

899 Table of Contents

900	A Videos and Website	1
901	B Symbols and Notations	1
902	C Elaboration on Proposition 4.1	1
903	C.1 Forward Diffusion Map Under Zero-Terminal SNR	2
904	C.2 Breakdown of Injectivity	2
905	C.3 Implications for Deterministic Inversion	3
906	D Elaboration on Proposition 4.2	3
907	D.1 Proof for the Discrete Case: $k \in \mathbb{N}_{>0}$	3
908	D.2 Proof for the Continuous Case: $k \in \mathbb{R}_{>0}$	4
909	E Elaboration on Stochastic Latent Modulation	5
910	E.1 Technical Details of Stochastic Latent Modulation	5
911	E.2 Algorithm for Stochastic Latent Modulation	6
912	F More Ablation Studies	6
913	F.1 Adaptive Reference Latent Index δ Ablations	6
914	G Discussion on Quantitative Results	6
915	H Technical Elaborations Why not set $\bar{\alpha}_T = 0$?	8
916	I Parameter Settings	8
917	I.1 How To Choose k ?	8
918	I.2 How To Choose δ ?	8
919	J Proposed method on Wan 2.1 (for Flow Matching models in general)	8

920 A Videos and Website

921 To facilitate comprehensive evaluation and enhance result accessibility, we provide 100+ video results
922 including motivation examples, qualitative results, ablation studies, qualitative comparisons, and
923 limitations in our project page.

924 B Symbols and Notations

925 In this section, we present the symbols and notations used throughout the paper to ensure clarity and
926 consistency in our mathematical and algorithmic descriptions.

927 C Elaboration on Proposition 4.1

928 Consider a variance-preserving noise schedule $\{\alpha_t\}_{t=0}^T$ with cumulative products defined as $\bar{\alpha}_t =$
929 $\prod_{s=1}^t (1 - \beta_s)$, where the schedule enforces a zero terminal signal-to-noise ratio (SNR), such that

Symbol	Description
Video and Frame Symbols	
\mathbf{V}	Source video
\mathbf{I}_i	Individual frame of the source video
$\mathbf{D} = \{D_i\}_{i=1}^n$	Sequence of depth maps
D_i	Depth map for frame \mathbf{I}_i
\mathbf{K}	Camera intrinsics matrix
\mathbf{P}_i	Point cloud for frame \mathbf{I}_i
\mathbf{T}_i	Target camera pose for frame i
\mathbf{I}'_i	Rendered novel view for frame i
\mathbf{M}'	Visibility masks for novel views
Latent Space Symbols	
Φ_t	Diffusion for timestep t
α_t	Cumulative signal coefficient at timestep t
$\bar{\alpha}_t$	Cumulative product of α_t
\mathbf{x}_{init}	Initial noise for diffusion process
\mathbf{x}_0	VAE-encoded latent also our pivot latent
ϵ	Noise sample from standard normal distribution
ϵ^{inv}	DDIM inverted latent
$\epsilon^{(k)}$	K-order recursive noise representation
Mask and Modulation Symbols	
\mathbf{M}	Binary occlusion mask
\mathbf{D}	Depth-based near depth mask
\mathbf{S}	Visibility-aware sampling mask
$\mathcal{P}_{\mathbf{S}}$	Stochastic permutation operator
$\tilde{\mathbf{x}}_0$	Modulated content latent
$\tilde{\epsilon}^{\text{inv}}$	Modulated noise latent

Table 1: List of symbols used in the paper.

930 $\bar{\alpha}_T = 0$. The forward diffusion map is given by:

$$\Phi_T(x_0, \epsilon) = \sqrt{\bar{\alpha}_T}x_0 + \sqrt{1 - \bar{\alpha}_T}\epsilon,$$

931 where $x_0 \in \mathbb{R}^{F \times C \times H \times W}$ is the initial latent variable, and $\epsilon \sim \mathcal{N}(0, I)$ is a noise sample drawn from a
932 standard normal distribution.

