# INFINITE-RESOLUTION INTEGRAL NOISE WARPING FOR DIFFUSION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Adapting pretrained image-based diffusion models to generate temporally consistent videos has become an impactful generative modeling research direction. Training-free noise-space manipulation has proven to be an effective technique, where the challenge is to preserve the Gaussian white noise distribution while adding in temporal consistency. Recently, Chang et al. (2024) formulated this problem using an integral noise representation with distribution-preserving guarantees, and proposed an upsampling-based algorithm to compute it. However, while their mathematical formulation is advantageous, the algorithm incurs a high computational cost. Through analyzing the limiting-case behavior of their algorithm as the upsampling resolution goes to infinity, we develop an alternative algorithm that, by gathering increments of multiple Brownian bridges, achieves their infinite-resolution accuracy while simultaneously reducing the computational cost by orders of magnitude. We prove and experimentally validate our theoretical claims, and demonstrate our method's effectiveness in real-world applications. We further show that our method can readily extend to the 3-dimensional space.

## 1 INTRODUCTION

The success of diffusion models for image generation and manipulation (Rombach et al., 2022; Nichol et al., 2021; Ho et al., 2020; Zhang et al., 2023a) has spurred significant interest in lifting these capacities to the video domain (Singer et al., 2022; Durrett, 2019; Gupta et al., 2023; Blattmann et al., 2023; Ho et al., 2022; Guo et al., 2024). While building video models trained directly on spatiotemporal data is a natural idea, practical concerns such as limited availability of large-scale video data and high computational cost have motivated investigations into training-free alternatives. One such training-free approach is to use image models to directly generate video frames, utilizing various techniques such as cross-frame attention, feature injection and hierarchical sampling to promote cross-frame temporal consistency (Zhang et al., 2023b; Khachatryan et al., 2023; Cong et al., 2023).

One particularly effective consistency-promoting technique is the temporally consistent initialization of noise across frames. However, most existing approaches either result in a loss of Gaussianity in the noise image (and subsequently introduces a domain gap at inference time), or restrict themselves to simple manipulations of the noise image (*e.g.* filtering and blending). Recently, Chang et al. (2024) proposed a method that preserves both Gaussian white noise distribution and cross-frame temporal correlations via *integral noise warping*: each warped noise pixel is obtained by integrating a continuous noise field, where the integration is implemented by summing over a polygonal deformed pixel region in the upsampled prior noise image. However, the time and memory costs of this algorithm grows quadratically with the upsampling resolution, prohibiting the adoption of the method in certain applications (Kwak et al., 2024) due to this computational expense.

Regarding this challenge, our key insight is that when adopting an Eulerian perspective as opposed to the original Lagrangian perspective, the limiting-case algorithm of Chang et al. (2024) for computing a warped noise pixel reduces to summing over increments from multiple Brownian bridges (Durrett, 2019, Section 8.4). In place of the costly upsampling procedure, sampling the increments of a Brownian bridge can be done efficiently in an autoregressive manner (2). We thus propose *infinite-resolution integral noise warping* (Algorithm 1), which can directly and efficiently resolve noise transport in the continuous space, when supplied with an oracle that returns the overlapping area between a pixel square and a deformed pixel region (Section 2.3).

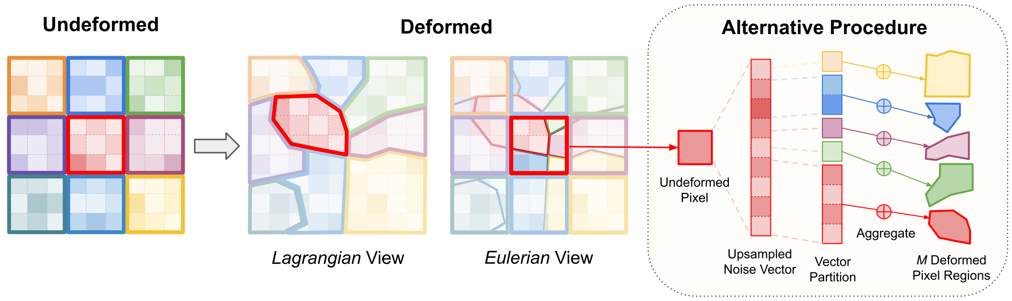


Figure 1: When the image grid deforms, the Lagrangian view tracks a deformed pixel region, while the Eulerian view tracks the undeformed pixel square as it gets partitioned into multiple regions. On the right, we leverage the exchangeability of upsampled subpixels to convert the Lagrangian gathering procedure into scattering noise subpixels to overlapped deformed pixel regions.

We propose two concrete methods for computing this oracle, leading to a *grid-based* and a *particle-based* variant. Following Chang et al. (2024), the *grid-based* variant (Algorithm 2) computes the area by explicitly constructing per-pixel deformed polygons, and is exactly equivalent to the existing approach (Chang et al., 2024) with an infinite upsampling resolution, while running $8.0\times$ to $19.7\times$ faster and using $9.22\times$ less memory. Inspired by hybrid Eulerian-Lagrangian fluid simulation (Brackbill et al., 1988), our novel *particle-based* variant (Algorithm 3) computes area in a fuzzy manner, which not only offers a further $5.21\times$ speed-up *over our grid-based variant*, but is also agnostic to non-injective maps. In real-world scenarios, the particle-based variant shows no compromise in generation quality compared to the grid-based one (see video results), while offering superior robustness, efficiency, simplicity, and extensibility to higher dimensions.

In summary, we propose a novel method for computing temporally correlated noise to facilitate consistent video generation with image-based diffusion models. Our algorithm computes the integral noise from Chang et al. (2024) at infinite resolution, warping a $1024 \times 1024$ noise image in $\sim 0.045$s (grid variant) and $\sim 0.0086$s (particle variant), achieving orders of magnitude speed-up compared to Chang et al. (2024) while retaining the distribution-preserving and temporally-coherent properties.

## 2 METHODOLOGY

In this section, we introduce our method as follows:

- We present an equivalent Eulerian interpretation (Figure 1) for the method by Chang et al. (2024), which was developed from a Lagrangian viewpoint.
- We show that the limiting algorithm of the Eulerian formulation as upsampling level goes to infinity is equivalent to sampling increments of Brownian bridges.
- We present our main algorithm (Algorithm 1) which, given a partition record that returns the overlapping area between a pixel square and a deformed pixel region, samples increments of Brownian bridges and scatters the increments to form the warped noise image.
- We propose two practical algorithms for computing the overlap areas. The *grid-based* Algorithm 2 extends Chang et al. (2024) to infinite resolution without the overhead of upsampling. The *particle-based* Algorithm 3 departs from grid-based discretization and uses particles instead, resulting in a simpler algorithm that is robust to degenerate maps.

### 2.1 NOISE WARPING: AN ALTERNATIVE EULERIAN PERSPECTIVE

Given a $D \times D$ prior noise image $I_W \in \mathbb{R}^{D\times D}$[1] and a deformation map $\psi : [0,1]^2 \to [0,1]^2$, the noise-warping algorithm (Chang et al., 2024) computes the warped noise image $\widetilde{I}_W \in \mathbb{R}^{D\times D}$ with upsampling level $N \in \mathbb{Z}_{\geq 1}$ as follows:

---

[1] In practice, the noise image has multiple channels. Since channels are always treated independent of one another, to simplify the notation, we will assume the image has a single channel.

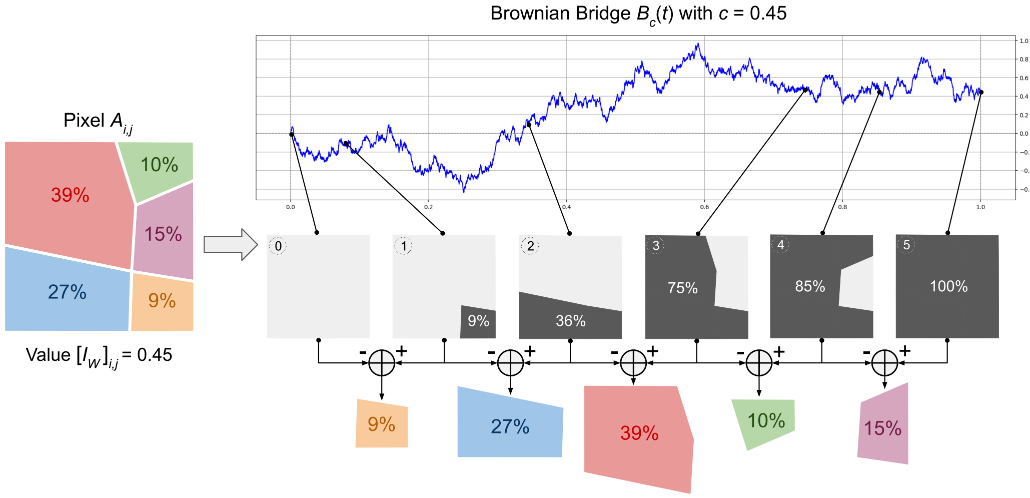

Brownian Bridge $B_c(t)$ with $c$ = 0.45
Pixel $A_{i,j}$
39%
10%
15%
27%
9%
Value $[I_W]_{i,j}$ = 0.45
0
1
2
3
4
5
9%
36%
75%
85%
100%
9%
27%
39%
10%
15%

Figure 2: Connection between Eulerian noise-warping and increments of a Brownian bridge for a fixed prior noise pixel $[I_W]_{i,j}$. The overlapping area of each colored warped region becomes the time increment for the Brownian bridge. Hence, sampling the Brownian bridge at these times and taking consecutive differences yields integral noise that is scattered to form each warped noise pixel.

