arXiv:2611.32820v1 [cs.MM] 24 Nov 2026

实时交互式视频生成的残差引导

NaHyeon Park[1] Kyeong-HwanMoon[2] Youjin Jeon[1] Hyunjung Shim[11]韩国科学技术院 [2]
浦项科技大学项目页面： https://residual-guidance.github.io



图1. **残差引导（RG）使流式视频生成能够跟随提示词变化。上：** 流中属性切换。对于每个提示词，我们展示基础流式模型生成的首帧和末帧，并对比有无 RG 的情况。由于< figure>上下文惯性，基础模型会一直保持在切换前的状态直至结束：每个块都是从前一块延续生成，而不是反映新提示词；相比之下，同一模型在使用 RG 后能够实现属性变化，同时保留主体和场景布局。**下：** 来自 StoryEval 的三事件故事。虚线表示提示切换。基础模型无法推进到第一个事件之后，而 RG 则能到达每个后续事件。RG 无需训练，并且可以轻松集成到现有流式模型和基于缓存的方法中。

摘要

实时交互式视频生成允许用户在生成过程中更改提示词，以引导演化视频。然而，现有的流式模型以块为单位生成视频，切换提示词后往往仍会继续遵循先前提示。我们将这一问题归因于上下文惯性：前一块会成为强先验，从而抑制新提示。我们还发现，即使缓存中只保留前一块，这种惯性仍然存在；但丢弃上下文又会破坏视觉连续性，限制了编辑缓存的方法。我们不编辑缓存，而是提出残差引导（*RG*），一种一种简单的免训练方法，可通过在缓存之外测量校正量来抵消上下文惯性。具体而言，残差引导会在不使用缓存的情况下，根据新提示词对重新加噪的前一块进行一次去噪，并将由此得到的残差添加到下一块的早期去噪步骤中；此时该块的内容基本已确定，而新提示词受到的抑制最强。从单次切换到长时间的多切换生成，在各种场景下，残差引导都能显著提升现有流式模型的提示词切换能力，包括那些配备专用切换机制的模型，同时基本保持视觉质量。由于无需训练或针对单个模型进行调优，且每块只增加一次前向传播，残差引导可轻松集成到现有的实时视频生成流程中。

1. 引言

传统的视频扩散模型 [13, 18, 27, 31, 40,51, 58, 61, 62] 通常采用双向注意力，使视频标记能够关注过去和未来的帧。因此，它们会联合生成预先设定的帧序列，提示词需在生成前指定，而视频只有在整个序列完成去噪后才能呈现。这使用户无法在视频逐步生成时与生成过程互动。

近期的自回归视频扩散模型转而采用因果注意力 [6], , 其中每个块通过 KV缓存关注先前的块 [41]。结合从双向教师模型进行的少步蒸馏 [9, 19, 26,34, 56, 60, 63], , 这类模型能够实现实时流式生成，即每个块一经生成便立即显示。由于视频在生成过程中逐步展开，用户可以在途中更改提示词，以引导演化视频 [22, 33, 43, 47, 56, 57, 59], , 例如引入新事件、修改现有场景或控制模拟环境。然而，这种交互仍不可靠。如图 1 和图2a所示，流式模型往往无法反映提示词变化，仍会继续生成与先前提示词一致的内容。现有方法会在提示词边界处修改 KV缓存，以提升响应性 [56, 59],，但问题依然存在。

为理解这一失败，我们分析了模型对切换块的预测（图2b），**发现缓存的视觉内容会作为一种强先验，抑制新提示词，这一现象被我们称为**上下文惯性。我们的分析揭示了上下文惯性的两个特性。首先，惯性源于缓存帧所呈现的内容，而不是用于生成这些帧的提示词；即使只缓存前一块，这种惯性仍会持续存在。这解释了为什么基于缓存的方法效果不佳，因为重新缓存 [56]只会更新提示词信息，同时保留缓存帧；而刷新缓存 [59]仍会保留汇点和最后一帧。其次，这种抑制在第一个去噪步骤中最为明显，此时块的内容基本已经确定。完全移除上下文虽然可以消除惯性，却会破坏主体身份和场景连续性（图2c），这表明应当保留缓存，并在预测中抵消其影响。一种自然的做法是采用无分类器引导 [17], , 但其方向是在相同缓存条件下计算的，因此同样会受到抑制（见补充材料F.2）。综合这些观察结果可知，应当在没有缓存、上下文惯性不存在的情况下测量提示信号，并将其应用于惯性最强的位置。

基于这一洞见，我们提出了**残差引导（RG）**，这是一种简单的免训练方法，无需修改缓存即可增强新提示。具体而言，RG 在不使用缓存的情况下，根据新提示对前一个块的重新加噪副本进行去噪，并将其与原始块之间的差异加到下一个块的早期去噪步骤中。这一残差捕捉了前一个块的内容在新提示下、摆脱上下文惯性的情况下会如何重新绘制，从而使预测摆脱该内容的影响，并补充被抑制的提示信号。由于校正作用于预测而非缓存，每个块仍会关注其完整上下文，从而保留主体身份和场景连续性。出于同样的原因，RG 与缓存管理方式正交，并可与现有的基于缓存的切换方法结合使用。RG 每个块只需额外进行一次前向传播，并且在不同模型间使用同一套配置。

我们在七种流式基线方法上评估了我们的方法，其中包括采用专用切换机制的方法，测试场景涵盖从 5 秒视频中的单次切换到 60 秒内切换六个提示词。对于单属性切换，残差引导将平均切换达成率从 21% 提升至 58%；对于三事件故事，从 47% 提升至 66%；对于长时交互生成，从 66% 提升至 81%，同时基本保持了视觉质量。这些提升在所有基线方法上均成立，并且在连续切换中持续保持。在一项用户研究中，残差引导获得了 70.7% 的总体偏好得分。在单次切换场景中，残差引导仅使吞吐量降低 6%，因此适用于实时生成。

我们的贡献总结如下：(1) 我们刻画了流式视频生成中的上下文惯性，并指出，抑制新提示词的原因是缓存的视觉内容，而非过时的提示信息；这种抑制在第一个去噪步骤中尤为明显。(2) 我们提出残差引导，这是一种简单的免训练方法。它通过在不使用缓存的情况下衡量新提示词的影响，并将其应用于上下文惯性最强之处，从而在预测中而非缓存中抵消上下文惯性。(3) 我们在七种流式基线方法和多种切换场景中证明了该方法的有效性，在质量和速度仅付出极小代价的情况下，持续提升了提示切换响应能力。

2. 相关工作

实时视频生成中的提示切换。 最近的流式模型将双向教师模型蒸馏为少步数的因果学生模型 [9, 19, 26, 34, 56, 60, 63]，通过 KV缓存 [41]使每个块以前面生成的内容为条件。为处理提示切换，LongLive [56]将失败归因于缓存中过时的提示信息，并在新提示下重新编码过去的帧；而 Infinity-RoPE [59] 仅保留一个汇点和最后一帧，Anchor Forcing [57] 则利用存储的锚点状态重建缓存。MemFlow [22]和 SWIFT [47] 则分别通过检索与提示相关的历史信息以及注入感知转场的信号来管理记忆。尽管这些方法采用了不同的记忆策略，但它们都会在缓存中保留先前生成的视觉内容，而我们发现，这些内容仍会持续抑制新提示

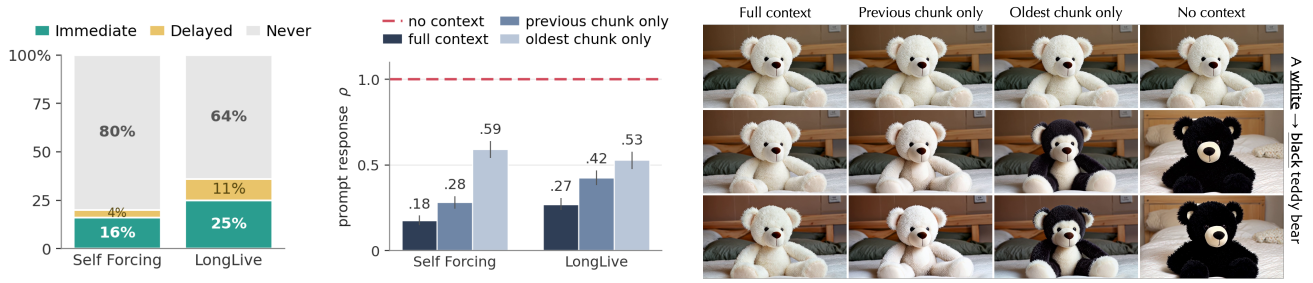


图 2。**上下文惯性下的提示切换**。(a) 100 次提示切换的结果，其中大多数从未实现。(b) 不同可见上下文下的相对提示响应 ρ。即使只有一个缓存块，也会抑制提示响应。(c) 同一切换在各个上下文（列）下生成的块。使用完整上下文或前一块上下文时，白熊仍然存在；仅使用最早的块时，白熊部分变暗；不使用上下文时，白熊变为黑色，但会牺牲身份和背景。

（第 3节）。残差引导则在预测阶段而非缓存中抵消这种抑制，从而保持连续性，并与这些方法兼容。

视频生成中的引导。无分类器引导（CFG） [17] 通过将条件预测沿远离无条件预测的方向外推来增强条件控制，已广泛用于视频扩散模型 [13, 51, 58]。然而，流式学生模型是在固定引导尺度下进行蒸馏的 [60], ，在这种蒸馏引导之上应用 CFG 只能部分恢复提示响应性。其两个分支都在相同缓存下计算，因此引导方向会与提示一同受到抑制，而更大的尺度会引入伪影（补充材料 F.2）。历史引导 [45] 利用逐帧噪声级别训练，将不同历史条件下的预测结合起来，以提升时间一致性，而非提示响应性。MAGI-1 [49]使用独立的历史引导尺度和文本引导尺度，但只有其非蒸馏模型提供了可独立调节的文本引导。相比之下，RG 使用已生成的块测量无缓存预测，因此其引导方向不会受到上下文抑制，并且不需要无条件分支。补充材料中给出了详细比较。F.1。

基于扩散的编辑。 许多基于扩散和基于流的编辑方法从源目标预测差异中推导编辑信号。Delta 去噪分数 [16]和对比去噪分数 [38] 使用去噪预测差异，而 DreamMotion [21] 使用带有时空自相似性约束的分数蒸馏。Delta 整流流采样 [4] 将基于增量的引导扩展到速度场。FlowEdit [28]在源分布和目标分布之间进行传输，FlowDirector [29] 使用基于 ODE 的流校正。ProxEdit [14] 对源-目标差异进行正则化，而 TurboEdit [10] 和 ChordEdit [35]分别实现少步和一步图像编辑。StreamEdit [23] 同样建立在无需训练的少步流式视频生成器之上，但在编辑源视频时采用条件化双分支采样器的方式；而残差引导没有源视频，并抵消正在进行的生成的缓存上下文。SDEdit [36]只是对输入重新加噪并加以重建，以进行图像编辑。残差引导也会对前一个块重新加噪，但会将其无缓存重建与原始块之间的差异应用到下一个块，而无需先前提示或反演。

3. 上下文惯性

记号。 流式模型以自回归方式生成视频，每次生成一个或多个潜在帧，这些帧被称为块。我们将块 k 的干净潜变量记为 s_k，将其在噪声水平 t 下的含噪状态记为 $x_k^t = \alpha_t s_k + \sigma_t \epsilon$。块 k通过少量步骤 $t_1 > t_2 > \cdots > t_Q$ （通常为$Q = 4$）由学生去噪器 $\hat{x}_\theta$ 进行去噪；该去噪器根据其含噪状态预测干净块，并以提示词 y_k 和因果键值缓存 $\mathcal{C}_{<k}$ 为条件，后者存储了从前一个块 s_1、……、s_{k-1}计算得到的键值状态。我们将这一预测记为 $\hat{x}_\theta(x_k^t \mid \mathcal{C}_{<k}, y_k)$；对于固定的噪声输入，也可简写为 $\hat{x}(\mathcal{C}, y)$。当提示词从 $y_{\text{old}} = y_{k-1}$ 变为 $y_{\text{new}} = y_k \neq y_{\text{old}}$时，块 k 处会发生提示切换。我们将块 k 称为切换块，因此 s_{k-1} 是根据 y_{old}生成的最后一个块。

3.1. 提示切换很少得到体现

我们分析了 Self Forcing [19]，这一具有代表性的因果流式学生模型，它在切换过程中保留滚动 KV缓存；以及 LongLive [56]，这一交互式模型，它会在提示词边界处根据新提示词重新计算缓存状态。我们构建了一个包含 100 个切换对 (y_{old}, y_{new}) 的开发集，并使用大语言模型 [39]，涵盖多种变化（如颜色和运动），其中 y_{old} 和 y_{new} 描述变化前后的状态。对于每一对，我们生成一个时长为 10 秒、包含 14 个块的视频，并在第 5 秒进行切换，留出 7 个块来体现新提示词。随后，视觉语言模型 [1]判断每个切换后块是否体现了 y_{new}。我们将切换归类为立即实现，如果 y_{new} 在切换后的前 2 个块内（约 1.5 秒，允许一个块用于

平滑过渡），如果在后续块中出现，则归类为延迟实现；如果在视频中未出现，则归类为从未实现。如图所示，图2a显示，Self Forcing 和 LongLive 的切换中，分别只有 16% 和 25% 是立即实现的，而 80% 和 64% 从未实现。补充材料A.1 详细介绍了提示词的构建和视觉语言模型评估，补充材料A.2则按提示词变化类型报告结果。

3.2. 为什么提示切换会失败?