933 C.1 Forward Diffusion Map Under Zero-Terminal SNR

934 Since the zero-terminal SNR noise schedule specifies $\bar{\alpha}_T = 0$, substitute this into the definition of
935 Φ_T :

$$\Phi_T(x_0, \epsilon) = \sqrt{\bar{\alpha}_T}x_0 + \sqrt{1 - \bar{\alpha}_T}\epsilon = \sqrt{0}x_0 + \sqrt{1 - 0}\epsilon = 0 \cdot x_0 + 1 \cdot \epsilon = \epsilon.$$

936 Thus, $\Phi_T(x_0, \epsilon) = \epsilon$, which depends solely on the noise ϵ and is independent of the initial latent x_0 .
937 For any two initial latents $x_0, x'_0 \in \mathbb{R}^{F \times C \times H \times W}$ and a fixed noise sample ϵ , it follows that:

$$\Phi_T(x_0, \epsilon) = \epsilon \quad \text{and} \quad \Phi_T(x'_0, \epsilon) = \epsilon.$$

938 Therefore, $\Phi_T(x_0, \epsilon) = \Phi_T(x'_0, \epsilon) = \epsilon$, regardless of whether $x_0 = x'_0$ or $x_0 \neq x'_0$.

939 C.2 Breakdown of Injectivity

940 A function $f : A \rightarrow B$ is injective if, for all $a, a' \in A$, $f(a) = f(a')$ implies $a = a'$. Consider
941 the map $\Phi_T(\cdot, \epsilon) : \mathbb{R}^{F \times C \times H \times W} \rightarrow \mathbb{R}^{F \times C \times H \times W}$ with ϵ fixed. From §C.1, for any distinct $x_0, x'_0 \in$
942 $\mathbb{R}^{F \times C \times H \times W}$ where $x_0 \neq x'_0$, we have:

$$\Phi_T(x_0, \epsilon) = \epsilon = \Phi_T(x'_0, \epsilon).$$

943 Since $\Phi_T(x_0, \epsilon) = \Phi_T(x'_0, \epsilon)$ holds even when $x_0 \neq x'_0$, the condition for injectivity is violated.
944 Hence, $\Phi_T(\cdot, \epsilon)$ is not injective in x_0 , as multiple (indeed, all) initial latents x_0 map to the same
945 output ϵ for a given ϵ .

C.3 Implications for Deterministic Inversion

In diffusion models, the terminal state is denoted $x_T = \Phi_T(x_0, \epsilon)$, which, under the condition $\bar{\alpha}_T = 0$, simplifies to $x_T = \epsilon$. Deterministic inversion methods, such as DDIM inversion, aim to recover the original latent x_0 from x_T by reversing the forward diffusion process. These methods assume that the forward map Φ_T can be inverted uniquely, which requires Φ_T to be injective. However, since $\Phi_T(\cdot, \epsilon)$ is not injective, multiple distinct x_0 produce the same $x_T = \epsilon$. Consequently, given only x_T , it is impossible to determine which x_0 among the infinitely many possible initial latents was the original, rendering unique recovery via deterministic inversion unfeasible.

D Elaboration on Proposition 4.2

In this section, we prove the closed-form expressions associated with the recursive noise initialization process K-RNR outlined in Proposition 4.2. The recursive process is defined as follows: for an initial step where $k = 1$, the expression is given by

$$\epsilon^{(1)} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon^{\text{inv}},$$

and for subsequent steps where $k > 1$, the expression becomes

$$\epsilon^{(k)} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon^{(k-1)}.$$

Here, $x_0 \in \mathbb{R}^{F \times C \times H \times W}$ represents the pivot latent variable, $\bar{\alpha}_t > 0$ denotes the cumulative signal coefficient at timestep t , and ϵ^{inv} is the initial noise term.

The proposition posits two closed-form expressions. For the discrete recursion depth, where $k \in \mathbb{N}_{\geq 0}$, the expression is

$$\epsilon^{(k)} = \left(\sum_{i=1}^k \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^k \epsilon^{\text{inv}}.$$

For the continuous recursion depth, where $k \in \mathbb{R}_{\geq 0}$, the expression is

$$\epsilon^{(k)} = \left(\sqrt{\bar{\alpha}_t} \frac{1 - (\sqrt{1 - \bar{\alpha}_t})^k}{1 - \sqrt{1 - \bar{\alpha}_t}} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^k \epsilon^{\text{inv}}.$$

The proof is divided into two parts: the discrete case is addressed in §D.1 and continuous case is addressed in §D.2

D.1 Proof for the Discrete Case: $k \in \mathbb{N}_{\geq 0}$

To verify the closed-form expression for discrete values of k , mathematical induction is employed as a method of proof.