1. For $i, j = 1, \dots, D$, upsample noise pixel $[I_W]_{i,j}$ to an $N \times N$ subimage $[\widehat{I}_W]_{i,j} \in \mathbb{R}^{N\times N}$:

$$[\widehat{I}_W]_{i,j} = \frac{[I_W]_{i,j}}{N^2} + \frac{1}{N}\left(Z - \frac{S}{N^2}\right), \text{ with } Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ and } S = \sum\nolimits_{k=1}^{N^2} Z_k. \tag{1}$$

   The subimage for each pixel assembles into an $ND \times ND$ upsampled noise image $\widehat{I}_W$.

2. For $i, j = 1, \dots, D$, the pixel square $A_{i,j} := [\frac{i-1}{D}, \frac{i}{D}] \times [\frac{j-1}{D}, \frac{j}{D}]$ is warped to a deformed pixel region $\widetilde{A}_{i,j} := \psi(A_{i,j})$, and the warped noise pixel $[\widetilde{I}_W]_{i,j}$ is set to be the sum of all subpixels in $\widehat{I}_W$ covered by $\widetilde{A}_{i,j}$ divided by $\sqrt{|\widetilde{A}_{i,j}|}$, where $|A|$ denotes the Lebesgue measure of a Borel set $A \subset \mathbb{R}^2$.

We describe an alternative but equivalent procedure by making the following two observations, which are illustrated in Figure 1.

**Gathering Noise → Scattering Noise.** While the original procedure computes the warped noise image by *gathering* the upsampled noise subpixels in each deformed pixel region $\widetilde{A}_{i,j}$ in a *Lagrangian* fashion, we can instead use an alternative procedure by *scattering* the upsampled noise subpixels in each pixel square $A_{i,j}$ to overlapping deformed pixel regions. This new *Eulerian* procedure does not change the output, but it yields new insights in conjunction with our second observation.

**Scattering Noise → Counting Overlapping Subpixels.** Observe that the $N \times N$ subpixels in $[\widehat{I}_W]_{i,j}$, for every $i, j$, are correlated only through their sum $S$ when conditioning on $[I_W]_{i,j}$ (1), so they are exchangeable. Hence, when scattering these upsampled noise subpixels to deformed pixel regions, the order of scattering does not matter, and we only need to count *the number of subpixels* covered by each deformed pixel region.

**Alternative Eulerian Procedure.** Putting both observations together, we now describe an alternative procedure to Chang et al. (2024) with unaltered output:

1. For each noise image pixel $[I_W]_{i,j}$, draw an upsampled subimage, now represented as a 1D vector $X \in \mathbb{R}^{N^2}$ using (1). Then, compute a prefix sum $H_{i,j}$ via $[H_{i,j}]_k := \sum_{q=1}^{k} X_q$ for $k = 1, \dots, N^2$.

2. Warp each pixel square and compute deformed pixel regions $\widetilde{A}_{i,j}$ as before.

3. For each $A_{i,j}$, identify all $M$ deformed pixel regions $\{\widetilde{A}_{\ell_k,m_k}\}_{k=1,\dots,M}$ that overlap with $A_{i,j}$. Form $L \in \mathbb{Z}^M_{\geq 0}$ where $L_k$ represents the number of upsampled subpixels covered by the $k^{\text{th}}$ overlap. Then, compute a prefix sum $[C_{i,j}]_k \coloneqq \sum_{q=1}^{k} L_q$. For $k = 1, \dots, M$, accrue $[H_{i,j}]_{[C_{i,j}]_k} - [H_{i,j}]_{[C_{i,j}]_{k-1}}$ to $[\widetilde{I}_W]_{\ell_k,m_k}$, the $k^{\text{th}}$ overlapped warped noise pixel.
4. Divide each warped noise pixel $[\widetilde{I}_W]_{i,j}$ by $\sqrt{|\widetilde{A}_{i,j}|}$.

**Discussion.** Compared to the original procedure by Chang et al. (2024), this alternative but equivalent algorithm highlights how the upsampled subpixels of $[I_W]_{i,j}$ are scattered to form the warped noise pixels. In particular, each warped noise pixel receives the *sum of a continuous segment* in $H_{i,j}$. Since $H_{i,j}$ is a summation of weakly correlated and exchangeable subpixels, once conditioned on $[I_W]_{i,j}$, *can we avoid explicitly instantiating every single subpixel*, but instead model the *sum* of these weakly correlated subpixels?

The key insight of this paper is that when the upsampling resolution $N \to \infty$, the scaling limit of the prefix sum $H_{i,j}$ (with proper interpolation and time scaling to a continuous function) is precisely the Brownian bridge (Durrett, 2019, Section 8.4) conditioned on $[I_W]_{i,j}$. Once this connection is established, it is easy to progressively sample increments of the Brownian bridge, resulting in a clean and efficient noise-warping algorithm that bypasses the need for upsampling in Chang et al. (2024).

### 2.2 INFINITE-RESOLUTION NOISE SCATTERING

In this section, we first derive a scaling limit result to Brownian bridges. We then illustrate that the limiting version of the Eulerian procedure from the previous section matches precisely this scaling limit result. Lastly, we demonstrate an autoregressive way to sample increments of a Brownian bridge that is linear in runtime in terms of the number of increments.

**Theorem 1** (Scaling limit to Brownian bridge)**.** Let $\{Z_n\}$ be a sequence of i.i.d. random variables with finite variance that are normalized such that $\mathbb{E}[Z_n] = 0$ and $\mathrm{Var}(Z_n) = 1$. For $c \in \mathbb{R}$, define

$$S_n \coloneqq \sum_{i=1}^{n} Z_i, \qquad X_{i,n} \coloneqq \frac{c}{n} + \frac{1}{\sqrt{n}}\left(Z_i - \frac{S_n}{n}\right).$$

Consider the sequence of random continuous functions $\{H_n(t)\} \subset C[0,1]$ defined as

$$H_n(t) \coloneqq \sum_{i=1}^{\lfloor nt \rfloor} X_{i,n} + (nt - \lfloor nt \rfloor) X_{\lfloor nt \rfloor + 1, n}.$$

Then the sequence $\{H_n\}$ converges in distribution under the sup-norm metric on $C[0,1]$ to $B_c(t) \coloneqq W(t) - tW(1) + tc$, the Brownian bridge ending at $c$, where $W(t)$ is standard Brownian motion. Moreover, in distribution, we have $B_c(t) \overset{d}{=} (W(t) \mid W(1) = c)$, where $(W(t) \mid W(1) = c)$ is the disintegrated measure (Pachl, 1978) of $W(t)$ on $W(1) = c$.

We prove Theorem 1 in Appendix A. To connect the Eulerian procedure with the setup in Theorem 1, let us fix a pixel $[I_W]_{i,j}$, and let $B \coloneqq B_{[I_W]_{i,j}}$, $H \coloneqq H_{i,j}$, $C \coloneqq C_{i,j}$ to simplify the notation. By setting $n = N^2$ and $c = [I_W]_{i,j}$, the sequence $\{X_{k,n}\}$ from the theorem has exactly the same law as the upsampled subpixels in $[\widehat{I}_W]_{i,j}$. Moreover, $H_{nt} = H_n(t)$ when $nt \in \mathbb{Z}_{\geq 1}$. By taking $N \to \infty$, implying $n \to \infty$, for any $t_1, \dots, t_M \in [0,1]$, we have the convergence in distribution of $\left(H_{\lfloor nt_1 \rfloor}, \dots, H_{\lfloor nt_M \rfloor}\right) \xrightarrow{d} (B(t_1), \dots, B(t_M))$. Recall in the Eulerian procedure, we only need to access the prefix sum $H$ at indices $\{C_k\}_{k=1}^{M}$, where $C_k$ counts the number of upsampled subpixels covered by the first $k$ overlaps. This suggests that if we choose

$$t_k = \lim_{N \to \infty} \frac{C_k}{N^2} = \sum_{k'=1}^{k} \left| A_{i,j} \cap \widetilde{A}_{\ell_{k'},m_{k'}} \right|,$$

and use $B(t_k)$ in place of $H_k$, then we just need to sample from $B$ at times $t_1, \dots, t_M$ — precisely the limiting algorithm of the Eulerian procedure. We illustrate this connection in Figure 2.