为了分离提示词和上下文的影响，我们让模型在切换块 k 之前保持不变，并在每个去噪步骤固定其噪声输入 x_k^t。基于这个固定输入，我们计算可见上下文 $\mathcal{C} \subseteq \mathcal{C}_{<k}$下的 $\hat{x}(\mathcal{C},\ y)$，其中通过隐藏自注意力中的缓存键来限制上下文，$\mathcal{C} = \varnothing$则会隐藏所有缓存键。我们将提示响应衡量为从 y_{old} 切换到 y_{new}所引起的位移，并以无上下文时的相同位移进行归一化：

$$\rho(\mathcal{C}) = \frac{\|\hat{x}(\mathcal{C}, y_{\text{new}}) - \hat{x}(\mathcal{C}, y_{\text{old}})\|}{\|\hat{x}(\varnothing, y_{\text{new}}) - \hat{x}(\varnothing, y_{\text{old}})\|}, \tag{1}$$

因此，$\rho(\mathcal{C})$表示在 $\mathcal{C}$下保留的无上下文提示响应的比例，其中 $\rho(\varnothing) = 1$。对于分子，由于缓存可能携带构建时所用提示词的信息，我们在 y_{new}下重新缓存 LongLive，以确保这种抑制不能归因于过时的提示信息。对于分母，$\mathcal{C} = \varnothing$重现了每个视频首个块的无上下文场景，从而使评估保持在分布内。

发现1：上下文会抑制提示词。 我们从定量和定性两个方面考察缓存上下文如何影响对新提示的响应。除非另有说明，我们报告的是 ρ 在第一个去噪步骤中的结果，下面将说明这一步受影响最大。如图2b所示，在第一个去噪步骤中，对于 Self Forcing 和 LongLive，完整上下文下的响应分别仅为无上下文时提示词响应的 0.18 和 0.27；缓存内容对新提示的这种抑制，我们称为上下文惯性。此外，仅使用前一块时，ρ 两个模型的数值分别仅为 0.28 和 0.42；即使仅使用最早的一块，也只有 0.59 和 0.53，远低于 1。因此，单个较近的块就造成了大部分抑制，而较远的帧也会产生影响。由于 LongLive 的测量是在重新缓存后进行的，这种抑制主要源于缓存的视觉内容本身，而非过时的提示信息。图2c中的生成块也呈现出相同趋势：在完整上下文或仅使用前一块时，白熊都会持续保持白色；仅使用最早的一块时，它也只是部分变暗。完全移除上下文后，才会生成黑熊，但这会牺牲其身份和背景，因为此时生成块完全由噪声和提示词决定。人们可能会认为，更强的无分类器引导能够抵消这种抑制。由于引导方向$\hat{x}(\mathcal{C},\ y_{\text{new}}) - \hat{x}(\mathcal{C},\ y_{\varnothing})$，其中 $y_{\varnothing}$ 表示空提示词，并且是在相同缓存条件下计算的，因此它也会受到抑制；增大引导尺度主要会放大伪影，而非恢复提示响应（补充材料 F.2）。

发现2：上下文惯性在步骤1最强。为了确定这种抑制出现于少步轨迹的哪个位置，我们在每个去噪步骤重复测量，发现两种不同的效应都在步骤1达到峰值：仅提示词对预测的影响最大，而上下文对这种变化的抑制也最强。首先，在没有上下文的情况下，将 y_{old} 切换为 y_{new}，$\|\hat{x}(\varnothing,\ y_{\text{new}}) - \hat{x}(\varnothing,\ y_{\text{old}})\| / \|s_{k-1}\|$所引起的切换块干净预测变化，对于 Self Forcing 在步骤1为0.62，但在步骤2仅为0.19，在步骤4仅为0.03。其次，在有上下文的情况下，这一响应中得以保留的比例 ρ在步骤1最低，而且在该步骤丢失的响应不会在后续步骤中恢复（详情请参见补充材料A.3 ）。我们将这两种现象都归因于扩散采样由粗到细的特性：早期的高噪声步骤主要决定布局和内容，而后续步骤则细化细节 [3, 7, 15], 这一效应在少步学生模型中可能会进一步加剧，因为其步骤1覆盖了大部分噪声范围。

4. 方法

第3节表明，遵循新提示需要在不使用缓存的情况下对其进行评估，而保持视觉连续性则需要使用缓存生成。残差引导（*RG*）通过在不使用缓存的情况下测量提示信号，同时使用缓存生成块，实现了二者兼顾。具体而言，残差引导（RG）在不使用缓存的情况下，根据新提示重建前一个块的重新加噪副本，并在早期去噪步骤中，将其相对于原始块的残差添加到下一个块的缓存条件预测中；在这些步骤中，上下文惯性最为显著。

4.1. 问题的后验视角

上述直觉可以得到简单的概率解释。学生模型的干净预测近似于给定其含噪状态 x_k^t、前一个缓存 $\mathcal{C}_{<k}$ 和提示词 y_{new} 时的后验均值 s_k。由于前向加噪过程不依赖于提示词（即，$x_k^t \perp y_{\text{new}} \mid s_k,\ \mathcal{C}_{<k}$），该后验可分解为

$$p(s_k \mid x_k^t, \mathcal{C}_{<k}, y_{\text{new}}) \ \propto\ p(s_k \mid x_k^t, \mathcal{C}_{<k})\, p(y_{\text{new}} \mid s_k, \mathcal{C}_{<k}). \tag{2}$$

第一项有利于延续缓存内容，第二项则有利于新提示。当前一项占主导地位时，就会产生上下文惯性。在切换时编辑缓存的方法 [56, 59] 会削弱第一项，而更强的文本引导会放大第二项；然而，这一项仍是在缓存条件下计算的，因此大部分抑制效果依然存在。残差引导则会加入一个在无缓存条件下测得的残差，将预测推向新提示

并远离缓存内容。这两种效果共同带来了增益（表3）。与此同时，由于该块仍是在完整缓存条件下生成的，因此与前一个块的视觉连续性基本得以保留。

4.2. 残差引导

残差估计。 为衡量新提示词在没有上下文惯性的情况下所产生的影响，首先将前一个块 s_{k-1}重新加噪至噪声水平 τ，然后在不使用KV缓存的情况下，依据 y_{new} 通过单步去噪得到结果：

$$\begin{aligned} \tilde{x}^{\tau}_{k-1} &= \alpha_\tau s_{k-1} + \sigma_\tau \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \\ r_k &= \hat{x}_\theta(\tilde{x}^{\tau}_{k-1} \mid \varnothing, y_{\text{new}}) - s_{k-1}. \end{aligned} \tag{3}$$

这里， $\varnothing$ 表示空的KV缓存，因此前一个块仅通过其重新加噪后的潜变量参与重建。该残差衡量在 y_{new} 条件下、不使用缓存进行重建所得结果与原始块之间的差异，其中一部分独立于 y_{new}，从而抵消前一个块内容本身的影响。因此，它并非对方程（2）中提示词项的精确估计，而是对缓存条件预测的近似校正（见补充材料 B.1）。噪声水平 τ独立于采样步骤 t_i进行选择，它决定重建保留前一个块内容的程度，进而决定校正的强度，其作用类似于无分类器引导中的引导尺度 [17]。

残差注入。 在块$k-1$ 上测得残差 r_k，并将其应用于块 k。上下文惯性本身就促使我们进行这种迁移。由于块 k 的缓存条件预测与 s_{k-1}接近，因此根据s_{k-1} 得出的校正也适用于该块。在早期去噪步骤中，将残差加到干净预测上：

$$\hat{x}^{(i)}_k = \hat{x}_\theta(x^{t_i}_k \mid \mathcal{C}_{<k}, y_{\text{new}}) + \lambda_i r_k, \tag{4}$$

在 $\lambda = (1、0.5、0、0)$ 这四个步骤中。由此，校正作用于抑制最强的位置（第 3节），其余步骤则在完整缓存下保持不变地运行，使该块与其历史保持一致。不同于无分类器引导在使用缓存时对比两次预测，残差是在不使用缓存的情况下测得的，因此不会受到缓存的削弱。

实现。 对于在 y_{new}下生成的每个块，都会根据其前一个块重新计算残差，直到下一次提示切换。我们为所有模型设置 $\tau = 850$，并在每次切换时对 ϵ 采样一次，在整个提示词区间内共享该结果（补充材料 D.6）。其他干预方式，包括针对空提示词或先前提示的残差、部分移除缓存以及更强的无分类器引导，只能带来较小的增益，或会降低视觉连续性（补充材料 F.3）。消融实验和替代方案见第 5.3节，完整算法见补充材料 B.2。

5. 实验

5.1. 设置

基线方法。 我们将残差引导应用于 Self Forcing [19], CausalForcing [63], 和 MemFlow [22], （它们在切换时保留缓存），应用于 LongLive [56]和 Infinity-RoPE [59],（它们会修改缓存），以及应用于 Anchor Forcing [57] 和 SWIFT [47], （它们使用专用的切换机制）。由于残差引导不会改动缓存，因此无论七种基线方法如何管理缓存，都可以接入其中，无需重新训练或针对每个模型进行调优，并且始终使用相同的配置。所有方法均从 Wan2.1-1.3B [51],生成 832×480 视频，并使用其原生采样方式。每个模型及其残差引导变体共享相同的随机种子，从而确保切换前的帧完全一致。我们还将方法应用于滚动强制，详见补充材料 [34] E.2。

数据集。 我们在持续时间和切换次数逐渐增加的三种设置中评估残差引导（RG），每种设置都探究提示切换的不同方面。首先，将 T2V-CompBench 的 200 个动态属性提示词 [46] 拆分为初始状态和最终状态，并生成 5 秒视频，在第 2 秒进行切换。由于只有同一主体的一个属性发生变化，该设置最直接地测试上下文惯性。其次，将 StoryEval 中的 100 个三事件故事 [53] 生成为 10 秒视频，并将其中的事件作为连续提示词，在第 3 秒和第 7 秒进行切换，以测试模型能否反复切换。第三，遵循 MemFlow [22] 及其他交互式方法 [47, 56, 57], ，我们使用 100 个包含六个提示词的脚本，每个提示词在 60 秒视频中持续生效 10 秒，我们将其称为 InteractiveBench。提示词构建的详细信息见补充材料 C.1。补充材料 E.1、E.3和 F.4 进一步涵盖动作绑定提示词（切换时会出现新的主体）、扩展至 240 秒的 InteractiveBench，以及使用 Infinity-RoPE [59]及其 RoPE Cut 的显式场景切换。

指标。 视觉语言模型评判器 [1] 判断切换后视频是否呈现新提示词（可靠性见补充材料C.3）。 达成度 是视频呈现新提示词的切换次数占总切换次数的比例。我们排除模型即使将其用作初始提示词也无法生成的新提示词，以确保失败反映的是切换问题，而非提示词问题。对于切换后片段，我们报告其与新提示词的 CLIP 相似度 [42] 和 VBench [20]质量分数，该分数按照官方归一化方法和权重汇总七个质量维度。我们还报告 VBench 动态程度，以确保一致性并非仅通过生成静态视频实现。遵循 VBench 总分的计算方式，总体 计算如下$(4 \cdot$ 质量 $+$达成度$)/5$，其中以达成度替代语义分数。对于动态属性，我们还报告官方 T2V-CompBench 分数 [46]。指标定义详见补充材料C.2。

表 1. **T2V-CompBench 动态属性上的提示切换**。 达成度：反映新提示词的视频占比。 T2V-CB：官方基准测试分数 [46]。质量和总体遵循 VBench [20]，，并以达成度作为语义分数。

方法	达成度 ↑	T2V-CB 动态属性 ↑	CLIP ↑	动态程度 ↑	质量 ↑	总体 ↑
Self Forcing	0.106	0.013	24.97	0.250	0.813	0.672
+RG	0.568	0.119	26.29	0.380	0.806	0.758
Causal Forcing	0.147	0.024	24.78	0.425	**0.821**	0.686
+RG	0.558	0.085	25.68	**0.600**	0.813	0.762
MemFlow	0.210	0.035	25.55	0.175	0.805	0.686
+RG	0.493	0.091	26.41	0.235	0.802	0.740
LongLive	0.296	0.039	25.46	0.145	0.805	0.703
+RG	0.622	0.145	26.82	0.215	0.802	0.766
Infinity-RoPE	0.148	0.023	24.75	0.155	0.806	0.674
+RG	0.578	0.110	25.99	0.310	0.803	0.758
Anchor Forcing	0.328	0.058	25.61	0.365	0.811	0.714
+RG	0.606	**0.152**	26.45	0.540	0.813	0.772
SWIFT	0.252	0.028	25.32	0.120	0.806	0.695
+RG	**0.667**	0.144	**26.86**	0.150	0.799	**0.773**

