GT	29.43
Route B (half blur, queried at its endpoint) vs. GT	28.49
Agreement (Route A vs. Route B)	27.91

Table 12: **Warping-based temporal consistency on GoPro-7.** Photometric residual after warping each frame onto the next with RAFT flow.

Sequence	warp L1	warp PSNR	vs. GT
GT (natural motion)	0.0096	35.23	–
Jin et al. [16]	0.0106	34.52	rougher than GT
Blur2Vid [43]	0.0072	37.77	smoother than GT
Ours	0.0089	36.49	closest to GT

Motion direction. Being deterministic, PickMoment returns the same sequence for a given input, so genuine ambiguity reduces to whether it picks one direction or blends the two. We score each recovered sequence against the forward and the reversed ground truth. Across all 1,744 GoPro-7 clips, the median PSNR gap between the two scores is 9.2 dB, and 93% of clips show a gap above 1 dB, so the model commits to one direction rather than averaging both. The temporal order of the subframes fixes the direction during supervision, and $\mathcal{L}_{\text{add}}$ ties the sub-interval predictions to one consistent trajectory. We report this as an empirical characterization, not a guarantee.

G Inference Cost

We measure inference wall-clock time on a single NVIDIA RTX A6000 GPU at 1280×640 resolution. PickMoment generates a 7-frame sequence in $\sim$8 seconds (mean 7.90 s over 10 runs after 3 warmup runs, covering all 7 forward passes), while Blur2Vid [43] requires $\sim$3.5 minutes per video at the default 50 DDPM steps, yielding roughly a $25\times$ inference speedup for PickMoment. This follows the one-step use of generative priors in recent restoration models [23, 39, 40], which likewise replace iterative sampling with a single network evaluation per output.

H Additional Visual Results

We provide additional qualitative comparisons that complement the main paper's figures. Where applicable, zoomed-in crops are included to highlight texture preservation and motion-trajectory coherence across frames.

Single-image deblurring. Fig. 10 shows additional single-image deblurring results on GoPro, HIDE, and RealBlur, comparing PickMoment against the strongest baseline from Table. 1. PickMoment recovers sharper edges and finer texture details than the baseline across all three datasets, including challenging real-blur cases on RealBlur where the blur kernel deviates from the synthetic GoPro distribution.

Blur-to-video. Fig. 11 shows additional blur-to-video sequences on the GoPro-7 split, where each row presents the input blurry image, the recovered 7-frame sequence by Blur2Vid [43], our recovery, and the ground-truth subframes. PickMoment produces sharper per-frame content with a deterministic, direction-consistent motion trajectory across the recovered sequence.

I Limitations

We discuss three principal limitations of PickMoment. **(i) Single-source supervision.** The model is trained exclusively on GoPro_Large_All without auxiliary or domain-augmented data. While it generalizes to HIDE and RealBlur, on RealBlur-J it is competitive but does not consistently outperform FideDiff [23], suggesting headroom from broader training distributions. **(ii) Temporal smoothness vs. per-frame fidelity trade-off.** PickMoment predicts each interval independently with one forward pass per interval. While this maximizes per-frame quality (PSNR_p, LPIPS_p in Table 2), the video-diffusion-based Blur2Vid retains a small advantage on cross-frame smoothness metrics (FVD, EPE), reflecting the implicit temporal regularization of its iterative