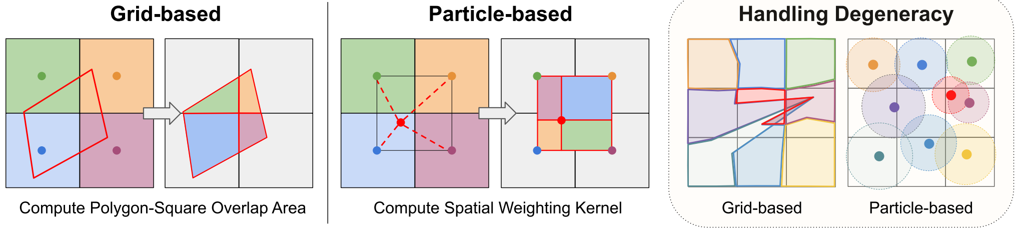

Grid-based
Particle-based
Handling Degeneracy
Compute Polygon-Square Overlap Area
Compute Spatial Weighting Kernel
Grid-based
Particle-based

Figure 3: In the grid-based variant, we compute the overlapping area by explicitly constructing the polygonal region of the deformed pixel region. In the particle-based variant, we approximate the overlapping area with a linear kernel-based interpolation. On the right, we show how the two variants differ when the map is non-injective: the grid-based variant yields intersecting regions, causing stability issues, whereas the particle-based variant remains stable.

**Autoregressive Sampling of Brownian Bridges.** Since a Brownian bridge is a Markov process (Oksendal, 2013, Exercise 5.11), we can sample the vector $(B_c(t_1), \dots, B_c(t_M))$ in an autoregressive fashion, each time sampling $B_c(t_{k+1})$ conditioned on $B_c(t_k)$:

$$(B_c(t_{k+1}) \mid B_c(t_k) = q) \overset{d}{=} \mathcal{N}\left(\frac{1-t_{k+1}}{1-t_k}q + \frac{t_{k+1}-t_k}{1-t_k}c, \frac{(t_{k+1}-t_k)(1-t_{k+1})}{1-t_k}\right). \tag{2}$$

Once the Brownian bridge at times $t_k$ is sampled, we just need to accrue the increments $B_c(t_k) - B_c(t_{k-1})$ to $[\widetilde{I}_W]_{\ell_k, m_k}$, the $k^{\text{th}}$ overlapped warped noise pixel. This allows us to present Algorithm 1. Compared to the discrete procedures described earlier, we no longer need upsampling. In addition, we exploited the autoregressive nature of Brownian bridges to bring down the time complexity to linear in the number of overlapping warped pixel regions.

**Algorithm 1** Infinite-Resolution Integral Noise Warp

```
Input: prior noise image $I_W \in \mathbb{R}^{D\times D}$, deformation map $\psi : [0,1] \to [0,1]$
Output: warped noise image $\widetilde{I}_W \in \mathbb{R}^{D\times D}$
  Build a partition record $\mathcal{P}$ from $\psi$ (Section 2.3)
  Initialize $\mathcal{A}_{i,j} \leftarrow 0$ for all $i, j = 1, \dots, D$      ▷ $\mathcal{A}_{i,j}$ will eventually be the area of $\widetilde{A}_{i,j}$
  parallel for each $u, v = 1, \dots, D$ do
      $t, q, M \leftarrow 0, 0, |\mathcal{P}_{u,v}|$
      for $k = 1, \dots, M$ do
          $(a, i, j) \leftarrow [\mathcal{P}_{u,v}]_k$      ▷ $a$ is the overlapping area between $A_{i,j}$ and $\widetilde{A}_{u,v}$
          Sample $q' \sim (B_c(t+a) | B_c(t) = q)$ by (2) with $c = [I_W]_{u,v}$
          $[\widetilde{I}_W]_{i,j} \leftarrow [\widetilde{I}_W]_{i,j} + (q' - q)$
          $\mathcal{A}_{i,j} \leftarrow \mathcal{A}_{i,j} + a$
          $q, t \leftarrow q', t + a$
  Normalize $[\widetilde{I}_W]_{i,j} \leftarrow \mathcal{A}_{i,j}^{-\frac{1}{2}} [\widetilde{I}_W]_{i,j}$ for all $i, j = 1, \dots, D$
  return $\widetilde{I}_W$
```

**Preservation of Gaussian White Noise.** A central desideratum of noise warping is that the resulting warped noise image $\widetilde{I}_W$ needs to have pixels that are i.i.d. standard Gaussians when the prior noise image $I_W$ is Gaussian white noise. This ensures that the warped noise is in-distribution for a pre-trained diffusion model. Our algorithm automatically guarantees this preservation of Gaussianity, as long as the warping function $\psi$ is injective. To see this, the injectivity of $\psi$ implies that the warped pixel regions are non-overlapping in the square $[0,1]^2$. For each $A_{i,j}$, since $[I_W]_{i,j} \overset{d}{=} \mathcal{N}(0,1) \overset{d}{=} W(1)$, by the conditional interpretation of Brownian bridges (1), when marginalizing out $[I_W]_{i,j}$, the Brownian bridge $B_{[I_W]_{i,j}}$ reduces to standard Brownian motion. Since the increments of the Brownian motion are independent Gaussians, the contribution to a deformed pixel region is simply a zero-mean Gaussian with variance equal to the overlapping area. Therefore, each deformed pixel region will receive the sum of a number of independent Gaussians whose variances sum to the area of the region. The scaling by the inverse square root of the area in Algorithm 1 thus makes each warped noise pixel an i.i.d. standard Gaussian.

### 2.3 BUILDING PARTITION RECORDS

In this section we present two algorithms, one grid-based and one particle-based, for building partition records that return the area between each pixel square and its overlapping deformed pixel regions. Both versions are illustrated in Figure 3.

**Algorithm 2** Grid-based Partition

```
Input: Deformation map $\psi$
Output: Partition record $\mathcal{P}$
  parallel for each $i, j$ do
      $A^* \leftarrow$ DiscretizeSquare($A_{i,j}$)
      $S \leftarrow \psi(A^*)$
      $u^-, u^+, v^-, v^+ \leftarrow$ AABB($S$)
      for $u \in [u^-, u^+]$ do
          for $v \in [v^-, v^+]$ do
              $a \leftarrow$ PolygonArea(Clip($S, u, v$))
              $\mathcal{P}_{u,v} \leftarrow \mathcal{P}_{u,v} + [(a, i, j)]$
  return $\mathcal{P}$
```

**Algorithm 3** Particle-based Partition

```
Input: Deformation map $\psi$
Output: Partition record $\mathcal{P}$
  parallel for each $i, j$ do
      $(x, y) \leftarrow \psi(\frac{i+0.5}{D}, \frac{j+0.5}{D})$
      $w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1} \leftarrow$ BilinearWeights($X$)
      for $s, t \in [0, 1]$ do
          $x', y' \leftarrow \lfloor x \rfloor + s, \lfloor y \rfloor + t$
          $\mathcal{P}_{x',y'} \leftarrow \mathcal{P}_{x',y'} + [(w_{s,t}, i, j)]$
  parallel for each $u, v$ do
      Normalize total area of $\mathcal{P}_{u,v}$ to $D^{-2}$
  return $\mathcal{P}$
```

Our grid-based method (Algorithm 2 and Figure 3, left) follows the same approach as Chang et al. (2024), treating each deformed pixel as an octagon and each undeformed pixel as a grid cell. Our particle-based method (Algorithm 3 and Figure 3, middle) borrows from the grid-to-particle (G2P) technique in fluid particle-in-cell method methods (Brackbill et al., 1988), where we treat each deformed pixel as a particle and each undeformed pixel as grid cell. Each particle requests area from nearby grid cells based on distance; upon receiving requests, each grid cell normalizes the requests to ensure partition-of-unity, and distributes its area to contacting particles.