For the initial step, consider the case where $k = 1$. The recursive definition states that

$$\epsilon^{(1)} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon^{\text{inv}}.$$

To confirm this, the proposed closed-form expression is evaluated at $k = 1$:

$$\epsilon^{(1)} = \left(\sum_{i=1}^1 \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^1 \epsilon^{\text{inv}}.$$

The summation involves only one term, corresponding to $i = 1$. This term is calculated as follows:

$$\sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{1-1} = \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^0 = \sqrt{\bar{\alpha}_t} \cdot 1 = \sqrt{\bar{\alpha}_t}.$$

Thus, the closed-form expression becomes

$$\epsilon^{(1)} = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon^{\text{inv}},$$

which is identical to the recursive definition. This establishes the validity of the expression for the base case.

975 Next, suppose that for some positive integer $n \geq 1$, the closed-form expression holds true:

$$\epsilon^{(n)} = \left(\sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^n \epsilon^{\text{inv}}.$$

976 The objective is now to demonstrate that this expression remains valid for the next integer, $k = n + 1$.

977 According to the recursive definition,

$$\epsilon^{(n+1)} = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon^{(n)}.$$

978 The inductive hypothesis is substituted into this equation, yielding

$$\epsilon^{(n+1)} = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \left[\left(\sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^n \epsilon^{\text{inv}} \right].$$

979 The factor $\sqrt{1 - \bar{\alpha}_t}$ is applied to each term within the brackets. For the summation term, this results
980 in

$$\sqrt{1 - \bar{\alpha}_t} \cdot \sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} = \sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i,$$

981 and for the noise term,

$$\sqrt{1 - \bar{\alpha}_t} \cdot (\sqrt{1 - \bar{\alpha}_t})^n = (\sqrt{1 - \bar{\alpha}_t})^{n+1}.$$

982 Thus, the expression for $\epsilon^{(n+1)}$ is written as

$$\epsilon^{(n+1)} = \sqrt{\bar{\alpha}_t} x_0 + \left(\sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^{n+1} \epsilon^{\text{inv}}.$$

983 The terms involving x_0 are then grouped together:

$$\epsilon^{(n+1)} = \left(\sqrt{\bar{\alpha}_t} + \sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^{n+1} \epsilon^{\text{inv}}.$$

984 To express this as a single summation, it is noted that $\sqrt{\bar{\alpha}_t}$ can be written as $\sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^0$. This
985 allows the expression to be rewritten by adjusting the summation indices:

$$\sqrt{\bar{\alpha}_t} + \sum_{i=1}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i = \sum_{i=0}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i.$$

986 This summation from $i = 0$ to n corresponds exactly to the desired form when re-indexed:

$$\sum_{i=0}^n \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^i = \sum_{i=1}^{n+1} \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1},$$

987 since each term aligns appropriately with the change in index. Therefore, the expression becomes

$$\epsilon^{(n+1)} = \left(\sum_{i=1}^{n+1} \sqrt{\bar{\alpha}_t} (\sqrt{1 - \bar{\alpha}_t})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^{n+1} \epsilon^{\text{inv}},$$

988 which matches the proposed closed-form expression for $k = n + 1$. This step confirms the inductive
989 hypothesis for the next integer, and by the principle of mathematical induction, the closed-form
990 expression is valid for all positive integers k which completes the proof ■

991 **D.2 Proof for the Continuous Case: $k \in \mathbb{R}_{\geq 0}$**

992 To extend the result to real values of k , the discrete case's summation is analyzed as a geometric series.
993 Let the ratio $r = \sqrt{1 - \bar{\alpha}_t}$, where, given $0 < \bar{\alpha}_t < 1$, it follows that $0 < r < 1$. The summation in
994 the discrete expression is expressed as

$$\sum_{i=1}^k \sqrt{\bar{\alpha}_t} r^{i-1} = \sqrt{\bar{\alpha}_t} \sum_{i=0}^{k-1} r^i.$$