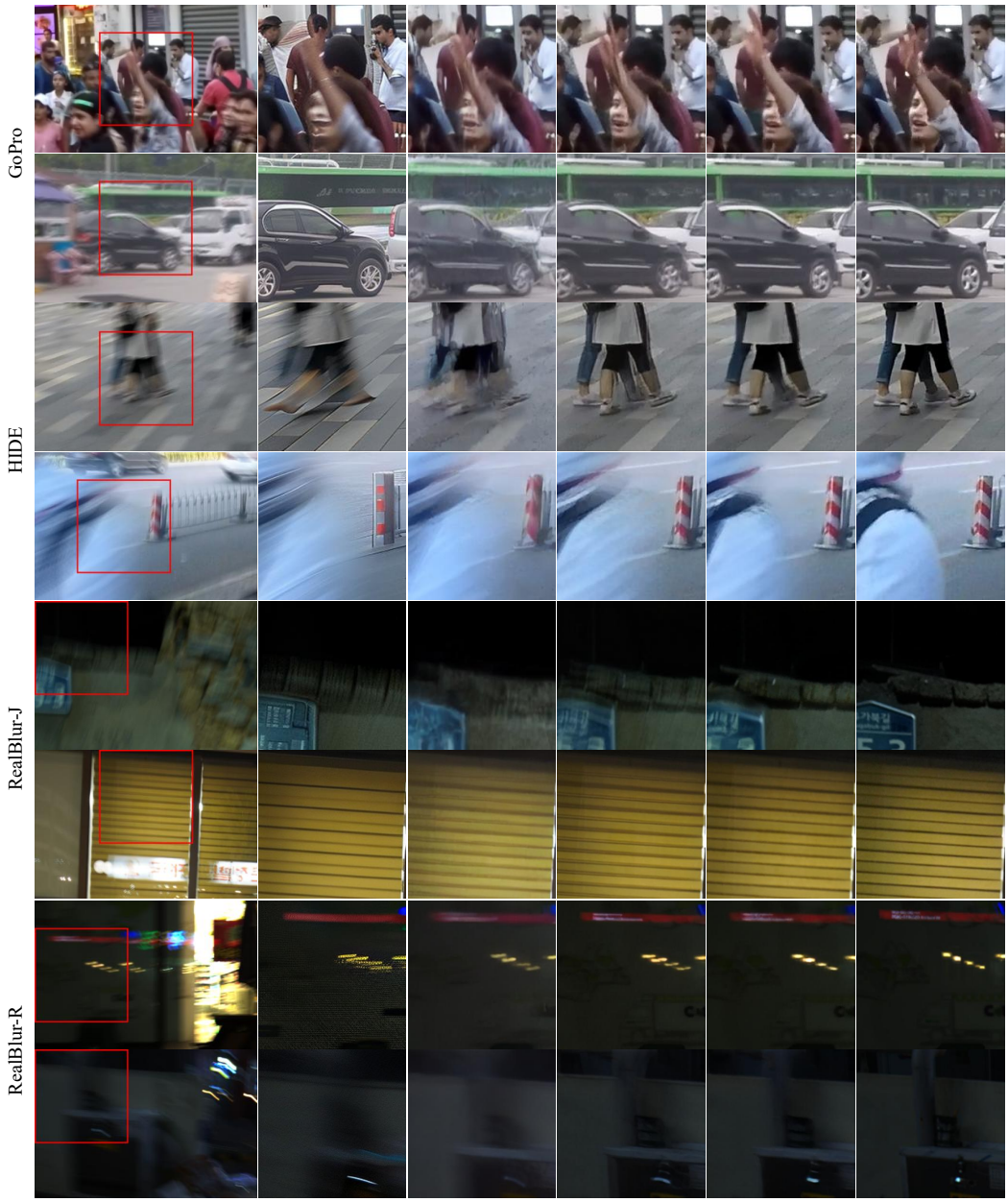


Figure 10: Additional single-image deblurring comparisons. Two rows per dataset, GoPro, HIDE, and RealBlur-J, RealBlur-R (top to bottom).

sampling. **(iii) Blur formation assumption.** Eq. 1 assumes the blur is well modeled as a uniform temporal integral of sharp frames over a known exposure window, which fits the GoPro synthesis protocol but does not directly cover non-uniform exposures, severe rotational or radial blur, or camera-pipeline effects (e.g., rolling shutter, gamma). Generalization to such regimes is left for future work. Explicitly modeling the degradation process has proven effective beyond deblurring, e.g., for dehazing [15, 39]. A natural extension is a weighted interval mean $B_w(s,t) = \int_s^t w(\tau)S(\tau)\,\mathrm{d}\tau \,/\, \int_s^t w(\tau)\,\mathrm{d}\tau$ with a calibrated or learned exposure profile w, for which the additivity identity still holds with ω replaced by the ratio of the weight integrals.