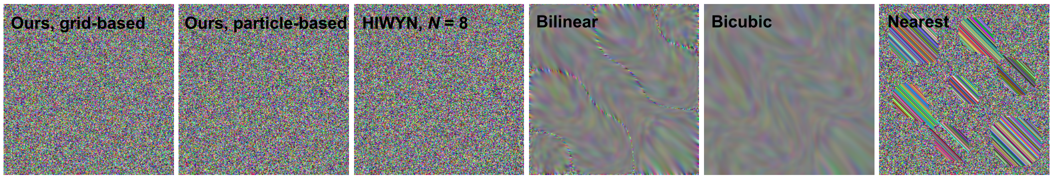
**Discussion.** Conceptually, our grid-based and particle-based methods correspond to two different interpretations of $\psi$ when it is available only as discrete samples (*e.g.* optical flow image). The grid-based method implicitly reconstructs the continuous $\psi$ field by linear interpolation, whereas the particle-based method makes no such interpolation and assumes $\psi$ is only known point-wise. This implies that when $\psi$ is smooth, linear interpolation works well and the grid-based method will yield a higher-quality warp as seen in Figure 10. But when $\psi$ is non-smooth, which is usually the case in real-world data, linear interpolation becomes problematic and can lead to degenerate polygons as illustrated on the right of Figure 3. These degenerate polygons can violate the assumption that deformed pixel regions do not overlap, which is required for independence in the warped noise. We note that, in practice, both Chang et al. (2024) and our grid-based method are equipped with fail-safes that ensures this independence by "patching up" degenerate regions with new Gaussian noise, but these mechanisms do not handle the overlaps in a principled manner, while the particle-based variant fundamentally circumvents such overlaps with its topology-free nature.

In addition, we also highlight the simplicity of the particle-based approach, as it boils down the partition record computation to evaluating bilinear kernels, which is highly efficient and parallelizable. Such simplicity offers a unique opportunity to extend the noise warping to higher dimensions, as it effectively only requires the change from the bilinear kernel to its higher-dimensional counterpart. We showcase this possibility using a 3-dimensioal noise warp example as shown in Figure B.2.

## 3 RESULTS

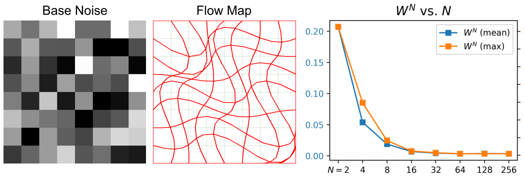
We verify our theoretical claims by showing that both variants of our method preserve Gaussian white noise distribution, and that Chang et al. (2024) (HIWYN) converges to our grid-based variant as $N$ increases. We analyze the behaviors of our grid-based and particle-based variants under diffeomorphic and non-diffeomorphic deformations. We then apply our method in video generation and benchmark against existing methods (Ge et al., 2023; Chen et al., 2023; Chang et al., 2024). Finally, we extend our method to warping volumetric noise and demonstrate a use case in 3D graphics.

**Gaussian White Noise Preservation.** In Figure 4, we iteratively warp a noise image by the same deformation map for 50 timesteps. We gauge the output noise's resemblance to Gaussian white noise by measuring normality using one-sample Kolmogorov-Smirnov (K-S) test and detecting spatial cor-



| Distribution Preservation Metrics | | | | | | |
|---|---|---|---|---|---|---|
| Method | Ours, grid-based | Ours, particle-based | HIWYN, $N = 8$ | Bilinear | Bicubic | Nearest Neighbor |
| Moran's $I$ | 5.103e-4 / 0.849 | -1.995e-3 / 0.475 | 3.215e-3 / 0.243 | 0.612 / 0 | 0.983 / 0 | 2.974e-2 / 6.103e-27 |
| K-S Test | 3.410e-3 / 0.430 | 3.023e-3 / 0.586 | 3.274e-3 / 0.482 | 0.366 / 0 | 0.422 / 0 | 9.806e-3 / 6.681e-06 |

Figure 4: Preservation of Gaussian white noise achieved by different warping methods. We report scores and p-values for both Moran's $I$ (spatial correlation) and K-S test (normality). We show that results from our method (both variants) and HIWYN are indistinguishable from white Gaussian noise, while generic warping methods lead to corrupted noise.



| $W^N$ for Varying Values of $N$ | | | | | |
|---|---|---|---|---|---|
| $N$ | mean $W^N$ | max $W^N$ | $N$ | mean $W^N$ | max $W^N$ |
| 2 | 2.072e-1 | 7.253e-1 | 32 | 4.320e-3 | 1.391e-2 |
| 4 | 5.394e-2 | 2.962e-1 | 64 | 3.236e-3 | 8.325e-3 |
| 8 | 1.881e-2 | 8.310e-2 | 128 | 3.616e-3 | 8.134e-3 |
| 16 | 6.792e-3 | 2.361e-2 | 256 | 3.387e-3 | 8.228e-3 |

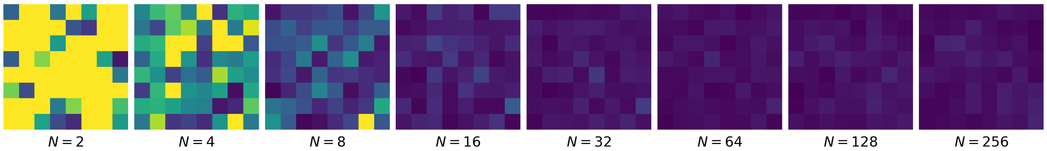


Figure 5: Convergence of HIWYN to our method as $N$ increases. Top left: experimental setup with prior noise and deformation map. Top middle: 2-Wasserstein distance $W^N$ between the output of HIWYN and ours. Top right: statistics table. Bottom: $W^N$ difference image between the output of HIWYN and ours as $N$ increases. Notice $W^N$ becomes statistically insignificant for $N \geq 64$.

relation using Moran's $I$. Our results show that both HIWYN and our method generate warped noise images indistinguishable from Gaussian white noise which significantly improve upon baselines.

**Convergence of Chang et al. (2024).** We validate that our method is the limiting case of HIWYN. Starting with an $8 \times 8$ prior noise image and a flow map (Figure 5, top left), we run our method along with HIWYN for $N \in \{2, 4, 8, \ldots, 256\}$ for 100,000 independent runs to estimate the distribution of the warped noise image. For each upsampling resolution $N$, we compute the 2-Wasserstein distance $W^N$ between the output of HIWYN and that of our method. The results in Figure 5 demonstrate the convergence of HIWYN to our method as $N$ increase, and reveal that $N = 8$ (recommended by Chang et al. (2024)) is not yet in the converged phase to yield a negligible $W^N$.

**Performance Comparison.** For our methods and HIWYN with upsampling levels $N \in \{2, 4, 8\}$, we perform 100 independent runs on a $1024 \times 1024$ image. We report the kernel time with CPU and GPU backends (Figure 7) as well as the memory usage. The runtime and memory usage of our methods are largely comparable with those of HIWYN with $N = 2$. Compared to HIWYN with $N = 8$, both our methods offer order-of-magnitude improvements in runtime and memory usage. Specifically, our grid-based method achieves infinite upsampling resolution while being $19.7\times$ faster on CPU and $8.0\times$ faster on GPU, using $9.22\times$ less memory, and our particle-based method, albeit not strictly equivalent to HIWYN at $N = \infty$, achieves a $41.7\times$ speedup on GPU. In the following sections, we show that the particle-based version consistently achieves comparable quality to the grid-based version in real-world scenarios.

**Comparison between Grid-Based and Particle-Based Variants.** In Figure 10, we compare both variants when the deformation map is diffeomorphic under different levels of distortion. Visually, the difference between the two variants is negligible at frame 25 and becomes noticeable in frame 100. We measure this difference by comparing the deformed regions for each pixel in terms of IoU

| Comparison of Time and Memory Costs | | | |
|---|---|---|---|
| Method | Time (CPU) | Time (GPU) | Memory |
| $N = 2$ (HIWYN) | 19.30s | 2.597s | 293.7MB |
| $N = 4$ (HIWYN) | 75.33s | 9.247s | 746.6MB |
| $N = 8$ (HIWYN) | 398.3s | 35.91s | 2147MB |
| $N = \infty$ (ours, grid) | 20.16s | 4.491s | 232.9MB |
| $N = \infty$ (ours, particle) | 15.68s | 0.862s | 320.9MB |

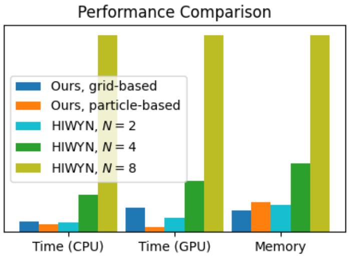


Figure 7: Runtime and memory usage of our method vs. HIWYN with $N = 2, 4, 8$. We compare total allocated memory and kernel time on a CPU/GPU. The computation is done on a laptop with Intel i7-12700H and GeForce RTX 3070 Ti.