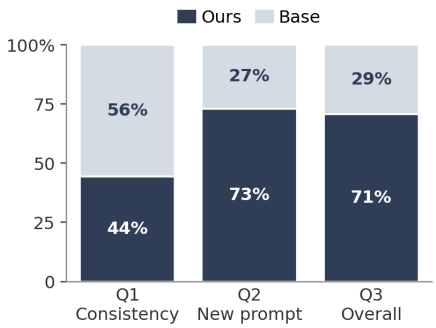


图3. **用户研究**。 20 位评审者针对同一提示词对对+残差引导 的偏好（%），评估维度包括一致性（除所请求的变化外没有非预期变化）、对新提示的遵循程度以及总体偏好。残差引导在遵循程度和总体偏好方面广受青睐，而一致性略低于平均水平。

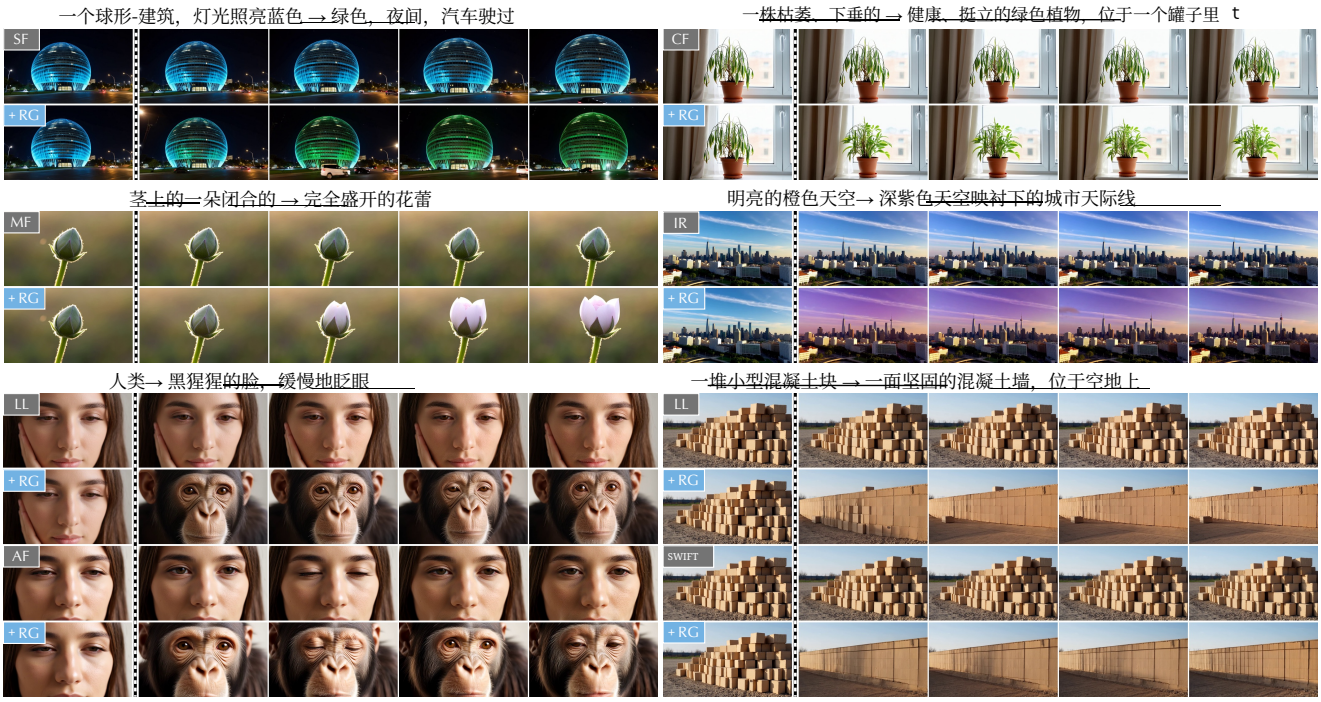


图4。**T2V-CompBench 上的定性结果**。白色至黑色的条带标志着提示切换。没有残差引导（RG）时，视频在切换后仍保持初始状态；而残差引导（RG）能够让新状态出现，同时保持主体和场景布局一致。SF：Self Forcing，CF：Causal Forcing，MF：MemFlow，IR：Infinity-RoPE，LL：LongLive，AF：Anchor Forcing。

5.2 主要结果

单次切换。 表。 1 显示，上下文惯性严重限制了现有模型，尤其是在仅要求改变单个属性时。不使用残差引导时，Self Forcing、Causal Forcing、Mem-Flow 和 Infinity-RoPE 的达成度最高仅为 0.21，而 LongLive、Anchor Forcing 和 SWIFT 的达成度为 0.25 至 0.33。残差引导使每个基础模型的达成度提升了 0.28 至 0.46，例如 Self Forcing 从 0.106 提升至 0.568。这些提升并非特定于视觉语言模型评判器、CLIP 和官方 T2V-CompBench 分数（它们会评估提示词的两种状态）在所有基础模型上也都有所提升。RG 与 LongLive、Anchor Forcing 和 SWIFT 结合时取得了最高分，这表明 RG 能够补充现有的切换机制，而非取代它们。这种提升的代价很小，因为质量与基础模型的差距始终不超过 0.01。大多数模型的质量略有下降，这反映了已知的一种权衡：一致性指标与动态程度之间存在取舍 [20]，所有模型的动态程度都会上升，

表2。 **多提示词和更长的场景设置**。 StoryEval：在10秒内使用100个包含三个事件的提示词，并进行切换。InteractiveBench：按照 MemFlow [22] 和 LongLive [56]，在60秒内使用100个由六个10秒提示词组成的脚本。标记为 + 残差引导 的行会对上方模型应用残差引导。

方法	StoryEval（10 秒） 成就 ↑	CLIP ↑	动态程度 ↑	质量 ↑	总体 ↑	InteractiveBench（60 秒） 成就 ↑	CLIP ↑ 10–20	20–30	30–40	40–50	50–60	动态程度 ↑	质量 ↑	总体 ↑
MemFlow	0.385	22.41	0.625	**0.830**	0.741	0.587	27.04	27.38	26.98	26.89	26.38	0.254	0.810	0.766
+ RG	0.659	23.01	0.750	0.826	0.793	0.724	27.79	27.92	27.49	27.73	27.39	0.302	0.806	0.790
LongLive	0.511	22.31	0.515	0.824	0.762	0.631	27.35	27.73	27.58	27.78	26.81	0.284	0.815	0.778
+ RG	0.687	**23.13**	0.680	0.823	**0.795**	0.826	**28.03**	**28.38**	27.73	**28.19**	**27.49**	0.340	0.809	0.812
Infinity-RoPE	0.402	21.87	0.530	0.821	0.738	0.609	26.83	27.19	26.85	26.65	26.30	0.388	0.824	0.781
+ RG	0.586	22.58	0.765	0.819	0.772	0.783	27.10	27.55	27.16	27.03	27.07	0.560	0.827	0.818
Anchor Forcing	0.538	22.59	0.895	0.821	0.764	0.765	26.98	27.12	27.03	26.75	26.62	0.796	0.831	0.818
+ RG	**0.732**	23.11	**0.965**	0.802	0.788	**0.871**	27.32	27.54	27.41	27.25	27.43	**0.886**	**0.832**	**0.840**
SWIFT	0.505	22.23	0.465	0.823	0.759	0.693	27.54	27.90	27.51	27.71	26.84	0.306	0.815	0.791
+ RG	0.654	23.08	0.645	0.821	0.788	0.823	27.86	28.17	**27.82**	27.82	27.42	0.394	0.811	0.813



(a) StoryEval



(b) InteractiveBench

图 5. **多事件生成的定性结果**。 每个示例中，顶排为基础模型，底排为应用残差引导（RG）的同一模型。虚线表示提示切换。(a) 三事件故事（10 秒）。基础模型始终停留在第一个事件，而残差引导会按照提示推进到每个事件。(b) 包含六条提示词的 60 秒交互序列。残差引导依次遵循每条指令，包括新主体的出现与互动。

属性变化本身会引入运动（更多讨论见补充材料D.1）。因此，总体表现都有所提升，最高的两个分数分别来自结合 RG 的 SWIFT 和 Anchor Forcing。图4 显示，使用 RG 时，所要求的属性会在切换后迅速显现（例如，花蕾绽放、人脸变成黑猩猩），同时主体和布局基本得以保留（见补充材料D.2）；而基础模型大多仍停留在初始状态。

多次切换。表。2（左）评估了 StoryEval 三事件故事上的短多次切换生成。此处的基础成就度高于动态属性任务，因为动作或场景的变化比同一主体上的属性变化更容易呈现。残差引导（RG）仍使每个基础模型的达成度提高 0.15 至 0.27，例如 MemFlow 从 0.385 提高到 0.659，并且——在所有模型上提升了 CLIP 和动态程度，同时将质量保持在基础模型的 0.02 以内。因此，总体表现全面提升，且 LongLive 和 MemFlow 搭配 RG 时最高。图5a 显示，RG 能依次实现两次切换（例如，行李箱拉链拉上，气泡爆裂成细小的星星），而基础模型几乎无法推进到第一个事件之后。

长时交互生成。表。2 （右）评估 InteractiveBench 上的长时多切换生成，该基准遵循先前交互方法的 60 秒协议 [22,56]，，以便在各自的场景下进行测试。RG 在每个基础模型上都将达成度提高了 0.11 至 0.20，例如在 LongLive 上从 0.631 提高到 0.826。提升并不局限于初期切换：首次切换后的每个 10 秒片段（包括最后一个片段）的 CLIP 得分都有所提高。定性-



图6. **噪声水平和步数调度**。 上：使用我们的调度时的噪声水平 τ $\lambda = (1, 0.5, 0, 0)$。下：在列出的步骤中完整添加的残差（$\lambda_i = 1$），其中 $\tau = 850$。切换前的帧（黄色霓虹灯牌）显示在上方。

结果在动态程度提高的同时，与基础模型的差距保持在 0.006 以内，因此所有模型的总体表现均有所提升，其中 Anchor Forcing 与 RG 结合时最高。图 5b 显示，LongLive 仅部分遵循切换（例如，酿酒师拿着葡萄，但儿子并未出现），而 RG 能很好地满足每条提示词。VBench-Long 的各维度细分以及更多定性结果见补充材料 D.4和 D.3。

用户研究。为评估用户如何看待这些提升，20 位评审者在相同提示对下，对比每个基础模型使用和不使用残差引导的效果（图3）。对各基础模型取平均后，在 73.1% 的比较中，评审者更倾向于选择残差引导来遵循新提示，而一致性略低于持平水平，为 44.3%。当被问及实时切换提示词时更倾向于哪种结果，评审者有 70.7% 的时间选择残差引导，表明在实际使用中，响应性的提升足以弥补一致性的小幅损失（补充材料C.4）。

计算开销。残差引导会为每个切换后块增加一次无缓存的前向传播。在单张 RTX A6000 上，对于包含一次切换的 5 秒视频，残差引导会使吞吐量降低约 6%。与此同时，切换生效更快：在 LongLive 上，两块内实现切换的比例从 0.21 提升至 0.54。详情见补充材料 D.5 ；关于切换块本身，请参阅补充材料D.2 。

5.3. 消融实验与替代方案

消融实验。 图 6展示了两个主要的设计选择。噪声水平 τ 控制重建保留前一个块内容的程度，其作用类似于引导尺度，是一个强度旋钮：在 $\tau = 700$时，标志仍为黄色；在 $\tau = 1000$时，它会像无上下文预测（图 2c）一样以新的形状重新绘制；而在 $\tau = 850$时，其颜色会发生变化，但形状保持不变。步骤调度在校正与细化之间取得平衡。加入

表3. **残差引导的替代方案**。 在 Self Forcing 和 LongLive 上对 100 组随机动态属性对进行评估。非预期变化：意外变化，通过切换前后变更对象被遮挡时的 DreamSim [12] 距离测量。

	Self Forcing			LongLive		
方法	成就↑	质量↑	非预期变化↓	成就↑	质量↑	非预期变化↓
基础模型	0.130	0.789	0.026	0.350	0.770	0.021
无上下文（全部）	1.000	0.726	0.191	0.963	0.733	0.163
无上下文（步骤1）	0.948	0.803	0.165	0.863	0.787	0.114
CFG（w=3）	0.234	0.793	0.033	0.550	0.772	0.033
仅文本残差	0.351	0.788	0.037	0.525	0.772	0.032
残差引导（我们的方法）	0.623	0.788	0.089	0.675	0.778	0.046