995 The formula for the sum of a finite geometric series is applied here:

$$\sum_{i=0}^{k-1} r^i = \frac{1 - r^k}{1 - r}.$$

996 This allows the summation to be rewritten as

$$\sqrt{\bar{\alpha}_t} \sum_{i=0}^{k-1} r^i = \sqrt{\bar{\alpha}_t} \cdot \frac{1 - r^k}{1 - r}.$$

997 Substituting $r = \sqrt{1 - \bar{\alpha}_t}$ back into the expression, it becomes

$$\sqrt{\bar{\alpha}_t} \cdot \frac{1 - (\sqrt{1 - \bar{\alpha}_t})^k}{1 - \sqrt{1 - \bar{\alpha}_t}}.$$

998 Incorporating this into the discrete closed-form expression, the result is

$$\epsilon^{(k)} = \left(\sqrt{\bar{\alpha}_t} \cdot \frac{1 - (\sqrt{1 - \bar{\alpha}_t})^k}{1 - \sqrt{1 - \bar{\alpha}_t}} \right) x_0 + (\sqrt{1 - \bar{\alpha}_t})^k \epsilon^{\text{inv}}.$$

999 This formulation is well-defined for all real $k \geq 0$, as the exponential terms are continuous functions
1000 over the real numbers which completes the proof ■

1001 E Elaboration on Stochastic Latent Modulation

1002 In this section, we provide a detailed technical elaboration of the Stochastic Latent Modulation
1003 (SLM) mechanism, a key component of our approach to dynamic view synthesis. SLM addresses
1004 the challenge of synthesizing plausible content for regions that become newly visible due to camera
1005 motion, operating directly in the latent space of a pre-trained video diffusion model. This process
1006 modulates both the VAE-encoded latent \mathbf{x}_0 and the inverted latent ϵ^{inv} using a single binary occlusion
1007 mask and depth map, ensuring a consistent and efficient strategy for handling occlusions. By
1008 leveraging visibility-aware sampling and stochastic permutation, SLM enables the diffusion model to
1009 infer content for occluded regions without requiring architectural changes or additional training.

1010 E.1 Technical Details of Stochastic Latent Modulation

1011 The SLM process modulates the latents \mathbf{x} and ϵ by filling their occluded regions with values sampled
1012 from visible, depth-specific areas, using a single mask \mathbf{M} and depth map \mathbf{D} to guide the operation.
1013 This begins with the computation of a visibility mask, defined as $\mathbf{V} = (1 - \mathbf{M}) \cdot (\mathbf{D})$, which identifies
1014 regions that are both visible (where $\mathbf{M} = 0$) and depthwise near (where \mathbf{D}). These regions serve as
1015 the source pool for sampling, as they contain stable and contextually relevant latent values from the
1016 scene. The target regions, where content synthesis is needed, correspond to the occluded areas where
1017 $\mathbf{M} = 1$.

1018 The modulation proceeds by identifying the spatial indices of the source and target regions. The set
1019 of source indices, $\mathcal{I}_{\text{source}}$, consists of all positions where $\mathbf{V} = 1$, while the set of target indices, $\mathcal{I}_{\text{target}}$,
1020 includes all positions where $\mathbf{M} = 1$. For each latent, SLM counts the number of occluded elements
1021 (i.e., the size of $\mathcal{I}_{\text{target}}$) and randomly selects an equal number of indices from $\mathcal{I}_{\text{source}}$. These randomly
1022 chosen source values are then assigned to the target positions. Specifically, for \mathbf{x} , the values at indices
1023 $\mathbf{i} \in \mathcal{I}_{\text{target}}$ are replaced with values from randomly selected indices $\mathbf{j} \in \mathcal{I}_{\text{source}}$, such that $\mathbf{x}_{\mathbf{i}} = \mathbf{x}_{\mathbf{j}}$.
1024 The same process is applied to ϵ , where $\epsilon_{\mathbf{i}} = \epsilon_{\mathbf{j}}$ for corresponding pairs of indices. This stochastic
1025 sampling ensures that the occluded regions of both latents are populated with plausible content drawn
1026 from the visible, near-depth areas of the scene.