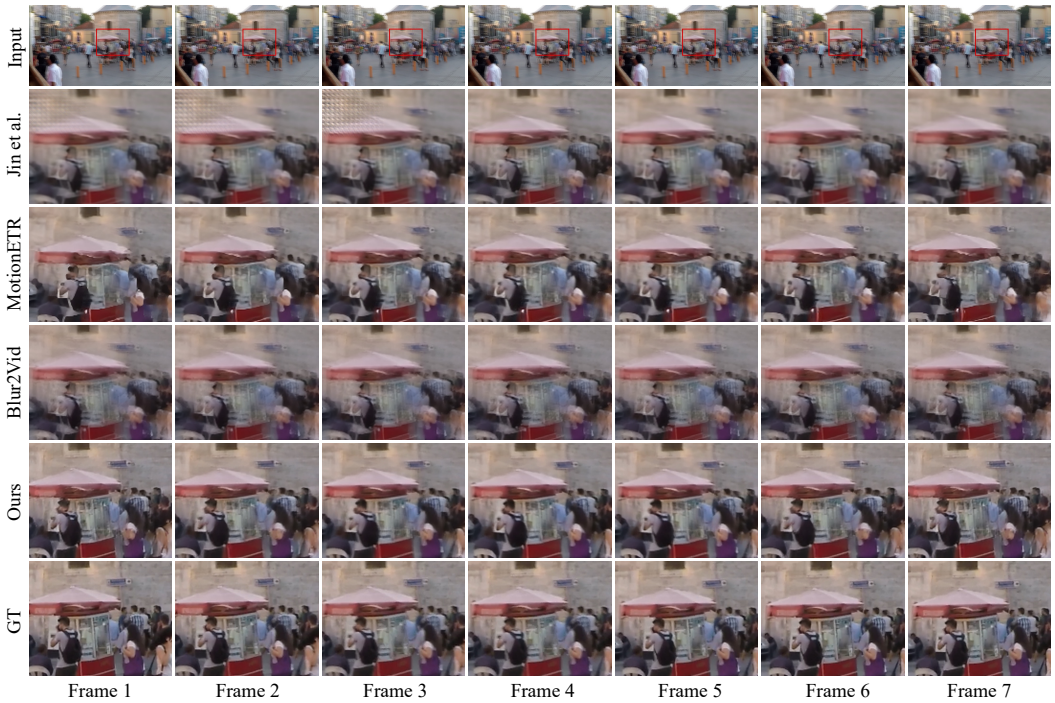


Figure 11: Additional blur-to-video comparisons on GoPro-7.

J Asset Licenses

Table 13 lists the datasets, pretrained models, and software used in this work, along with their licenses. We use each asset in accordance with its license; all assets are properly cited in the main paper.

Table 13: **Existing assets used in this work.** Datasets, pretrained models, and software with their licenses.

Asset	Type	Citation	License
GoPro_Large_All	Dataset	[30]	Research use (per authors)
HIDE	Dataset	[37]	Research use (per authors)
RealBlur-J / RealBlur-R	Dataset	[34]	CC BY-NC-SA 4.0
FLUX.2-klein	Pretrained model	[3]	FLUX.2 Community License (non-commercial)
Stable Diffusion (referenced)	Pretrained model	[35]	CreativeML Open RAIL-M
RAFT	Optical flow	[44]	BSD 3-Clause
LPIPS	Metric	[53]	BSD 2-Clause
DISTS	Metric	[7]	MIT
EA-LPIPS	Metric	[23]	MIT (per authors)
PyTorch	Framework	—	BSD 3-Clause

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: The main claims (state-of-the-art generative-prior-based single-image deblurring on GoPro and HIDE, competitive results on RealBlur, blur-to-video sequence recovery with one forward pass per output frame, and continuous-time generalization) are stated in the abstract and Section 1, and validated in Sections 4.2–4.3 (Tables 1, 2) and Section 4.4 (Tables 3–5).