仅在步骤1加入残差会产生部分切换，因为下一步会将该块拉回缓存上下文。若在步骤2也完整加入残差， $\lambda = (1, 1, 0, 0)$，则会抵消这种拉回作用，但也会覆盖从上下文继承的形状；将其延伸至步骤3则会产生伪影。在步骤2将权重减半， $\lambda = (1, 0。5, 0, 0)$，既能防止拉回，又能让缓存恢复形状，从而将后续步骤留给细化。定量结果见补充材料 D.6。

替代方案。 表。 3 将残差引导与更简单的干预措施进行了比较。仅在第一步移除上下文，就能实现大多数切换，但造成的非预期变化是残差引导的 1.9 至 3.5 倍。CFG 保持了视觉连续性，但达成度仍低于残差引导，因为上下文也抑制了其提示方向，而且每个引导步骤都需要额外进行一次前向传播。仅保留文本的残差会去掉与提示无关部分 $\hat{}x_\theta(\tilde{x}^\tau_{k-1} \mid \varnothing, y_\varnothing) - s_{k-1}$ 自 r_k，只能恢复约一半的增益，这表明，消除上下文施加的影响与强化提示同样重要。更多替代方案见补充材料F.3。

6. 结论

在这项工作中，我们将流式视频生成中的提示切换失败归因于上下文惯性：缓存的视觉内容会削弱新提示的影响，而现有的基于缓存的切换策略无法完全消除此抑制。残差引导（*RG*）通过在不使用缓存的情况下测量新提示的效果，并在早期去噪步骤中将由此得到的残差加入下一个块，来解决这一问题；上下文惯性在这些步骤中最为显著。RG 无需训练，可与现有切换方法结合使用，并且每个块仅增加一次前向传播，因此适用于实时使用。在七种流式基线方法以及从单次切换到长视频生成中的多次切换等不同场景下，RG 显著提升了切换达成率，同时保持视觉连续性。不过，有些变化仍然难以实现，例如移除物体；而且细节也可能随着请求的变化一同改变。我们在补充材料F.5 中讨论这些失败案例，并将其作为未来工作的方向。

参考

[1] Shuai Bai、Yuxuan Cai、Ruizhe Chen、Keqin Chen、Xionghui Chen、Zesen Cheng、Lianghao Deng、Wei Ding、Chang Gao、Chunjiang Ge 等。Qwen3-vl 技术报告。arXiv 预印本 arXiv:2511.21631，2025。3，5，1，4[2] Xingjian Bai、Guande He、Zhengqi Li、Eli Shechtman、Xun Huang 和 Zongze Wu。视频扩散器中的因果性与去噪是可分离的。arXiv 预印本 arXiv:2602 10095，2026 14 。.[3] Yogesh Balaji、Seungjun Nah、Xun Huang、Arash Vahdat、Jia-ming Song、Qinsheng Zhang、Karsten Kreis、Miika Aittala、Timo Aila、Samuli Laine 等。ediff-i：采用专家去噪器集成的文本到图像扩散模型。arXiv 预印本 arXiv:2211.01324，2022。4[4] Gaspard Beaudouin、Minghan Li、Jaeyeon Kim、Sung-Hoon Yoon 和 Mengyu Wang。用于文本到图像编辑的 Delta 整流流采样。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 18662–18672 页，2026。3[5] Mathilde Caron、Hugo Touvron、Ishan Misra、Herv´e J´egou、Julien Mairal、Piotr Bojanowski 和 Armand Joulin。自监督视觉变换器中的新兴特性。载于 2021 IEEE/CVF国际计算机视觉会议（ICCV），第 9630–9640 页。IEEE，2021。4[6] Boyuan Chen、Diego Mart´ı Mons´o、Yilun Du、Max Sim-chowitz、Russ Tedrake 和 Vincent Sitzmann。扩散强制：下一词元预测与全序列扩散相结合。神经信息处理系统进展，37：24081–24125，2024。2[7] Jooyoung Choi、Jungbeom Lee、Chaehun Shin、Sungwon Kim、Hyunwoo Kim 和 Sungroh Yoon。以感知为优先的扩散模型训练。载于 2022年IEEE/CVF计算机视觉与模式识别会议（CVPR），第 11462–11471 页。IEEE，2022。4[8] Jacob Cohen。名义量表的一致性系数。《教育与心理测量》，20 46，1960 1 37 1，5 ( ): – 。[9] Jiaxing Cui、Jie Wu、Ming Li、Tao Yang、Xiaojie Li、Rui Wang、Andrew Bai、Yuanhao Ban 和 Cho-Jui Hsieh。自强制++：迈向分钟级高质量视频生成。载于国际学习表征会议，第 85802–85822 页，2026。2[10] Gilad Deutch、Rinon Gal、Daniel Garibi、Or Patashnik 和 Daniel Cohen-Or。TurboEdit：使用少步扩散模型进行基于文本的图像编辑。载于 SIGGRAPH Asia 2024 会议论文集，第 1–12 页，2024。3[11] Bradley Efron。Tweedie公式与选择偏差。《美国统计学会杂志》，106(496)：1602–1614，2011。3[12] Stephanie Fu、Netanel Tamir、Shobhita Sundaram、Lucy Chai、Richard Zhang、Tali Dekel 和 Phillip Isola。DreamSim：利用合成数据学习人类视觉相似性的新维度。arXiv 预印本 arXiv:2306 09344，2023 8，7 。.[13] Yoav HaCohen、Nisan Chiprut、Benny Brazowski、Daniel Shalem、Dudu Moshe、Eitan Richardson、Eran Levin、Guy Shiran、Nir Zabari、Ori Gordon 等。LTX-Video：实时视频潜扩散。arXiv 预印本 arXiv:2501 00103，2024 2, 3 . .[14] Ligong Han 、 Song Wen 、 Qi Chen 、 Zhixing Zhang、Kunpeng Song、Mengwei Ren、Ruijiang Gao、Anastasis Stathopou- los、Xiaoxiao He、Yuxiao Chen 等。ProxEdit：利用近端引导改进免调优真实图像编辑。载于《2024年IEEE/CVF冬季计算机视觉应用会议（WACV）论文集》，第 4279–4289 页。IEEE，2024。3[15] Amir Hertz、Ron Mokady、Jay Tenenbaum、Kfir Aberman、Yael Pritch 和 Daniel Cohen-Or。利用交叉注意力控制实现 Prompt-to-Prompt 图像编辑。arXiv 预印本 arXiv:2208.01626，2022。4[16] Amir Hertz、Kfir Aberman 和 Daniel Cohen-Or。Delta 去噪分数。载于《2023年 IEEE/CVF 国际计算机视觉会议（ICCV）论文集》，第 2328–2337 页。IEEE，2023。3[17] Jonathan Ho 和 Tim Salimans。无分类器扩散引导。arXiv 预印本 arXiv:2207.12598，2022。2, 3, 5, 14, 15[18] Jonathan Ho、William Chan、Chitwan Saharia、Jay Whang、Ruiqi Gao、Alexey Gritsenko、Diederik P Kingma、Ben Poole、Mohammad Norouzi、David J Fleet 等。Imagen 视频：利用扩散模型生成高清视频。arXiv 预印本 arXiv:2210.02303，2022。2[19] Xun Huang、Zhengqi Li、Guande He、Mingyuan Zhou 和 Eli Shechtman。Self Forcing：弥合自回归视频扩散中的训练—测试差距。《神经信息处理系统进展》，38:167283–167308，2025。2, 3, 5[20] Ziqi Huang、Yinan He、Jiashuo Yu、Fan Zhang、Chenyang Si、Yuming Ji ang、Yuanhan Zhang、Tianxing Wu、Qingyang Jin、Nattapol Chanpaisit、Yaohui Wang、Xinyuan Chen、Limin Wang、Dahua Lin、Yu Qiao 和 Ziwei Liu。VBench：视频生成模型综合基准套件。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，2024。5, 6, 4[21] Hyeonho Jeong、Jinho Chang、Geon Yeong Park 和 Jong Chul Ye。DreamMotion：用于零样本视频编辑的时空自相似分数蒸馏。载于《欧洲计算机视觉会议论文集》，第 358–376 页。Springer，2024。3[22] Sihui Ji、Xi Chen、Shuai Yang、Xin Tao、Pengfei Wan 和 Hengshuang Zhao。MemFlow：流动的自适应记忆，实现连贯且高效的长视频叙事。arXiv 预印本 arXiv:2512.14699，2025。2, 5, 7, 4, 12[23] Guanlong Jiao、Chenyangguang Zhang、Jia Jun Cheng Xian、Zewei Zhang 和 Renjie Liao。Streamdit：通过少步流式视频生成实现免训练视频编辑。载于《欧洲计算机视觉会议论文集》。Springer，2026。3[24] Junjie Ke、Qifei Wang、Yilin Wang、Peyman Milanfar 和 Feng Yang。Musiq：多尺度图像质量变换器。载于《2021年IEEE/CVF国际计算机视觉大会（ICCV）论文集》，第 5128–5137 页。IEEE，2021。4[25] Lei Ke、Mingqiao Ye、Martin Danelljan、Yu-Wing Tai、Chi-Keung Tang、Fisher Yu 等。高质量的万物分割。《神经信息处理系统进展》，36:29914–29934，2023。7

[26] Akio Kodaira、Tingbo Hou、Ji Hou、Markos Georgopou- los、Felix Juefei-Xu、Masayoshi Tomizuka 和 Yue Zhao。Streamdit：实时流式文本到视频生成。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 29200–29210 页，2026。2[27] Weijie Kong、Qi Tian、Zijian Zhang、Rox Min、Zuozhuo Dai、Jin Zhou、Jiangfeng Xiong、Xin Li、Bo Wu、Jianwei Zhang 等。Hunyuanvideo：大规模视频生成模型的系统性框架。arXiv 预印本 arXiv:2412.03603，2024。2[28]Vladimir Kulikov、Matan Kleiner、Inbar Huberman- Spiegelglas 和 Tomer Michaeli。FlowEdit：利用预训练流模型进行无反演文本编辑。载于《2025年 IEEE/CVF 国际计算机视觉会议（ICCV）论文集》，第 19721–19730 页。IEEE，2025。3[29] Guangzhao Li、Yanming Yang、Chenxi Song、Xiaohong Liu 和 Chi Zhang。FlowDirector：用于精确文本到视频编辑的免训练流引导。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 7805–7815 页，2026。3[30] Zhen Li、Zuo-Liang Zhu、Ling-Hao Han、Qibin Hou、Chun-Le Guo 和 Ming-Ming Cheng。AMT：用于高效帧插值的全对多场变换。载于《2023年IEEE/CVF计算机视觉与模式识别会议（CVPR）论文集》，第 9801–9810 页。IEEE，2023。4[31] Bin Lin 、 Yunyang Ge 、Xinhua Cheng、Zongjian Li、Bin Zhu、Shaodong Wang、Xianyi He、Yang Ye、Shenghai Yuan、Li-uhan Chen 等。Open-Sora Plan：开源大规模视频生成模型。arXiv 预印本 arXiv:2412.00131，2024。2[32] Haotian Liu、Chunyuan Li、Qingyang Wu 和 Yong Jae Lee。视觉指令微调。《神经信息处理系统进展》，36:34892–34916，2023。4, 5[33] Jinzhuo Liu、Jiangning Zhang、Wencan Jiang、Yabiao Wang、Dingkang Liang、Zhucun Xue、Ran Yi 和 Yong Liu。通过免训练身份感知记忆推进叙事长视频生成。arXiv 预印本 arXiv:2605 18733，2026 2 . .[34] Kunhao Liu、Wenbo Hu、Jiale Xu、Ying Shan 和 Shijian Lu。Rolling Forcing：实时自回归长视频扩散。载于《国际表征学习会议论文集》，第 91177–91196 页，2026。2, 5, 12, 13[35] Liangsi Lu、Xuhang Chen、Minzhe Guo、Shichu Li、Jingchao Wang 和 Yang Shi。ChordEdit：用于图像编辑的一步低能量传输。arXiv 预印本 arXiv:2602 19083，2026 3 . .[36] Chenlin Meng、Yutong He、Yang Song、Jiaming Song、Jiajun Wu、Jun-Yan Zhu 和 Stefano Ermon。SDEdit：利用随机微分方程进行引导式图像合成与编辑。arXiv 预印本 arXiv:2108.01073，2021。3[37] Matthias Minderer、Alexey Gritsenko 和 Neil Houlsby。扩展开放词汇目标检测。《神经信息处理系统进展》，36:72983–73007，2023。7[38] Hyelin Nam、Gihyun Kwon、Geon Yeong Park 和 Jong Chul Ye。用于文本引导的潜在扩散图像编辑的对比去噪分数。载于《2024年IEEE/CVF大会——计算机视觉与模式识别会议（CVPR），页码 9192 9201 2024 3 – 。IEEE， 。 OpenAI。[39] 5 5《GPT- . 系统卡》，2026年。 3，1[40] Adam Polyak、Amit Zohar、Andrew Brown、Andros Tjandra、Animesh Sinha、Ann Lee、Apoorv Vyas、Bowen Shi、Chih-Yao Ma、Ching-Yao Chuang 等。Movie gen：一组媒体基础模型。arXiv 预印本 arXiv:2410 13720，2024年 2。 。[41] Reiner Pope、Sholto Douglas、Aakanksha Chowdhery、Jacob Devlin、James Bradbury、Jonathan Heek、Kefan Xiao、Shiv-ani Agrawal 和 Jeff Dean。高效扩展 Transformer 推理。机器学习与系统会议论文集，5：606–624，2023年。 2[42] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。从自然语言监督中学习可迁移视觉模型。载于国际机器学习会议，页码 8748–8763。PmLR，2021年。 5，4[43] Zejing Rao、Ketong Ren、Xiaoqiang Liu、Yiping Meng、Guoxin Zhang 和 Fan Tang。毫不走捷径：视频生成中流中提示切换的状态约束转换。arXiv 预印本 arXiv:2609.38691，2026年。 2[44] Christoph Schuhmann、Romain Beaumont、Richard Vencu、Cade Gordon、Ross Wightman、Mehdi Cherti、Theo Coombes、Aarush Katta、Clayton Mullis、Mitchell Worts-man 等。LAION-5B：用于训练下一代图文模型的开放大规模数据集。神经信息处理系统进展，35：25278–25294，2022年。 4[45] Kiwhan Song、Boyuan Chen、Max Simchowitz、Yilun Du、Russ Tedrake 和 Vincent Sitzmann。历史引导视频扩散。arXiv 预印本 arXiv:2502.06764，2025年。 3，14，15[46] Kaiyue Sun、Kaiyi Huang、Xian Liu、Yue Wu、Zihan Xu、Zhenguo Li 和 Xihui Liu。T2V-CompBench：组合式文本生成视频的综合基准测试。载于 2025 IEEE/CVF计算机视觉与模式识别会议（CVPR），页码 8406–8416。IEEE，2025年。 5，6，4[47] Shanwen Tan、Hao Li、Jingtao Zhang、Xiaosong Jia、Xue Yang、Shaofeng Zhang 和 Yanyong Zhang。SWIFT：用于高效交互式长视频生成的提示自适应记忆。arXiv 预印本 arXiv:2605.09442，2026年。 2，5，14[48] Zachary Teed 和 Jia Deng。RAFT：用于光流的循环全对场变换。载于欧洲计算机视觉会议，页码 402–419。Springer，2020年。 4[49] Hansi Teng、Hongyu Jia、Lei Sun、Lingzhi Li、Maolin Li、Mingqiu Tang、Shuai Han、Tianning Zhang、WQ Zhang、Weifeng Luo 等。MAGI-1：大规模自回归视频生成。arXiv 预印本 arXiv:2505.13211，2025年。 3，14，15[50] Narek Tumanyan、Omer Bar-Tal、Shai Bagon 和 Tali Dekel。拼接 ViT 特征进行语义外观迁移。载于 2022 IEEE/CVF计算机视觉与模式识别会议（CVPR），页码 10738–10747。IEEE，2022年。 7[51] Team Wan、Ang Wang、Baole Ai、Bin Wen、Chaojie Mao、Chen-Wei Xie、Di Chen、Feiwu Yu、Haiming Zhao、Jianxiao