1027 The use of a single mask and depth map for both \mathbf{x} and ϵ ensures that the source and target regions
1028 remain consistent across the two latents, while the independent application of the sampling process to
1029 each latent preserves their distinct roles in the diffusion pipeline. The randomness in selecting source
1030 indices introduces variability, allowing the diffusion model to explore diverse completions for the
1031 occluded regions, all while maintaining coherence with the visible parts of the scene.

1032 E.2 Algorithm for Stochastic Latent Modulation

Algorithm 1 Stochastic Latent Modulation

```

1: Input:  $\mathbf{x} \in \mathbb{R}^{B \times F \times C \times H \times W}$ ,  $\epsilon \in \mathbb{R}^{B \times F \times C \times H \times W}$ ,  $\mathbf{M} \in \{0, 1\}^{B \times F \times C \times H \times W}$ ,  $\mathbf{D} \in \mathbb{R}^{B \times F \times C \times H \times W}$ 
2: Output: Modulated  $\mathbf{x}$ , Modulated  $\epsilon$ 
3: Compute visibility mask  $\mathbf{V} = (1 - \mathbf{M}) \cdot \mathbf{D}$ 
4: Let  $\mathcal{I}_{\text{source}} = \{\mathbf{i} \mid \mathbf{V}_{\mathbf{i}} = 1\}$ 
5: Let  $\mathcal{I}_{\text{target}} = \{\mathbf{i} \mid \mathbf{M}_{\mathbf{i}} = 1\}$ 
6: for each  $\mathbf{i} \in \mathcal{I}_{\text{target}}$  do
7:   Sample  $\mathbf{j} \sim \text{Uniform}(\mathcal{I}_{\text{source}})$ 
8:   Set  $\epsilon_{\mathbf{i}} = \epsilon_{\mathbf{j}}$ 
9:   Set  $\mathbf{x}_{\mathbf{i}} = \mathbf{x}_{\mathbf{j}}$ 
10: end for
11: return  $\mathbf{x}$ ,  $\epsilon$ 

```

1033 F More Ablation Studies

1034 In this section, we present additional ablation studies to further analyze the components of our
1035 approach. in §F.1 we analyze the role of the adaptive normalization latent depth δ .

1036 F.1 Adaptive Reference Latent Index δ Ablations

1037 Figure 1 presents an ablation study on the choice
1038 of the adaptive latent index δ , which determines
1039 the reference noise level used for adaptive nor-
1040 malization between the k -th order noise and the
1041 δ -order noise. In all our experiments, we set
1042 $\delta = 3$, and the results in this ablation empirically
1043 validate this design choice. When $\delta = 3$, the
1044 model achieves the highest reconstruction qual-
1045 ity across all evaluation metrics, with a PSNR of
1046 24.97, SSIM of 0.885, and LPIPS of 0.078.

δ Index	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
$\delta = 1$	10.32	0.342	0.883
$\delta = 2$	19.23	0.748	0.148
$\delta = 3$	24.97	0.885	0.078
$\delta = 4$	15.29	0.592	0.240
$\delta = 5$	13.92	0.468	0.329
$\delta = 6$	12.66	0.333	0.451
$\delta = 7$	11.28	0.244	0.604

Figure 1: Ablation on the adaptive index δ .

1047 Performance degrades notably when δ deviates
1048 from this setting. For instance, lower values of δ
1049 such as 1 and 2 lead to insufficient regularization, producing reconstructions with low fidelity and
1050 poor perceptual quality. Conversely, higher values of δ (i.e., $\delta \geq 4$) introduce excessive deviation
1051 in the normalization reference, which appears to destabilize the refinement process and result in
1052 less consistent outputs. This pattern suggests that $\delta = 3$ offers an optimal trade-off by aligning the
1053 reference noise distribution closely with the target generation stage, enabling more effective adaptive
1054 normalization. These findings confirm that careful selection of the latent reference index is critical
1055 for preserving quality in recursive refinement.