   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: A dedicated limitations discussion is provided in Appendix I, covering single-source training (GoPro only), the trade-off between per-frame fidelity and temporal smoothness against video-diffusion baselines, and the assumptions of the temporal-integration blur model in Eq. 1, including a possible extension to non-uniform exposure profiles.

   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: The paper does not include formal theoretical results requiring assumptions or proofs; the identity in Eq. 7 and the zero-interval limit in Eq. 9 follow directly from the definition of the interval mean in Eq. 6.

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: Section 4.1 specifies datasets, optimizer, learning-rate schedule, progressive-patch schedule, mixed-precision configuration, and identity-loss warmup; full hyperparameters are reported in Appendix A, and code and checkpoints are publicly released to enable end-to-end reproduction.

   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: Code and the trained checkpoint are released (CC BY-NC 4.0) at `https://github.com/junsung6140/PickMoment` with instructions to reproduce the main results; all training and evaluation datasets used (GoPro, HIDE, RealBlur) are publicly available, with licenses listed in Appendix J.

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: Section 4.1 reports data splits, optimizer (Adam), learning-rate and warmup/cosine schedule, the progressive-patch schedule with stage transitions, mixed-precision settings, and identity-loss warmup; full hyperparameter values are given in Appendix A.

   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: Following the standard practice in single-image deblurring [51, 29, 23] and blur-to-video [16, 43] literature, we report point estimates from a single training run; the per-image evaluation set sizes (GoPro: 1,111; HIDE: 2,025; RealBlur-J/R: 980 each) provide a large effective sample for averaging, and the diffusion-scale training cost makes multi-seed reporting impractical within a single submission.

   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: Section 4.1 and Appendix A report the GPU type and number of devices, total training iterations, mixed-precision configuration, and the wall-clock time required to obtain the reported results.

   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 research conforms to the NeurIPS Code of Ethics; we use only publicly available motion-blur datasets (GoPro, HIDE, RealBlur), do not collect new human-subject data, and the work poses no foreseeable misuse risks beyond those already present in motion-deblurring research.

   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: [Yes]

    Justification: On the positive side, PickMoment improves the recoverability of image content from motion blur, which benefits downstream applications such as photography, surveillance video review, and 3D reconstruction. On the negative side, the blur-to-video capability is adjacent to synthetic motion generation and could in principle be misused for deceptive content; however, the model is conditioned on an observed blurry image and recovers content consistent with that observation rather than generating arbitrary novel motion, which limits its standalone misuse potential.

    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: PickMoment is a restoration model that recovers sharp content from an observed blurry input rather than synthesizing arbitrary novel imagery; it does not pose high-risk dual-use or generative-misuse concerns, so no special safeguards beyond a standard research-use release license are required.

    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 datasets (GoPro, HIDE, RealBlur), pretrained backbones (FLUX.2-klein), and software/metric implementations (RAFT, LPIPS, DISTS, EA-LPIPS, PyTorch) used in this work are properly cited in the main text; their versions and licenses are listed in Appendix J, and we use each asset in accordance with its license.

    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: [Yes]

    Justification: Code and the trained checkpoint are released at `https://github.com/junsung6140/PickMoment` with a README covering usage, training, evaluation, and license (CC BY-NC 4.0).

    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: The paper does not involve crowdsourcing or research with human subjects; all evaluation uses publicly available image datasets.

    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: The paper does not involve research with human subjects, so IRB approval is not applicable.

    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: LLMs are not part of the core methodology of this work; any use of LLMs was limited to writing and editing assistance and does not affect the scientific content of the paper.

    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.