Yang 等。Wan：开放且先进的大规模视频生成模型。arXiv 预印本 arXiv:2503.20314，2025年。 2，3，5[52] Weiyun Wang、Zhangwei Gao、Lixin Gu、Hengjun Pu、Long Cui、Xingguang Wei、Zhaoyang Liu、Linglin Jing、Shenglong Ye、Jie Shao 等。InternVL3.5：在通用性、推理能力和效率方面推进开源多模态模型。arXiv 预印本 arXiv:2508.18265，2025年。 5[53] Yiping Wang、Xuehai He、Kuan Wang、Luyao Ma、Jianwei Yang、Shuohang Wang、Simon Shaolei Du 和 Yelong Shen。你的世界模拟器是优秀的故事讲述者吗？面向未来长视频生成的连续事件基准测试。arXiv 预印本 arXiv:2412.16211，2024年。 5，4[54] Mingqiang Wu、Weilun Feng、Zhefeng Zhang、Haotong Qin、Yuqi Li、Guoxin Fan、Xiaokun Liu、Zhulin An、Libo Huang、Yongjun Xu 等。Echo-Forcing：用于交互式长视频生成的场景记忆框架。arXiv 预印本 arXiv:2605.16003，2026年。 14[55] Lingxiao Yang、Liu Liu、Moran Li、Han Feng、Wenjian Cao、Jiangning Zhang 和 Ye Shi。In-Context Forcing：揭示自回归视频扩散中的上下文效应。arXiv 预印本 arXiv:2608.05237，2026年。 14[56]Shuai Yang、Wei Huang、Ruihang Chu、Yicheng Xiao、Yuyang Zhao、Xianbang Wang、Muyang Li、Enze Xie、Ying-cong Chen、Yao Lu 等。LongLive：实时交互式长视频生成。arXiv 预印本 arXiv:2509 22622，2025年 2，3，4，5，7，12，13，14。。[57] Yang Yang、Tianyi Zhang、Wei Huang、Jinwei Chen、Boxi Wu、Xiaofei He、Deng Cai、Bo Li 和 Peng-Tao Jiang。锚点强制：用于交互式流式视频扩散的锚点记忆和三域RoPE。载于欧洲计算机视觉会议，页码 399–416。Springer，2026年。 2，5[58] Zhuoyi Yang、Jiayan Teng、Wendi Zheng、Ming Ding、Shiyu Huang、Jiazheng Xu、Yuanming Yang、Wenyi Hong、Xi-aohan Zhang、Guanyu Feng 等。CogVideoX：采用专家Transformer的文本生成视频扩散模型。载于国际学习表征会议，页码 83048–83077，2025年。 2，3[59] Hidir Yesiltepe、Tuna Meral、Adil Kaan Akan、Kaan Ok-tay 和 Pinar Yanardag。Infinity-RoPE：从自回归自展开中涌现的动作可控无限视频生成。载于 IEEE/CVF计算机视觉与模式识别会议论文集，页码 40256–40265，2026年。 2，4，5，8[60] Tianwei Yin、Qiang Zhang、Richard Zhang、William T Free-man、Fredo Durand、Eli Shechtman 和 Xun Huang。从慢速双向视频扩散模型到快速自回归视频扩散模型。载于 2025 IEEE/CVF计算机视觉与模式识别会议（CVPR），页码 22963–22974。IEEE，2025年。 2，3[61] Zangwei Zheng、Xiangyu Peng、Tianji Yang、Chenhui Shen、Shenggui Li、Hongxin Liu、Yukun Zhou、Tianyi Li 和 Yang You。Open-Sora：让所有人都能进行高效视频制作。arXiv 预印本 arXiv:2412.20404，2024年。 2[62] Zangwei Zheng、Xiangyu Peng、Yuxuan Lou、Chenhui Shen、Tom Young、Xinying Guo、Binluo Wang、Hang Xu、Hongxin Liu、Mingyan Jiang 等。Open-Sora 2.0：训练一200 k 美元的商用级视频生成模型。 *arXiv* 预印本 *arXiv:2503.09642*，2025 年。 2[63] Hongzhou Zhu、Min Zhao、Guande He、Hang Su、Chongx- uan Li 和 Jun Zhu。Causal Forcing：Autoregressive diffu- sion distillation done right f or high-quality real-time inter- active video generation。*arXiv* 预印本 *arXiv:2602 02214*，2026 年2，5。。[64] Yixuan Zhu、Jiaqi Feng、Wenzhao Zheng、Yuan Gao、Xin Tao、Pengfei Wan、Jiwen Lu 和 Jie Zhou。Astra：General interactive world model with autoregressive denoising。在 *In-ternational Conference on Learning Representations*，第 79167–79184 页，2026 年。 14[65] Kai Zou、Dian Zheng、Hongbo Liu、Tiankai Hang、Bin Liu 和 Nenghai Yu。HiAR：Efficient autoregressive long video generation via hierarchical denoising。在 *European Confer- ence on Computer Vision*，第 225–242 页。Springer，2026 年。14

R残差引导的实时交互式视频生成

补充材料

A. 分析详情

A.1. 上下文惯性提示集与评估

提示词。 第 3 节所使用的100组切换对由GPT-5生成。5 [39] 这些切换对涵盖八类变化：人体运动、动物运动、物体运动、物理状态、颜色、场景、面部表情和光照。每组切换对由一个提示词组成， y_{old} 用于描述变化前的状态；另一个提示词 y_{new} 用于描述变化后的状态。两者保持主体和场景不变，因此只有发生变化的属性不同。每个提示词末尾都带有后缀“逼真视频、自然光照、固定摄像机。”，但表4中省略了该后缀。 4。请注意，该数据集与第 5 节所使用的T2V-CompBench提示词不重叠。

表4. **上下文惯性数据集的提示对。** 每个类别各展示一组示例。

类别	提示词对示例
人体运动	y_{old}：一名穿着灰色连帽衫的年轻男子正在城市公园里慢慢地行走。 y_{new}：一名穿着灰色连帽衫的年轻男子正在快速奔跑城市公园里。
动物运动	y_{old}：一匹棕色马正在开阔的草地上缓慢行走。 y_{new}：一匹棕色马正在一片开阔的草地。
物体运动	y_{old}：一个木制陀螺正直立在桌子上旋转。 y_{new}：一个木制陀螺侧躺着，静止地放在桌子上。
物理状态	y_{old}：花园里的一朵玫瑰还是紧闭的花蕾。 y_{new}：一朵盛开的玫瑰，花瓣舒展，在一个花园里。
颜色	y_{old}：一只白色泰迪熊坐在床上。 y_{new}：一只黑色泰迪熊坐在床上。
场景	y_{old}：一名男子在明媚的晴天站在一条城市街道上晴朗的一天。 y_{new}：一名男子在大雨中站在一条城市街道上。
表情	y_{old}：一个小男孩开心地微笑的特写，位于一教室。 y_{new}：一名哭泣的小男孩的特写，泪水挂在脸颊上，身处教室。
光照	y_{old}：一支白色蜡烛正燃烧着，火焰明亮地映照在一张深色桌子上。 y_{new}：一支白色蜡烛被吹灭，留下细细的一缕烟雾，飘散在深色桌子上。

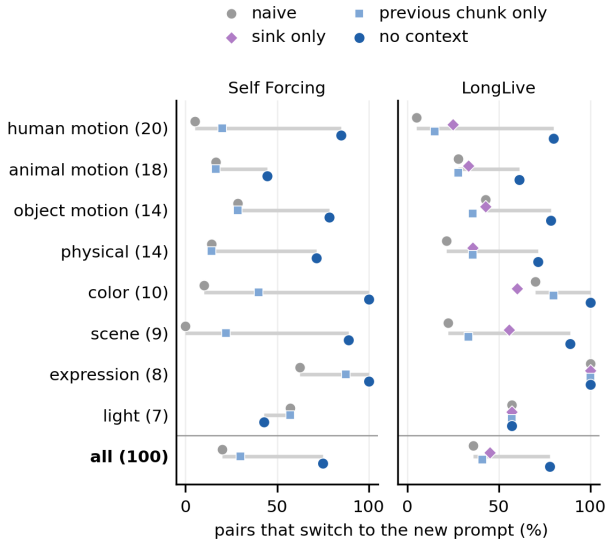


图 7. **按类别划分的上下文惯性。**在 Self Forcing（左）和 LongLive（右）中，当隐藏部分缓存并重新生成切换块时，切换到新提示词的配对占比。朴素方法表示基础模型。括号内为每个类别的配对数量。隐藏整个上下文可以恢复大多数切换，而只隐藏单个缓存块或汇点则只能恢复少量切换。

指标。我们使用视觉语言模型评判器来判断提示切换是否生效。切换后，我们以每次一个块的步长，在新片段上滑动一个 1.5 秒的窗口（两个块）。对于每个窗口，我们询问 Qwen3-VL-8B [1]“切换帧之后是否能看到以下内容： <新提示>? ”，并读取第一个答案 token 中“Yes”的概率。如果任意窗口的概率超过 0.5，则计为已触达。

对于上下文惯性数据集，我们使用 100 个人工标注视频对 Qwen3-VL- 8B 评审模型进行验证，标注过程对方法和评审模型的判定结果保持盲态。我们确认，评审模型对 86% 的视频给出了与人工标签一致的判定（Cohen’s κ [8] = 0.72）。