1056 G Discussion on Quantitative Results

1057 Table 1 and Table 2 in the main paper present a comprehensive quantitative evaluation of our frame-
1058 work against recent methods across multiple axes, including visual quality, camera pose accuracy,
1059 view synchronization, and reconstruction fidelity. The baseline methods span three architectural
1060 families: GCD and TrajectoryAttention are built upon the Stable Video Diffusion backbone, Diffu-
1061 sion as Shader (DaS) and TrajectoryCrafter share the CogVideoX foundation with our method, and
1062 ReCamMaster is based on the Wan architecture.

1063 In our experiments, we observe that methods relying on Stable Video Diffusion, such as GCD and
1064 TrajectoryAttention, consistently underperform in preserving the identity and motion dynamics of the
1065 original videos when camera transformations are introduced. This can be attributed to the limited
1066 expressiveness of the Stable Video Diffusion architecture compared to the more semantically rich
1067 representations offered by CogVideoX and Wan. Among the CogVideoX-based approaches, Diffusion



Figure 2: **Video Reconstruction Strategies.** We perform quantitative and qualitative evaluation on video reconstruction without camera transformation application. Video results can be found in the supplementary material.

as Shader struggles to maintain action fidelity, often generating semantically coherent frames that fail to reflect the intended motion trajectory. TrajectoryCrafter achieves a stronger balance between action fidelity and identity preservation; however, we note that identity consistency tends to degrade toward the latter segments of the video. ReCamMaster, while effective in its synthesis, incurs significant inefficiency due to its reliance on concatenating source and target video frames along the frame channel. This design increases the overall token sequence length, which not only limits scalability but also results in considerably slower inference speeds. In contrast, our proposed method retains both high fidelity and identity consistency across the video while maintaining efficient inference. The quantitative comparisons are shared in [website.html](#).

1077 H Technical Elaborations | Why not set $\bar{\alpha}_T = 0$?

1078 **Q: In experiments we use a strength of 0.95 to ensure $\bar{\alpha}_T > 0$, why not set $\bar{\alpha}_T = 0$?**

1079 When $\bar{\alpha}_T = 0$, Equation (1) reduces to standard DDIM inversion, which is the main motivation of
 1080 this paper: to demonstrate that standard DDIM inversion does not work under a zero terminal SNR
 1081 setting.

1082 Let’s see this situation step by step. In Equation (1), when $t = T$ where $\bar{\alpha}_T = 0$:

$$\begin{aligned} & \left(\sum_{i=1}^k \sqrt{\bar{\alpha}_T} (\sqrt{1 - \bar{\alpha}_T})^{i-1} \right) x_0 + (\sqrt{1 - \bar{\alpha}_T})^k \varepsilon_{\text{inv}} \\ &= \left(\sum_{i=1}^k 0 \times \sqrt{1} \right) x_0 + (\sqrt{1})^k \varepsilon_{\text{inv}} \\ &= \varepsilon_{\text{inv}} \end{aligned}$$

1083 Thus, when $\bar{\alpha}_T = 0$, the entire term collapses to the pure noise term ε_{inv} , showing that no image
 1084 content can be reconstructed, precisely why $\bar{\alpha}_T$ should remain nonzero.

1085 I Parameter Settings

1086 I.1 How To Choose k ?

1087 We obtained the best results when we set $k = 3$ and $k = 6$. Note that we do not tweak the k value
 1088 per video–camera pair. We also want to clarify an important point:

- 1089 • **Book reading example:** We presented video results for $k = 20$. This choice was not made
 1090 because $k = 20$ is optimal, but rather because it represents a relatively high value of k .
 1091 Our goal in that experiment is to highlight the effectiveness of our *adaptive normalization*
 1092 extension of K-RNR when k is high, which is why we chose to demonstrate the experiment
 1093 at a higher setting.
- 1094 • **Monkey example:** We wanted to demonstrate the K-RNR’s effect on rendered videos
 1095 with increasing k values. The logic behind that experiment is demonstrating to readers the
 1096 *evolution of videos* with different k settings. As stated earlier, $k = 6$ generates plausible
 1097 results.
- 1098 • **Elephant and duck examples:** We aimed to demonstrate the effectiveness of K-RNR in
 1099 source video reconstruction when there is no occlusion (i.e., no SLM involved). We reported
 1100 results using small values of k : $[k = 2, k = 3, k = 4]$, to show that $k = 3$ is sufficient for
 1101 direct video reconstruction. We will elaborate our parameter selection process in more detail
 1102 in the camera-ready version.