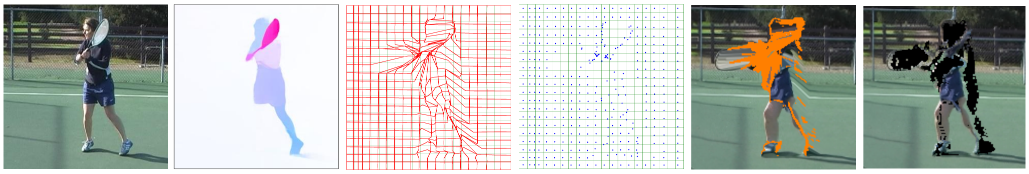

Video | Optical Flow | Grid-based | Particle-based | Area Occupation (Grid-based) | Area Occupation (Particle-based)

Figure 8: Comparison of grid-based vs. particle-based variants under non-diffeomorphic optical flow. Pixels with detected overlaps are colored in orange. Further results are given in Figure B.3.

and weighted Chamfer distance. We additionally compare the particle-based result with that of an identity-map baseline (right column in Figure 10), which shows that the gap between the two variants remains small even under large distortion. In Figure 8, we stress test both variants under non-diffeomorphic maps obtained using optical flow (Teed & Deng, 2020) on a real-world video (Brox & Malik, 2011). In images 3 and 4, we see that the real-world flow map induces inverted meshes for the grid-based variant and clustered particles for the particle-based variant. While clustered particles are always assigned disjoint regions due to the continuous nature of our algorithm, mesh inversions cause area contention issue due to overlaps. In images 5 and 6, we mark the grid cells with area contention in orange, which occurs in the grid-based version but not in the particle-based version.

**Conditional Video Generation.** We apply our method to conditional video generation by adapting SDEdit (Meng et al., 2021), a conditional image generation method, to producing consistent video frames. We apply Perturbed-Attention Guidance (Ahn et al., 2024) to the unconditional models with scale 3.0. Our two inputs are a conditioning video (generated by applying a median filter to real-world videos similar to Chen et al. (2023)) and an optical flow field (Teed & Deng, 2020). Without noise manipulation, if we run SDEdit frame-by-frame (Figure 9, bottom row), the details (e.g. in the tower and trees) would result in strong flickering. By warping the noise using the optical flow, the temporal consistency is much improved. As shown in Figure 9, our methods (both variants) and HIWYN yield comparable visual qualities. Full experiments that shows comparison with Control-A-Video (Chen et al., 2023) and PYoCo (Ge et al., 2023) and additional baselines are provided in Figures B.4 and B.5 with generation quality metrics reported in Figure B.1. We refer to our supplementary video for better visualization.

**3D Noise Warp.** We extend our particle-based algorithm to 3D by replacing the bilinear kernel with the bicubic kernel in Algorithm 3 and apply it to GaussianCube (Zhang et al., 2024), which denoises a dense 3D noise grid to reconstruct 3D Gaussians. We adapt it to perform conditional generation a la SDEdit. Starting with a 3D pickup truck generated unconditionally, we condition the model to generate vehicles with smaller and larger cabins by deforming the truck with a horizontal shear velocity field. We compare the results from using random noise to those using noise warped with our particle-based method. Using the warped noise improves the consistency, reducing the flickering of the cars' geometries and textures. We show the results in Figure B.2 and refer to our supplementary video for better visualization.

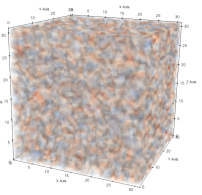


Figure 6: 3D noise warped by our particle variant.

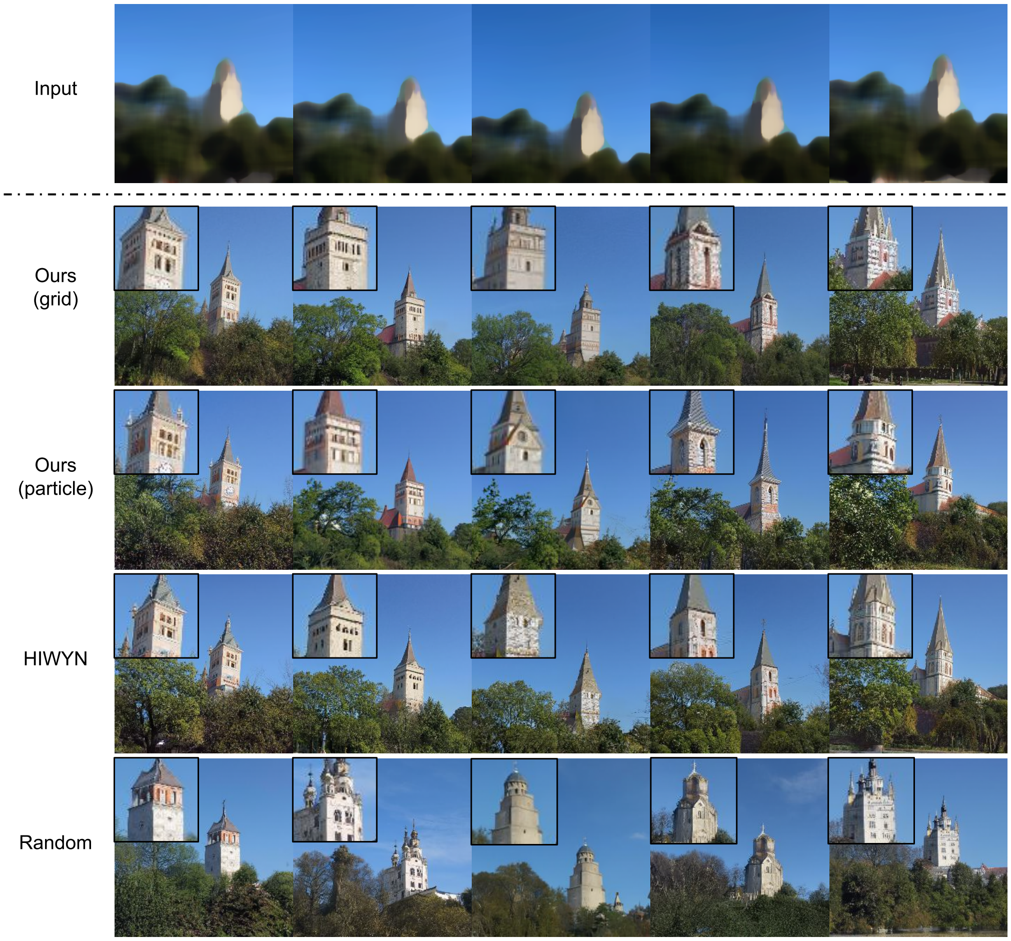


Figure 9: We compare the consistency-preserving efficacy with different noise warp schemes. We use SDEdit using the conditional signal on the top image. We show here our method (both variants) and HIWYN, while further results are given in Figure B.4. We highlight the details of the tower, which is preserved to similar extents by the integral noise-based methods.

## 4 RELATED WORKS

**Noise in Diffusion Models.** Diffusion models generate images from input noise, and noise can thus be considered the counterpart to the latent codes utilized in GAN models. As such, the outputs of diffusion models have dependencies and correlations to the initial input noise, making noise a useful handle to control temporal consistency (Khachatryan et al., 2023). In addition to Chang et al. (2024) which this work was inspired by and improves upon, there are various other temporal noise manipulation techniques that do not preserve Gaussian noise distribution– some methods (Ma et al. (2024); Ren et al. (2024)) blend high frequency Gaussian noise with low frequency motion, while others (Mokady et al. (2022); Wallace et al. (2022)) rely on approximating the inversion of noise from temporally coherent image sequences. Pandey et al. (2024) goes one step further and manipulates inverted noise in 3D space. These approaches are flexible but degrade the output of the diffusion model due to the domain gap between inference time noise and training time noise, and as such, have occasionally been accompanied by mitigation strategies such as anisotropic diffusion (Yu et al. (2024)). Noise manipulation is also not limited to the generation and stylization of videos, but has various applications in image editing (Hou et al. (2024); Pandey et al. (2024)) and 3D mesh texturing (Richardson et al. (2023)) as well.

**Noise in Computer Graphics.** While our noise warping work draws main inspiration from simulation techniques, spatial noise manipulation has been extensively studied in the graphics community through applications in animation and rendering. Works like (Kass & Pesare, 2011; Burley et al.,

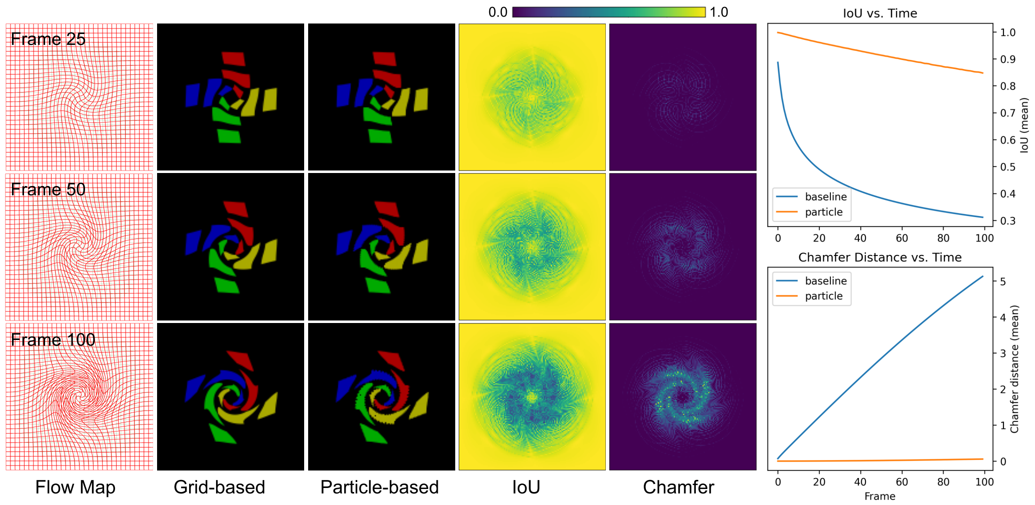


Figure 10: Comparison between the grid-based and particle-based variants for building partition records when the deformation map is diffeomorphic. The first column shows the deformation map at different frames. The second and third columns visualize warped pixel regions for the two methods. The right two columns show IoU (larger is better) and Chamfer distance (smaller is better) between the outputs from both variants. We plot the distance between particle and grid variants alongside a baseline, which is the distance between identity map and the grid variant, which show that the particle-based version remains close to the grid-based version even under large distortion.