A.2. 按类别划分的结果

节。3.2 衡量上下文对第 1 步预测的影响程度。这里我们考察这种影响是否会反映在生成的视频中，以及是否取决于变化的类型。我们使用上下文惯性数据集中的 100 对样本，涵盖八类变化；在每个去噪步骤中隐藏部分缓存，并仅重新生成切换块：无上下文、仅前一块，或对 LongLive 而言仅汇点。其他一切保持不变，若切换后任意一个连续两块的窗口被判定为呈现新提示词，则该样本计为

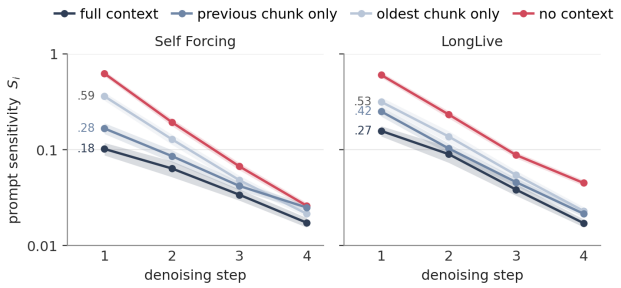


图 8. **可见上下文下逐步提示敏感度**。 在切换块的每个去噪步骤 i 中，固定噪声输入，并在每种可见上下文 $S_i(\mathcal{C})$ 下测量提示敏感度 $\mathcal{C}$，共 100 组配对（对数刻度，阴影区域表示 95% 自助法置信区间）。第 1 步数据点旁的数字表示 ρ_1，即图 2b中的数值。“仅保留最旧块”表示只保留仍在缓存中的最旧块：对于 Self Forcing，这是已被滚动窗口移除块 0 后的块 1；对于 LongLive，则是汇点（块 0）。

如果切换后任意两个块组成的窗口被判定为展示了新的提示词，则视为已切换。

图 7 显示，在不进行干预的情况下，Self Forcing 中只有 20% 的配对会发生切换，LongLive 中则为 36%。隐藏整个上下文后，这一比例升至 75% 和 78%；而仅保留前一块（30% 和 41%）或仅保留汇点（LongLive 中为 45%）则收效甚微。人体运动、场景和颜色变化的提升最大，因为新提示会改变其他方面完全相同的场景中的某一属性。相比之下，即使保留完整上下文，表情变化通常也能实现；而移除上下文并未改善光照变化，这表明问题可能在于基础模型本身的能力有限，而非上下文惯性。

A.3. 上下文惯性的逐步骤分析

在第 3.2 节中，我们测量了第一个去噪步骤的提示响应。本节在每个步骤重复测量，并表明第一步是提示词影响最大、上下文对其抑制也最强的步骤。所有结果均使用上下文惯性数据集中的100对样本，并遵循相同的实验协议。不同条件之间的差异通过对样本对进行配对自助法检验（5000次重采样），并计算95%的置信区间。

设置。 在切换块的每个步骤 i 中，模型在该步骤之前均保持不变运行，并且噪声输入 $x_k^{t_i}$ 保持固定。我们报告公式（1）的分子，即提示切换在干净预测中造成的位移，相对于块范数的比率，

$$S_i(\mathcal{C}) = \frac{\|\hat{x}^{(i)}(\mathcal{C}, y_{\text{new}}) - \hat{x}^{(i)}(\mathcal{C}, y_{\text{old}})\|}{\|s_{k-1}\|}, \qquad (5)$$

因此 $\rho_i(\mathcal{C}) = S_i(\mathcal{C})/S_i(\varnothing)$ 对于每一对，图2b展示了 ρ_1 按不同的配对取平均后的结果。我们绘制 S_i 而不是 ρ_i，因为无上下文敏感性本身在步骤1之后就会崩溃，所以尽管提示词在有无上下文的情况下对预测的影响都很小，比率在后续步骤中仍会趋近于1。图8 报告了 S_i 在四种可见上下文下的结果。

提示词在第一步发挥作用，而上下文也在这一步对其进行抑制。在没有上下文时，提示敏感性集中于步骤1： $S_1(\varnothing)$在 Self Forcing 上为0.62，在 LongLive 上为0.60，到步骤4时分别下降了 24× 和 13× 。加入完整上下文后，步骤1的敏感性降至0.10和0.16，逐对比率的平均值为 $\rho_1 = 0.18$ 和0.27。因此，在四个步骤中，步骤1单独就占了上下文导致的总敏感性损失的约75%和67%。尽管后续步骤中的相对抑制有所减弱，但绝对提示敏感性到步骤2时已经大幅下降（0.19和0.23），几乎没有机会恢复新提示响应。

单个缓存块就足够了。在步骤1中，随着上下文缩减，敏感性单调恢复。仅能看到前一块时， ρ_1 在 Self Forcing 上为 0.28，在 LongLive 上为 0.42；而仅能看到第一块（距离切换最远）时，这两个数值分别为 0.59 和 0.53。每个相邻差异都具有显著性（完整上下文与仅前一块之间为 $+0.11$ 和 $+0.16$，仅前一块与仅第一块之间为 $+0.31$ 和 $+0.10$，仅第一块与无上下文之间为 $+0.41$和 $+0.47$，全部为 $p < 0.001$）。紧邻的前一块是抑制作用最强的单一来源，然而即使只有一个缓存块，也会使响应减半。这就是我们选择在没有任何缓存上下文的情况下恢复残差，而不是仅在没有最近一块的情况下恢复残差的原因。

B. 方法细节

B.1. 残差引导的原理

我们根据提示条件化的分数差异，对公式（3）中的残差进行了近似解释。这一解释说明了残差如何捕捉新提示词所诱导的变化，同时不以缓存历史为条件控制，并阐明了其中涉及的近似。

后验分解。公式（2）由贝叶斯法则应用于 $p(s_k, y_{\text{new}} \mid x_k^t, \mathcal{C}_{<k})$ 得出，并结合了如下条件独立性：给定干净块和缓存后，含噪状态与提示词相互独立：

$$p(y_{\text{new}} \mid s_k, x_k^t, \mathcal{C}_{<k}) = p(y_{\text{new}} \mid s_k, \mathcal{C}_{<k}), \qquad (6)$$

之所以成立，是因为 x_k^t 是通过对 s_k独立加噪得到的。取对数后，后验可分解为

一个上下文项，对数 $p(s_k \mid x_k^t,\ \mathcal{C}_{<k})$，以及一个提示词项，对数 $p(y_{\text{new}} \mid s_k,\ \mathcal{C}_{<k})$。后者仍以缓存为条件。我们的残差引导并不直接估计这一缓存条件项，而是构造了一个不依赖缓存的提示词引起的变化近似，这一做法的动机源于第3节中观察到的抑制现象。

残差作为分数差。 令 $z = \tilde{x}_{k-1}^{\tau}$，并令 $p_\tau(z \mid y)$ 表示噪声水平为 τ 时噪声块的密度，该密度通过对在提示词 y 下生成且不带缓存上下文的干净块进行加噪获得。对于预测后验均值的理想去噪器，Tweedie公式 [11] 给出

$$\hat{x}_\theta(z \mid \varnothing, y) \approx \frac{z + \sigma_\tau^2 \nabla_z \log p_\tau(z \mid y)}{\alpha_\tau}. \tag{7}$$

该近似考虑了学生模型并非精确的后验均值估计器这一事实。

在切换块 k 处，前一个块 s_{k-1} 是在旧提示词 y_{old} 下生成的。我们将其在 y_{old} 下的无缓存重建近似为

$$\hat{x}_\theta(z \mid \varnothing, y_{\text{old}}) \approx s_{k-1}. \tag{8}$$

将该近似代入式（3），并在两个提示词下应用式（7）可得

$$\begin{aligned} r_k &\approx \hat{x}_\theta(z \mid \varnothing, y_{\text{new}}) - \hat{x}_\theta(z \mid \varnothing, y_{\text{old}}) \\ &= \frac{\sigma_\tau^2}{\alpha_\tau}\left(\nabla_z \log p_\tau(z \mid y_{\text{new}}) - \nabla_z \log p_\tau(z \mid y_{\text{old}})\right) \\ &= \frac{\sigma_\tau^2}{\alpha_\tau} \nabla_z \log \frac{p_\tau(z \mid y_{\text{new}})}{p_\tau(z \mid y_{\text{old}})}. \end{aligned} \tag{9}$$

在提示词服从共同先验的情况下，最后一个表达式等价于 y_{new} 相对于 y_{old} 的对数后验优势比的梯度，因为共享边际 $p_\tau(z)$ 根据贝叶斯法则相互抵消。因此，残差近似于相对于旧提示词的无缓存、提示词引导的位移，而不是对空提示词的响应。它为式（2）中的提示词项提供了一个局部替代量，但并不是缓存条件似然的精确梯度。

近似与适用范围。 上述解释涉及三种近似。首先，式（7）假设学生模型的预测近似于无缓存噪声边缘分布下的后验均值。这是对学得的学生去噪器的一种理想化解释，而非蒸馏模型的精确恒等式。其次，式（8）假设无缓存重建在 y_{old} 下恢复 s_{k-1}。然而，s_{k-1} 是在前一个缓存下生成的，而 $p_\tau(\cdot \mid y_{\text{old}})$ 描述的是无缓存边缘分布。因此，s_{k-1} 未必是该边缘分布中的典型样本，旧提示词重建也未必能准确恢复它。此外，单次重建无法从

算法1 提示切换后在块 k 处进行残差引导

要求： 学生模型 $\hat{x}_\theta$，步数 $t_1 > \cdots > t_Q$，调度 λ，噪声水平 τ，缓存 $\mathcal{C}_{<k}$，最后一个块 s_{k-1}，新提示 y_{new}

$\epsilon \sim \mathcal{N}(0, I)$ (在此提示词区间内固定)
$j \leftarrow k$
当当前提示仍为 y_{new} **时**
  $\tilde{x}_{j-1}^{\tau} \leftarrow \alpha_\tau\, s_{j-1} + \sigma_\tau\, \epsilon$ （重新加噪前一块）
  $r_j \leftarrow \hat{x}_\theta(\tilde{x}_{j-1}^{\tau} \mid \varnothing,\ y_{\text{new}}) - s_{j-1}$ （无缓存重建构）
  $x_j^{t_1} \sim \mathcal{N}(0,\ I)$
  对于 $i = 1$、.....、Q **执行**
    $\hat{x}_j \leftarrow \hat{x}_\theta(x_j^{t_i} \mid \mathcal{C}_{<j},\ y_{\text{new}}) + \lambda_i\, r_j$ (引导预测-值)
    如果 $i < Q$**那么**
      $\epsilon' \sim \mathcal{N}(0,\ I)$
      $x_j^{t_{i+1}} \leftarrow \alpha_{t_{i+1}} \hat{x}_j + \sigma_{t_{i+1}} \epsilon'$
    结束如果
  结束 for
  $s_j \leftarrow \hat{x}_j$ 并将其键和值追加到 $\mathcal{C}$
  $j \leftarrow j + 1$
结束 while

较高噪声水平下的加噪无法逆转，因此即使对于典型样本，也会偏离 s_{k-1}。其余部分 $\hat{x}_\theta(z \mid \varnothing, y_{\text{old}}) - s_{k-1}$ 是残差引导与补充材料 F.3 中提示差分残差之间的差异；该比较表明，这一差异不可忽略。因此，式（9）仅描述残差中依赖提示的部分。噪声水平 τ 在保留前一个块作为锚点与允许新提示词修改其内容之间起到平衡作用：噪声过低会限制可能的位移，而噪声过高则可能丢弃维持正在进行的场景所需的细节。我们根据第 $\tau = 850$ 节中的消融实验选择该值。5.3。第三，将残差从前一个块迁移到下一个块，依赖于两者之间的局部视觉对应关系，其动机是上下文惯性，而非两者预测之间的精确等价。式（9）中的分数差解释直接适用于切换块 k，其前一个块是在 y_{old} 下生成的。对于同一提示词区间内后续的块，s_{j-1} 已经是在 y_{new} 下生成的。因此，它们的残差不再代表新旧提示词之间的对比。相反，这些残差提供了以最新生成内容为锚点、由当前提示词引导的局部位移；随着缓存纳入在新提示词下生成的块，这些位移会持续提供引导。

B.2. 算法

算法 1 给出了在块 k处发生提示切换后所采用的流程，该流程持续到下一次提示切换。一个

噪声样本 ϵ 在切换时抽取，并在整个提示词区间内的所有残差之间共享（第 1 行）。对于每个块 $j \geq k$，前一个块 s_{j-1} 被重新加噪至等级 τ，并在当前提示下通过一次无缓存前向传播完成重建。残差 r_j 表示该重建相对于 s_{j-1} 的位移（第 4–5 行）。随后使用原始采样器对该块进行去噪，在每个步骤中将 $\lambda_i r_j$加到干净预测上（第 8 行）。当λ = （1，0。5， 0， 0）时，残差仅注入四个去噪步骤中的前两个步骤。生成的干净块s_j 随后照常追加到缓存中（第 14 行）。因此，残差本身从未写入缓存；只有最终生成的块会对后续上下文产生作用。当新提示到来时，该流程会使用新的噪声样本和更新后的提示重新开始。在提示保持激活状态期间，该流程对每个生成块额外需要进行一次无缓存前向传播，而在其他去噪步骤中无需额外的模型评估。