1103 I.2 How To Choose δ ?

1104 In **Appendix F.3 Adaptive Reference Latent Index Ablations**, we conducted quantitative experi-
 1105 ments regarding different values (in the table, the rows correspond to different k values, while the
 1106 columns vary δ). In that experiment, we report PSNR, SSIM, and LPIPS results. As a result of this
 1107 experimental validation, we observe that the best PSNR, SSIM, and LPIPS scores are obtained when
 1108 $\delta = 3$. Therefore, in all of our experiments in the main paper and supplementary videos, we use
 1109 $\delta = 3$.

1110 J Proposed method on Wan 2.1 (for Flow Matching models in general)

1111 **K-RNR**, along with our dynamic view synthesis approach, is directly compatible with Wan 2.1
 1112 without requiring any modifications. Furthermore, in the section below, we illustrate how K-RNR

1113 enables us to **bypass traditional iterative inversion schemes**, offering a more efficient, non-iterative
 1114 alternative.

1115 In Wan noise scheduler, $\epsilon' = \alpha_t x_0 + \sigma_t \epsilon$ operation is performed, where x_0 is the VAE-encoded latent
 1116 and ϵ is sampled from a standard normal distribution.

1117 Furthermore, $\alpha_t + \sigma_t = 1$. From now on, we will use $\sigma_t = (1 - \alpha_t)$ parameterization.

1118 **We pose the following question:** *How effective is K-RNR when used without relying on any inversion*
 1119 *process?*

1120 To do so, we set $\epsilon^{(1)} = \epsilon \sim \mathcal{N}(0, I)$ and we followed our recursive noise representation formula:

K-RNR in Flow Matching

$$\epsilon^{(k)} = \alpha_t x_0 + (1 - \alpha_t) \epsilon^{(k-1)} \quad (1)$$

1121 When this recursion is solved, we obtain a closed-form solution again in the form of:

$$\epsilon^{(k)} = \left[\sum_{i=1}^k (1 - \alpha_t)^{i-1} \alpha_t x_0 \right] + (1 - \alpha_t)^k \epsilon \quad (2)$$

1122 Importantly, x_0 is sampled from the Wan 3D-VAE using `argmax-sampling`, which uses the mode =
 1123 mean of the latent distribution. Hence, $\mathbb{E}[x_0] = x_0$ and $\text{VAR}[x_0] = 0$.

1124 Now let's analyze the statistics and behavior of Eq. (2):

$$\mathbb{E}[\epsilon^{(k)}] = \left[\sum_{i=1}^k (1 - \alpha_t)^{i-1} \alpha_t \mathbb{E}[x_0] \right] = \left[\frac{1 - (1 - \alpha_t)^k}{\alpha_t} \right] \alpha_t \mathbb{E}[x_0] = [1 - (1 - \alpha_t)^k] \mathbb{E}[x_0]$$

$$\text{VAR}[\epsilon^{(k)}] = (1 - \alpha_t)^{2k}$$

1125 For the default setting, $\alpha_t = 0.07$.

1126 **Behavior of the mean.** When $k = 1$, $\mathbb{E}[\epsilon^{(1)}] = 0.07 \mathbb{E}[x_0]$. As $k \rightarrow \infty$, $\mathbb{E}[\epsilon^{(\infty)}] \rightarrow \mathbb{E}[x_0]$, so it
 1127 gets $\frac{1}{0.07} \approx 15 \times$ larger, hence **exploding**.

1128 **Behavior of the variance.** Note that we did not use inverted latents for the $\epsilon^{(1)}$ but directly set it
 1129 as standard normal, different from our paper setting. This results in a completely opposite behavior
 1130 when it comes to variance. As $k \rightarrow \infty$, $\text{VAR}[\epsilon^{(\infty)}] \rightarrow 0$, hence it is **vanishing**.

1131