2024) present 2D noise manipulation techniques that add a stylized organic hand-drawn look to computer-generated animation via dynamic noise textures (Perlin, 1985). In order to make sure the stylization is temporally consistent and visually pleasing, noise textures are deformed in a way that makes them consistent with the underlying animation, but little emphasis is given to the preservation/rigor of the noise distribution. On the other hand, properties of 2D spatial noise have been extensively and rigorously studied in rasterization and raytracing literature (Cook, 1986; Lagae & Dutré, 2008), originating from the idea of using dithering to reduce banding and quantization artefacts in image signal processing (Roberts, 1962). In particular, the lack of low frequency details and clumping in blue noise as opposed to white Gaussian noise has made it the choice of foundational antialiasing methods such as Poisson disc sampling (McCool & Fiume, 1992), and recent progress made in this line of antialiasing research has close ties with our methodology. For example, Wolfe et al. (2022) look at accelerating rendering tasks by extending spatial blue noise to the temporal domain, while Huang et al. (2024) show promising results in supplementing white noise with blue noise during diffusion model training.

## 5 CONCLUSIONS

We present infinite-resolution integral noise warping (Algorithm 1), a novel and efficient algorithm for warping a prior noise image into a sequence of noise frames while preserving the Gaussianity and temporal consistency for the purpose of video generation. This is achieved by a theoretically motivated analysis of the infinite-resolution case of the integral noise warping algorithm by Chang et al. (2024), which enables an orders-of-magnitude improvement in efficiency with no trade-offs in accuracy, as demonstrated by our experiments.

We note some directions for future work. Since our particle-based variant does not leverage the deformation gradient of $\psi$, it does not account for area contraction and expansion. Voronoi repartitioning may address this problem at the cost of extra computation. More broadly, since our method relies on the consistency of the deformation map and its alignment with the conditional video, it can be limited by the availability and accuracy of flow extraction techniques. It also remains an open problem of how the quality of noise warping affects video generation beyond empirical verification. Finally our method is only applied to pixel-space (or voxel-space) diffusion models, and its applicability to latent diffusion models remains to be investigated.

# REFERENCES

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. *arXiv preprint arXiv:2403.17377*, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jeremiah U Brackbill, Douglas B Kothe, and Hans M Ruppel. Flip: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, 1988.

T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011. URL `http://lmb.informatik.uni-freiburg.de/Publications/2011/Bro11a`.

Brent Burley, Brian Green, and Daniel Teece. Dynamic screen space textures for coherent stylization. In *ACM SIGGRAPH 2024 Talks*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705151. doi: 10.1145/3641233.3664321. URL `https://doi.org/10.1145/3641233.3664321`.

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023.

Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023.

Robert L Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)*, 5(1):51–72, 1986.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations*, 2024.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Xingzhong Hou, Boxiao Liu, Yi Zhang, Jihao Liu, Yu Liu, and Haihang You. Easydrag: Efficient point-based manipulation on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8404–8413, June 2024.

Xingchang Huang, Corentin Salaun, Cristina Vasconcelos, Christian Theobalt, Cengiz Oztireli, and Gurprit Singh. Blue noise for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Michael Kass and Davide Pesare. Coherent noise for non-photorealistic rendering. *ACM Transactions on Graphics (TOG)*, 30(4):1–6, 2011.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15954–15964, 2023.

Min-Seop Kwak, Donghoon Ahn, Ines Hyeonsu Kim, Jin-wha Kim, and Seungryong Kim. Geometry-aware score distillation via 3d consistent noising and gradient consistency modeling. *arXiv preprint arXiv:2406.16695*, 2024.

Ares Lagae and Philip Dutré. A comparison of methods for generating poisson disk distributions. In *Computer Graphics Forum*, volume 27, pp. 114–129. Wiley Online Library, 2008.

Xin Ma, Yaohui Wang, Gengyu Jia, Xinyuan Chen, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Cinemo: Consistent and controllable image animation with motion diffusion models. *arXiv preprint arXiv:2407.15642*, 2024.

Michael McCool and Eugene Fiume. Hierarchical poisson disk sampling distributions. In *Graphics interface*, volume 92, pp. 94–105, 1992.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

Peter Mörters and Yuval Peres. *Brownian motion*, volume 30. Cambridge University Press, 2010.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Jan K Pachl. Disintegration and compact measures. *Mathematica Scandinavica*, pp. 157–168, 1978.

Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J Mitra. Diffusion handles enabling 3d edits for diffusion models by lifting activations to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7695–7704, 2024.

Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.

Weiming Ren, Harry Yang, Ge Zhang, Cong Wei, Xinrun Du, Stephen Huang, and Wenhu Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023*, 2023.

L. Roberts. Picture coding using pseudo-random noise. *IRE Transactions on Information Theory*, 8 (2):145–154, 1962. doi: 10.1109/TIT.1962.1057702.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations, 2022. URL `https://arxiv.org/abs/2211.12446`.

Alan Wolfe, Nathan Morrical, Tomas Akenine-Möller, and Ravi Ramamoorthi. Spatiotemporal Blue Noise Masks. In Abhijeet Ghosh and Li-Yi Wei (eds.), *Eurographics Symposium on Rendering*. The Eurographics Association, 2022. ISBN 978-3-03868-187-8. doi: 10.2312/sr.20221161.

Xi Yu, Xiang Gu, Haozhi Liu, and Jian Sun. Constructing non-isotropic gaussian diffusion model using isotropic gaussian diffusion model for image editing. *Advances in Neural Information Processing Systems*, 36, 2024.

Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *arXiv preprint arXiv:2403.19655*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023a.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023b.

## A PROOF OF THEOREM 1

*Proof.* By unrolling the definitions, for $t \in [0,1]$, we have

$$H_n(t) = S_n^*(t) - tS_n^*(1) + tc, \qquad S_n^*(t) \coloneqq \frac{1}{\sqrt{n}}\left(\sum_{i=1}^{\lfloor nt \rfloor} Z_i + (nt - \lfloor nt \rfloor) Z_{\lfloor nt \rfloor + 1}\right).$$

By Mörters & Peres (2010, Theorem 5.22), $\{S_n^*\}_{n\in\mathbb{Z}_{\geq 1}}$ converges in distribution to $W(t)$ under the sup-norm metric of $C[0,1]$. To lift this convergence to the sequence $\{H_n\}_{n\in\mathbb{Z}_{\geq 1}}$, observe that the function $g : C[0,1] \to C[0,1]$ defined by

$$g(x(t)) \coloneqq x(t) - tx(1) + tc$$

is continuous under the sup-norm metric. To verify this, suppose $\lim_{n\to\infty} f_n = f$ for $\{f_n\}_{n\in\mathbb{Z}_{\geq 1}}, f \in C[0,1]$. Then

$$\begin{aligned}\|g(f_n) - g(f)\|_\infty &= \sup_{t\in[0,1]} |(f_n(t) - tf_n(1) + tc) - (f(t) - tf(1) + tc)| \\ &\leq \|f_n - f\|_\infty + \|f_n(1) - f(1)\| \leq 2\|f_n - f\|_\infty \to 0.\end{aligned}$$

Hence, by the continuous mapping theorem,

$$g(S_n^*) = H_n \xrightarrow{d} B(t) - tB(1) + tc.$$

To show

$$W(t) - tW(1) + tc = (W(t) \mid W(1) = c),$$

first of all, the conditioning $(W(t) \mid W(1) = c)$ is interpreted as the limit of $(W(t) \mid |W(1) - c| < \epsilon)$ as $\epsilon \to 0$. Denote $Y(t) \coloneqq W(t) - tW(1)$, so that $W(t) = Y(t) + tW(1)$. Since