C. 评估协议

C.1. 基准测试与提示构造

提示词详情。 T2V-CompBench 动态属性 [46] 的提示词在单个句子中描述属性变化，例如“随着秋季推移，一片叶子从绿色变为鲜红色的延时摄影”。因 y_{old} 此，我们将每个提示词拆分为一个用于初始状态的提示词和一个用于最终状态的提示词，同时固定主体和场景，例如y_{new} “树枝上的一片绿色叶子”和 y_{old} “树枝上的一片鲜红色叶子”。随后，将对运动、光照和摄像机的共享描述附加到两个提示词中（例如：“……，在微风中轻轻摇曳，柔和的自然光，摄像机缓慢推进”）。该描述有两个作用。首先，对某一状态的简短描述往往会生成几乎静止的视频，而附加的描述则赋予场景持续的运动，这种运动必须在切换过程中继续存在，正如交互式使用中的情况。其次，流式模型是在较长且详细的提示词上训练的，附加描述使输入更接近这一分布。由于该描述在 y_{new}和 [53] 中完全相同，两个提示词的差异仅在于状态发生了变化，因此切换后的任何差异都可以归因于该变化。拆分由大语言模型完成，并由作者手动检查。StoryEval [56]包含涉及两个到四个事件的提示词。由于包含两个事件的提示词只有一次切换，而动态属性设置已经涵盖了这种情况，因此我们随机抽取了100个包含三个事件的提示词。InteractiveBench [22], 采用 MemFlow 发布的版本，每个提示词激活10秒。

C.2. 指标详情

这三种场景使用相同的指标。对于每次切换，我们将新提示词下生成的块称为切换后的片段；衡量帧间一致性的质量指标会排除切换块，因此不会将提示词要求的变化视为不一致并予以惩罚。

达成度。 视觉语言模型评判器（Qwen3-VL-8B [1]）会将切换前的最后一帧与切换后片段中一个包含两个块（8 帧， 832×480）的滑动窗口进行比较，并判断是否能看到请求的变化（例如“绿色叶子” →“红色叶子”）。如果响应为 P(是) > 0.5，则视为达成了一次切换。由于模型可能完全无法呈现新状态，我们使用相同的随机种子和仅包含新提示词的提示词生成一个上界视频；达成度即为在上界视频实现了变化的切换中，实际达成的切换所占的比例。如果视频中包含多次切换，则对这些切换的达成度取平均值。我们会在补充材料 C.3 中进一步讨论这一指标的可靠性。C.3。

T2V-CompBench 动态属性。 对于动态属性场景，我们还会运行 T2V-CompBench [46], 的官方评估器，该评估器使用 LLaVA-1.6-34B [32] 逐帧评估属性是否从初始状态变为最终状态，并对整个视频和全部 200 对样本进行评分。对于始终没有变化的视频，该分数接近 0；根据其构造方式，分数的绝对值也较小。

CLIP。 CLIP 相似度 [42]通过比较切换后片段的帧与新提示词来衡量，对各块的结果取平均后进行报告 $\times 100$。在 InteractiveBench 上，按照以往工作的报告形式，按每个 10 秒片段分别与该片段的提示词进行计算。

质量。 我们严格沿用 VBench [20] 质量分数。七个维度均使用官方 VBench 常数进行最小-最大归一化，质量分数则为它们的加权平均值：动态程度的权重为 0.5，其余六项的权重均为 1，再除以总权重 6.5。主体一致性（跨帧 DINO ViT-B/16 [5] 特征相似度）、背景一致性（跨帧 CLIP 特征相似度）、时间闪烁（帧间平均绝对像素差异）、运动平滑度（AMT [30] 帧插值与生成帧之间的差异）、美学质量（CLIP ViT-L/14 上的 LAION [44] 美学预测器）和动态程度（RAFT [48]光流）均在切换后片段上计算，但不包括切换块；成像质量（MUSIQ [24]）则在整个切换后片段上计算。背景一致性仅针对提示词未改变背景的帧对计算（动态属性中的 200 对中有 150 对）。在 InteractiveBench 上，质量分数

还会使用 VBench-Long 在整个视频上计算（补充材料 D.4）。

总体。 $(4 \times$质量 $+$ 达成度$)/5$，即以达成度替代语义分数后的 VBench 总分。

C.3. 视觉语言模型评判器的可靠性

所有表格中的达成度均由 Qwen3-VL-8B [1] 评定。为评估结果是否会受到评判器选择的影响，我们使用同一模型的较大版本 Qwen3-VL-32B 和 InternVL3.5-38B [52] 重复评估。我们从动态属性设置中随机抽取 100 组提示词对，分布于七种基础模型，并对每组中的基线视频和残差引导视频进行评估，共计 200 个视频。三个评判器的两两一致率为 91% 至 93%，不同评判器对之间的 Cohen's κ [8] 值介于 0.81 至 0.86。

我们还将这三个评判器对同一批 200 个视频的评估结果与人工标注进行比较。为避免偏差，视频以随机顺序呈现，不透露基础模型，也不说明是否使用了残差引导。标注者观看每个视频的完整内容，而视觉语言模型评判器则使用由两个块组成的滑动窗口评估切换后片段。8B、32B 和 InternVL 评判器与人工标签的一致率分别为 84%、85% 和 85%，Cohen's κ 值分别为 0.67、0.68 和 0.70。这些结果表明，三个评判器与人工判断的一致程度相近，证实主表中使用的 8B 评判器并非异常值。

还需要注意的是，提示词保真度的提升得到了不依赖同一视觉语言模型评判器的指标支持。CLIP 不涉及视觉语言模型，而 T2V- CompBench 的官方动态属性得分使用了不同的模型（LLaVA-1.6- 34B [32]）和评估协议；这两项指标在表 1 中的每个基础模型上均有所提升。1，与达成度一致。

C.4. 用户研究详情

配对。 该研究采用表 1中的动态属性设置及其七个基础模型。每个项目都比较一个基础模型与同一模型在相同提示词对和随机种子下使用残差引导的结果，因此两个视频在切换之前完全相同。我们为每个基础模型随机选取 15 对（共 105 对）。

展示。 每个项目都是一张包含两行的图像，每行对应一个视频，分别标记为顶部和底部。每行在 832×480 处显示五帧：切换前最后一个块的中间帧（第 26 帧）、标记切换位置的红色条带（第 33 帧，即 16 帧率下的 2.06 秒），以及切换后的四个块的中间帧（第 38、50、62 和 74 帧）。我们仅在帧的上方打印提示词中发生变化的部分，前后对比。每行所采用的方法均被隐藏。在每种表单的半数项目中，残差引导会随机出现在顶部，且每种表单中的项目顺序都会被打乱。

问题。 对于每个项目，评审者回答三个问题。Q1 要求评审者分别针对每个视频判断：在标记点之后，除了提示词所要求的变化之外，他们是否注意到任何非预期变化，例如背景或其他物体的变化（否或是）。Q2 询问哪个视频在标记点之后更好地呈现了新提示词（顶部、底部或无差异）。Q3 询问如果他们在视频生成过程中实时切换提示词，他们更希望得到哪个视频（顶部、底部或无差异）。说明中指出，提示词所要求的变化不算作非预期变化。

表格与参与者。 用户研究不涉及任何敏感个人信息，也不会给参与者带来可预见的风险。所有参与者均在参与前提供了知情同意。105 对样本被分为 5 份表格，每份包含 21 对，每个基础模型对应 3 对，每位参与者完成一份表格。我们招募了 20 位未参与该项目的参与者，每份表格由 3 至 5 人完成。每份表格耗时 10 至 15 分钟。

分析。 对于 Q2 和 Q3，选择残差引导时计 1 分，选择基础模型时计 0 分，认为没有差异时计 0.5 分。对于 Q1，仅标记基础模型的视频时计 1 分，仅标记我们的视频时计 0 分，其他情况计 0.5 分。我们报告 420 个判断的均值，以及基于评审者和样本对计算的 95% 自助法置信区间（重采样 2000 次）：Q2 为 73.1% [66.8、79.2]，Q3 为 70.7% [64.7、76.9]，Q1 为 44.3% [38.9、49.6]。

D. 附加结果

D.1. 请求的变化对质量的影响

RG 对 VBench质量 的影响最多为 0.008，在七个模型中的六个模型上略有下降（表 1）。表 5 将该分数分解为七个维度。RG 使每个模型的三个一致性维度（主体一致性、背景一致性和时间闪烁）最多降低 0.03，同时提高动态程度；而美学质量、成像质量和运动平滑度与基础模型的差异均保持在约 0.01 以内。这两种变化源于同一个原因。切换后，基础模型保持初始状态，因此几乎是静止的，从而无需额外代价即可获得较高的一致性 [20], 而 RG 则实现了变化，变化前后的帧存在差异。在变更对象之外的场景，相比无上下文设置，与切换前帧的接近程度显著更高（表 3 中的非预期变化），但仍存在一些场景漂移（补充材料 D.2）。因此，下降幅度为

表5. **T2V-CompBench 动态属性上的 VBench 维度。** 每个单元格给出基础模型 → 以及采用残差引导的同一模型在切换后片段上的计算结果，如表 1所示。

方法	主体	背景	闪烁	平滑度	美学	成像	动态	质量
Self Forcing	0.987 → 0.960	0.985 → 0.967	0.986 → 0.977	0.992 → 0.988	0.578 → 0.566	67.09 → 66.80	0.250 → 0.380	0.813 → 0.806
Causal Forcing	0.981 → 0.956	0.982 → 0.965	0.977 → 0.962	0.988 → 0.978	0.584 → 0.572	68.22 → 67.92	0.425 → 0.600	0.821 → 0.813
MemFlow	0.987 → 0.976	0.984 → 0.976	0.985 → 0.980	0.992 → 0.989	0.571 → 0.565	67.51 → 67.63	0.175 → 0.235	0.805 → 0.802
LongLive	0.987 → 0.978	0.986 → 0.978	0.988 → 0.982	0.993 → 0.989	0.580 → 0.571	66.37 → 66.57	0.145 → 0.215	0.805 → 0.802
Infinity-RoPE	0.990 → 0.968	0.985 → 0.971	0.991 → 0.983	0.993 → 0.990	0.579 → 0.571	65.14 → 64.06	0.155 → 0.310	0.806 → 0.803
Anchor Forcing	0.978 → 0.965	0.979 → 0.969	0.975 → 0.965	0.987 → 0.980	0.578 → 0.578	67.12 → 67.06	0.365 → 0.540	0.811 → 0.813
SWIFT	0.990 → 0.978	0.988 → 0.979	0.990 → 0.984	0.994 → 0.990	0.580 → 0.573	66.47 → 66.71	0.120 → 0.150	0.806 → 0.799



图9。**VBench质量会对请求的变化给予惩罚**。每个示例中，顶行为基础模型，底行为残差引导，虚线标示提示切换。这些示例分别来自 LongLive（上）、Causal Forcing（中）和 Infinity-RoPE（下）。灰色数字表示基础模型和残差引导的 VBench维度分数。残差引导遵循新提示词，而基础模型保留旧状态，但残差引导在美学质量、主体一致性和成像质量上的得分分别更低。

一致性在一定程度上与请求的变化有关，但这并不排除周围场景发生意外变化。

为考察具体案例，我们检查了质量下降幅度最大的提示词对。图9 展示了其中三个案例，以及变化幅度最大的 VBench 维度（无论变化方向如何）。冰块案例（中间）体现了平均趋势。冰块融化成一洼水后，主体一致性从 0.999 降至 0.901，时间闪烁从 0.999 降至 0.960。另两个案例涉及美学和成像得分，这两项得分的平均值没有变化，但取决于所展示的内容。按照请求，三维像素艺术花朵变成一个纯白方块时，美学质量从 0.655 降至 0.293，成像质量从 74.1 降至 44.8。裸露的土壤变成繁花盛开的花园时，成像质量从 66.3 降至 56.0，而美学质量从 0.595 升至 0.711。因此，这些得分会随请求的内容朝任一方向变化，这与其平均值几乎没有变化的情况一致。在这三个案例中，基础模型之所以仍保持高分，只是因为视频没有发生变化。因此，质量下降在很大程度上是请求的变化所致；质量应与达成度一并解读，这也是总体得分将两者结合的原因。

D.2. 切换时的场景保持

更强烈的切换或许只需重新绘制场景即可实现。我们先在切换块中考察这一点，然后考察切换后片段。

图 10 展示了切换块中的帧。如第 3 节的无上下文条件所示，在切换时清空缓存，会遵循新提示词，但生成的块与前序帧脱节，气球和叶子也在不同的场景中被重新绘制。残差引导（RG）会在同一块内改变状态，同时比清除缓存保留更多前序主体和背景，使切换更加即时，又不至于完全破坏视觉连续性。