$\mathrm{Cov}(Y(t), tW(1)) = \mathrm{Cov}(W(t) - tW(1), tW(1)) = t\mathrm{Cov}(W(t), W(1)) - t^2\mathrm{Var}(W(1), W(1)) = 0$ and that $Y(t), tW(1)$ are jointly Gaussian, they are independent. Therefore,

$$\begin{aligned}\lim_{\epsilon\to 0}(W(t) \mid |W(1) - c| < \epsilon) &= \lim_{\epsilon\to 0}(Y(t) + tW(1) \mid |W(1) - c| < \epsilon) \\ &= Y(t) + \lim_{\epsilon\to 0}(tW(1) \mid |W(1) - c| < \epsilon) \\ &= W(t) - tW(1) + tc.\end{aligned}$$

□

## B ADDITIONAL RESULTS

In this section we include additional visual and numerical results. In Figure B.4 and Figure B.5, we showcase full comparisons with the addition of Control-A-Video (Chen et al., 2023) and PYoCo (Ge et al., 2023), along with baselines with fixed noise and interpolated noise using bilinear and nearest interpolating scehemes. The corresponding quantitative metrics for both church and cat scenes are reported in Figure B.1. In Figure B.3, we use additional examples to showcase the area contention issue caused by degenerate meshes that applies similarly to our grid-based variant and Chang et al. (2024), and highlight the robustness of our particle-based variant. In Figure B.2, we show additional results when combining our particle-based 3D noise warp with GaussianCube (Zhang et al., 2024).

| Video Generation Quality (Church) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | Ours (G) | Ours (P) | HIWYN | PYoCo | CaV | Random | Fixed | Bilinear | Nearest |
| *Consistency* ↓ | 9.868e-2 | 1.065e-1 | 1.060e-1 | 1.175e-1 | 1.359e-1 | 1.538e-1 | 1.120e-1 | 8.114e-2 | 1.305e-1 |
| *Realism* ↓ | 4.643e-2 | 5.180e-2 | 4.959e-2 | 4.119e-2 | 4.069e-2 | 3.731e-2 | 3.911e-2 | 2.301e-1 | 7.012e-2 |
| *Faithfulness* ↓ | 3.872e-2 | 4.309e-2 | 4.377e-2 | 3.764e-2 | 4.169e-2 | 3.976e-2 | 3.264e-2 | 5.623e-2 | 9.321e-2 |
| Video Generation Quality (Cat) | | | | | | | | | |
| Metric | Ours (G) | Ours (P) | HIWYN | PYoCo | CaV | Random | Fixed | Bilinear | Nearest |
| *Consistency* ↓ | 6.001e-2 | 5.898e-2 | 5.807e-2 | 6.383e-2 | 4.280e-2 | 1.219e-1 | 3.950e-2 | 3.503e-2 | 1.058e-1 |
| *Realism* ↓ | 1.559e-1 | 1.496e-1 | 1.528e-1 | 1.506e-1 | 1.486e-1 | 1.221e-1 | 1.588e-1 | 3.687e-1 | 3.343e-1 |
| *Faithfulness* ↓ | 2.039e-2 | 2.064e-2 | 2.022e-2 | 2.023e-2 | 1.817e-2 | 2.077e-2 | 1.972e-2 | 3.809e-2 | 2.201e-1 |

Figure B.1: We show the quality metrics for conditional video generating using SDEdit. The consistency is measured using Warp MSE, and the realism and faithfulness are measured as in Meng et al. (2021).

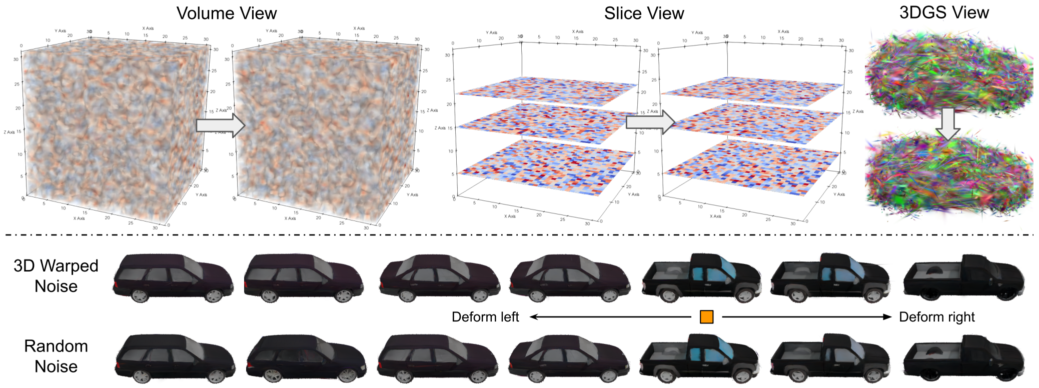


Figure B.2: Extension of our particle-based method to warping 3D noise. We show the volume render on the top left, the slice view on the top middle, and represent it as 3D Gaussian as used in GaussianCube representation on the top right. We then show that warping the noise in 3D space noticeably facilitates temporal consistency over random baseline when we perform 3D editing, which can be observed from the flickering of the color of the window in the bottom row. For best viewing of this experiment, please refer to our supplementary video.

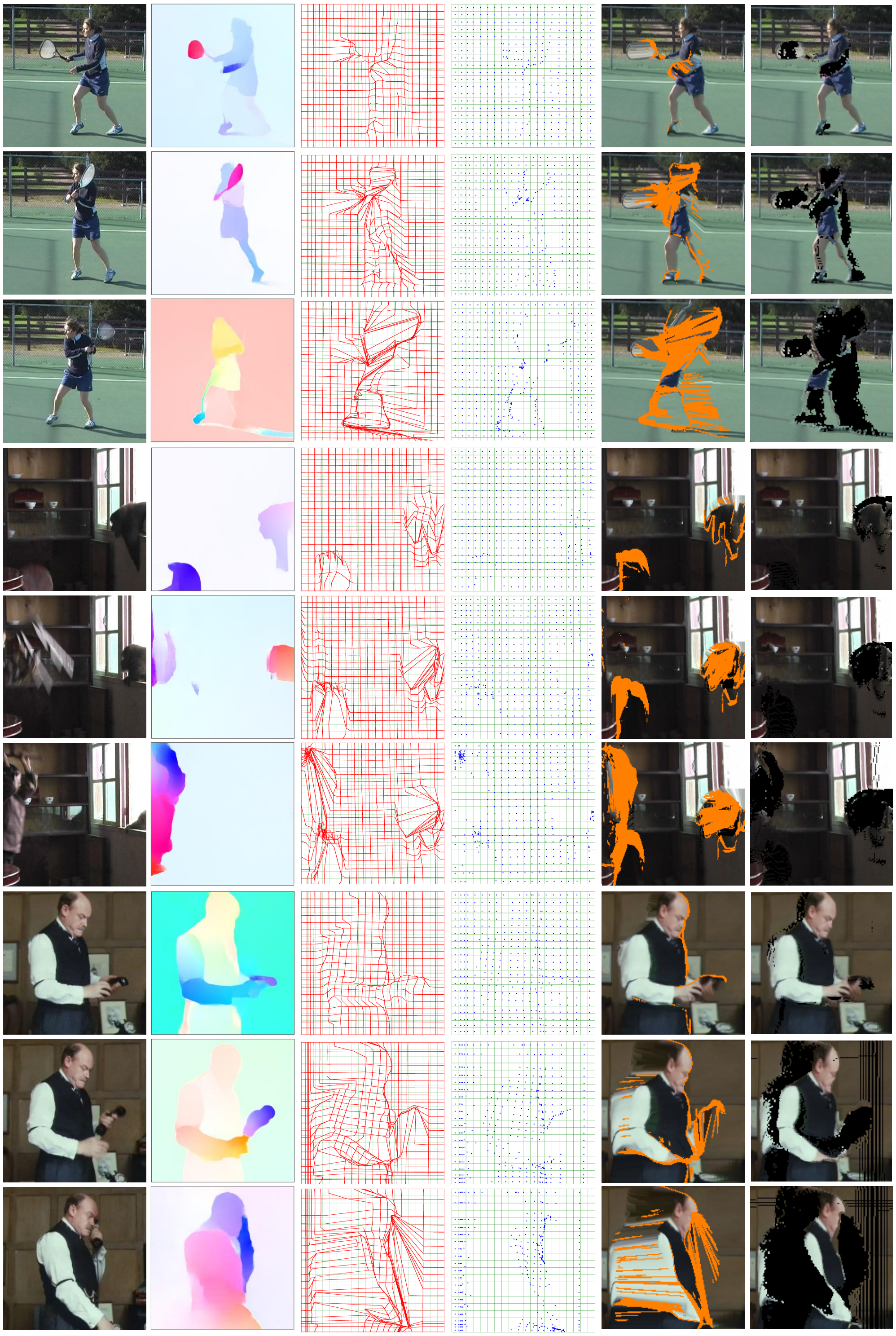

Video
Optical Flow
Grid-based
Particle-based
Area Occupation (Grid-based)
Area Occupation (Particle-based)

Figure B.3: Comparison of grid-based and particle-based variants under non-diffeomorphic deformation maps seen in real-world scenarios. The orange pixels are the invalid pixels where area overlap occurs. Flow maps are downsampled 10× for better visualization. Image sequence comes from Brox & Malik (2011) while optical flow is computed via Teed & Deng (2020).

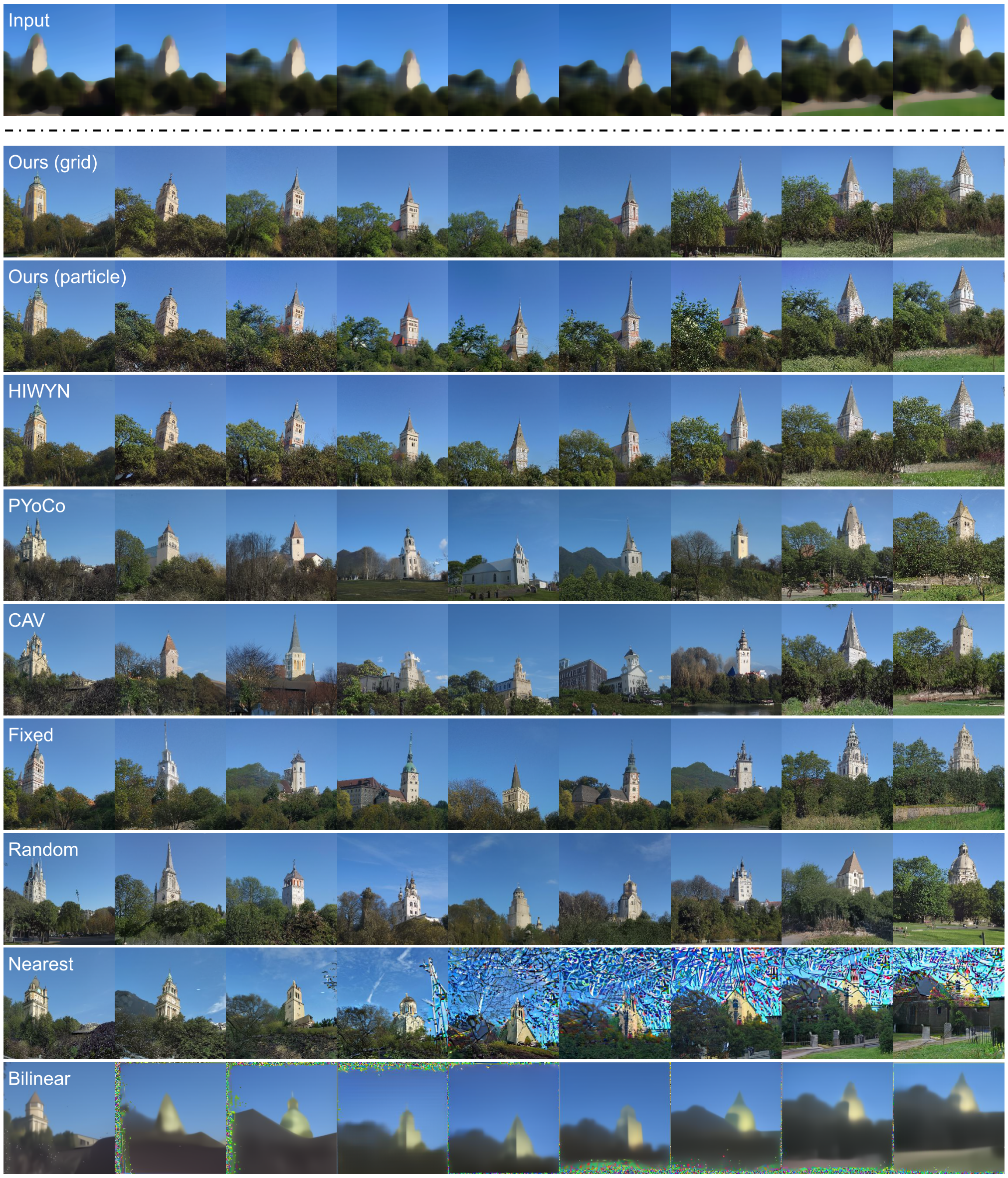


Figure B.4: The full results generated by all the compared methods on the church scene. The interpolation baselines yield noticeably corrupted results, but other yield similar quality *on the images level*. The difference lies in how the details are preserved across frames. Apart from the details of the main tower, the tree on the bottom left also exposes the interesting difference between noise initialization schemes. We refer to our supplementary video for the better visualization of these results.

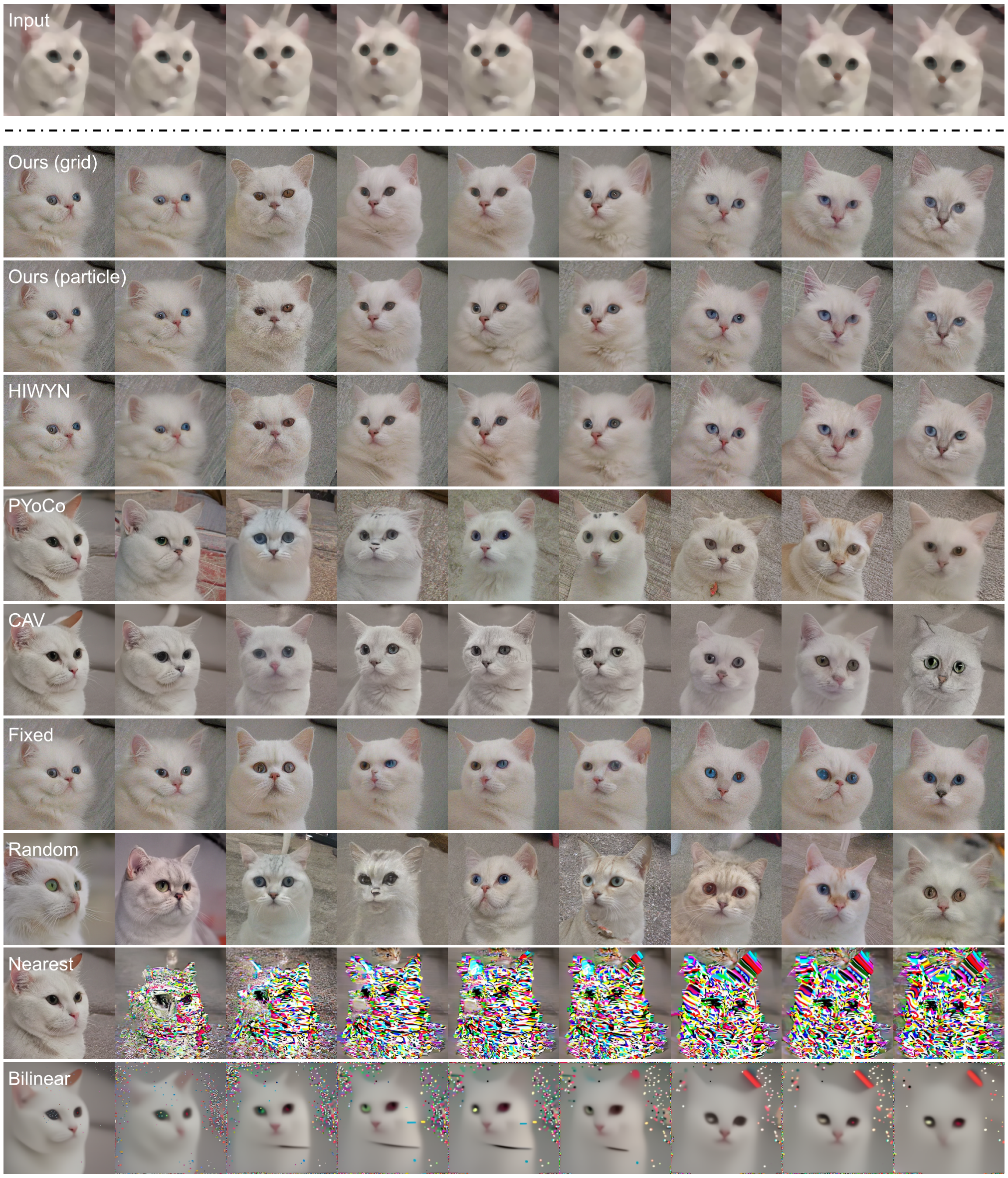


Figure B.5: The full results generated by all the compared methods on the cat scene. We observe that HIWYN and our methods (both variants) yield highly similar results. While both our variants are much faster and memory-efficient than HIWYN, this observation makes a particularly strong case for our particle-based variant considering its significantly improved simplicity and efficiency. We refer to our supplementary video for the better visualization of these results.