为量化这一点，我们使用结构距离来衡量切换后帧相对于最后一个切换前帧的偏移程度 [50]（DINO自相似度）和感知距离（DreamSim [12]），计算时将变更对象遮罩掉（表 6）。遮罩通过使用提示词的两种属性状态作为文本查询，利用 OWLv 检测发生变化的物体，再用 SAM 对检测到的框进行分割来获得2 [37], 。 [25]在每个模型上，残差引导使场景的变化幅度都大于基础模型。这是预期结果，因为基础模型通常根本不会发生切换，而且结构距离还包含了请求的变化。不过，这种变化仍然受到限制。使用残差引导时，平均结构距离介于 0.033 和 0.059 之间；在七个模型中的六个上，该数值不超过同一帧进行 1.2× 缩放时的距离（0.047）。

表 6。**T2V-CompBench 动态属性上的场景保持**。在全部 200 个提示对中，最后一个切换前帧与切换后帧之间的距离：分别统计切换块（切换）上的距离，以及四个切换后块的平均距离（平均值）。数值越低，表示越接近切换前场景。结构距离是整帧上的 DINO自相似度距离 [50] 。DreamSim [12]的计算将变更对象遮罩掉（OWLv2 [37] 和 SAM [25]）。参考行报告的是切换前帧与其变换副本之间的相同距离，计算范围为整帧。参考值不取决于块，并覆盖两列。

	结构 ↓		DreamSim（背景）↓	
方法	切换	Mean	切换	Mean
Self Forcing	0.010	0.019	0.015	0.028
+ RG	0.030	0.045	0.060	0.091
Causal Forcing	0.023	0.037	0.028	0.045
+ RG	0.043	0.059	0.081	0.117
MemFlow	0.009	0.020	0.011	0.024
+ RG	0.017	0.033	0.021	0.041
LongLive	0.007	0.016	0.009	0.023
+ RG	0.022	0.035	0.030	0.047
Infinity-RoPE	0.008	0.014	0.012	0.024
+ RG	0.027	0.040	0.047	0.072
Anchor Forcing	0.013	0.030	0.016	0.039
+ RG	0.026	0.046	0.032	0.060
SWIFT	0.007	0.013	0.009	0.018
+ RG	0.023	0.033	0.031	0.046
参考				
缩放 1.2×	[0.047]		[0.037]	
偏移 20%	[0.110]		[0.058]	
另一个视频	[0.133]		[0.782]	

以及远低于另一个视频（0.133）。背景距离介于 1.2× LongLive、MemFlow 和 SWIFT 上的缩放与 20% 偏移（0.037 和 0.058）之间，而在其他模型上最高约为偏移的两倍，比另一个视频（0.782）低一个数量级。切换本身不会引入突兀的断裂，因为在每个模型上，切换块处的结构距离都低于缩放参考值。相比之下，清空缓存对背景的改变明显大于残差引导（RG）：在 Self Forcing 和 LongLive 上，物体掩码 DreamSim 距离分别为 0.191 和 0.163，而残差引导（RG）对应为 0.089 和 0.046（表3）。

D.3. 更多定性结果

图11、12 和13 给出了这三种设置下的更多示例，每个示例使用一个基础模型，其下方为使用相同模型和残差引导（RG）的结果。

在 T2V-CompBench 动态属性（图 11）上，基础模型在切换后保持初始状态，而残差引导则在最初几个块中实现了形状变化（长方体展开成一张图形，冰块融化成一滩水）、状态变化（冰淇淋滴落，面包长出霉菌）和颜色变化（墨水变成绿色），同时保留物体、其位置以及

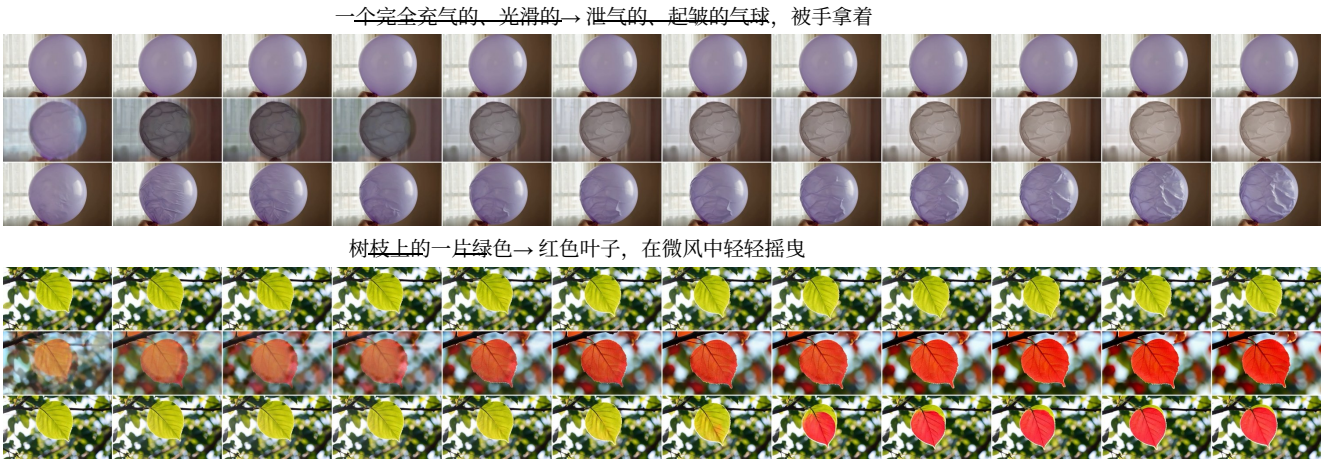


图10. **切换块的帧**。 LongLive（上）和 Anchor Forcing（下）。每个示例中，第一行是基础模型，它延续先前的状态。第二行在切换时清空缓存，如第3节的无上下文条件一样，并遵循新提示词，但在切换块中丢失场景。第三行是残差引导（RG），它在切换块中呈现新状态，同时保留主体和背景。

背景。在 Causal Forcing 上，细细的月球逐渐发生变化，变化过程历经数个块，而非一蹴而就。

在 StoryEval（图12）中，基础模型往往会稍作变化地延续第一个事件，例如熊始终待在篮子里，或药剂师一直在清点药片。残差引导（RG）会按照提示词逐一跟进事件，因此小袋鼠会跳出来奔跑，药剂师会装瓶并贴上标签，企鹅则会滑行并潜入水中。对于 SWIFT，残差引导（RG）也能让玻璃杯洒出的水汇成河流，而基础模型中的玻璃杯仍留在原处。

在 InteractiveBench（图13）上，基础模型只能响应六条提示词中的一部分，遗漏了其他提示词。例如，SWIFT 从未呈现歌手与吉他手同台的画面，而且人群直到最后才出现。使用残差引导（RG）时，每个新主体都会在收到提示词后登场，场景也随脚本逐步展开；新主体还会与已有主体互动（面包师递上羊角面包、母亲拥抱男孩），同时场景在全部五次切换中始终保持连贯。

D.4. InteractiveBench 上的 VBench-Long

表 7 报告了 Interac- tiveBench 上的 VBench-Long维度，这些数据基于完整的 60秒视频计算。因此，其数值与表2中的质量不同，后者基于切换后片段计算（补充材料 C.2）。这一模式在全部五个基础模型中都相同，并且与 5 秒结果一致。使用残差引导后，主体和背景一致性下降了 0.005 至 0.009，这是预期现象，因为五次提示切换现在都得到了实现，而不是被忽略；内容发生变化的视频，其帧级一致性不可能像

表7. **InteractiveBench 上的 VBench-Long维度**。在完整的 60 秒视频上计算（100 个视频）。质量是涵盖全部七个维度的 VBench 得分。各模型的时间闪烁、运动平滑度和美学质量变化均小于 0.005，因此省略。

方法	主体 ↑	背景 ↑	成像 ↑	动态 ↑	质量 ↑
MemFlow	0.976	0.960	0.718	0.269	0.815
+ RG	0.971	0.954	0.724	0.340	0.817
LongLive	0.975	0.960	0.719	0.280	0.818
+ RG	0.968	0.953	0.722	0.354	0.819
Infinity-RoPE [59]	0.981	0.965	0.713	0.370	0.825
+ RG	0.972	0.959	0.712	0.505	0.829
Anchor Forcing	0.959	0.954	0.724	0.729	0.835
+ RG	0.954	0.949	0.722	0.789	0.834
SWIFT	0.974	0.960	0.727	0.299	0.820
+ RG	0.966	0.954	0.725	0.378	0.820

内容不变的视频那样获得高分。成像质量在任一方向上的变化均不超过 0.006；在每个模型上，时间闪烁、运动平滑度和美学质量的变化都小于 0.005，因此该引导不会带来可见的质量下降。例如，在 MemFlow 上，动态程度从 0.27 大幅升至 0.34；在 Infinity-RoPE 上则从 0.37 升至 0.51，这反映出运动会跟随不断变化的提示词，而不是逐渐停留在静态场景中。按照官方加权，除 Anchor Forcing 外，在所有基础模型上，使用残差引导后的聚合质量分数均保持不变或略有提高；在 Anchor Forcing 上，该分数则降低了 0.001。在两种评测协议下，质量与基础模型的差异都不超过 0.006。

D.5. 实时实用性

表 8 表明，残差引导（RG）只需付出很小的代价，就能让切换更快生效。RG 会为每个块增加一次前向传播，因此，对于包含

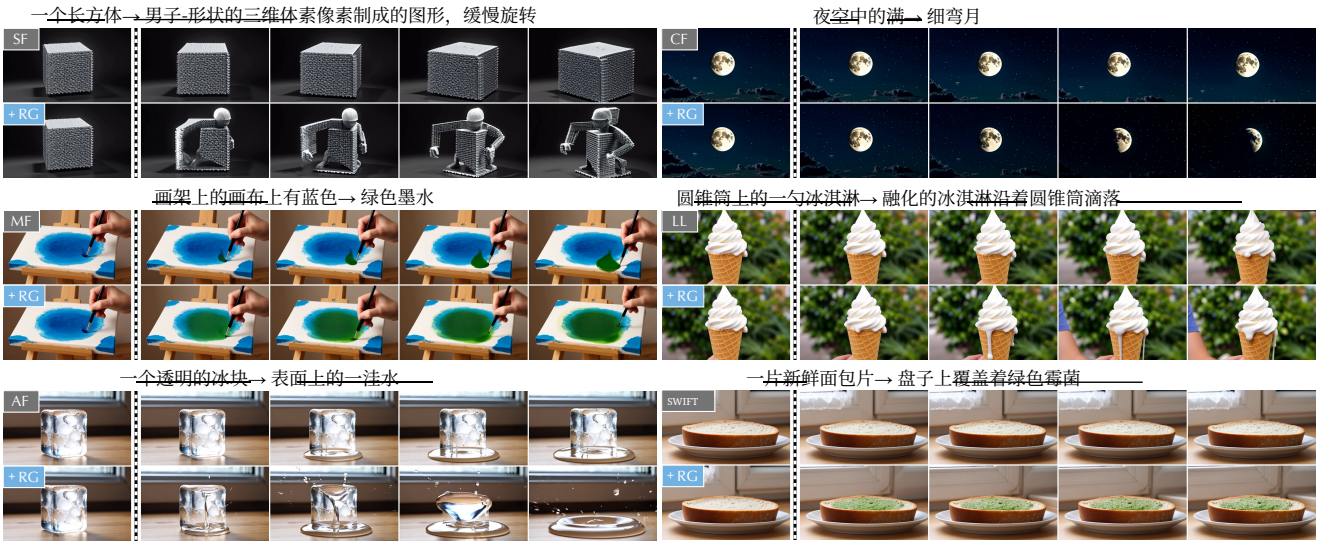


图11。 **定性结果** **在 T2V-CompBench 动态属性上的表现**。 每个示例展示了基础模型（上）以及同一模型 l
搭配残差引导（下）的结果。白到黑的色条标示提示切换。



图12。 **定性结果** **在 StoryEval 上**。每个示例展示了一个基础模型（上）以及加入残差引导后的同一模型（下），跨越三 e
个事件。白到黑的条带 标记了两次提示切换。

在 RTX A6000 和 H100 上各进行一次切换的 5 秒视频，端到端每秒帧数会降低约 6%。表 8 中的帧率是针对整个 5 秒视频测得的，其中包括 VAE解码，以及在未启用残差引导（RG）的情况下生成的三个切换前块。在残差引导（RG）对每个块都处于启用状态的长时段会话中，开销受每个块的额外开销限制。

仅测量切换后的块时，使用残差引导生成的块在 H100 上运行 Self Forcing 和 LongLive 的帧率分别为 21.7 和 18.2 FPS，比基础模型低 13% 和 16%，但仍高于 16 FPS 的播放帧率，且峰值内存不变。我们通过响应速度——

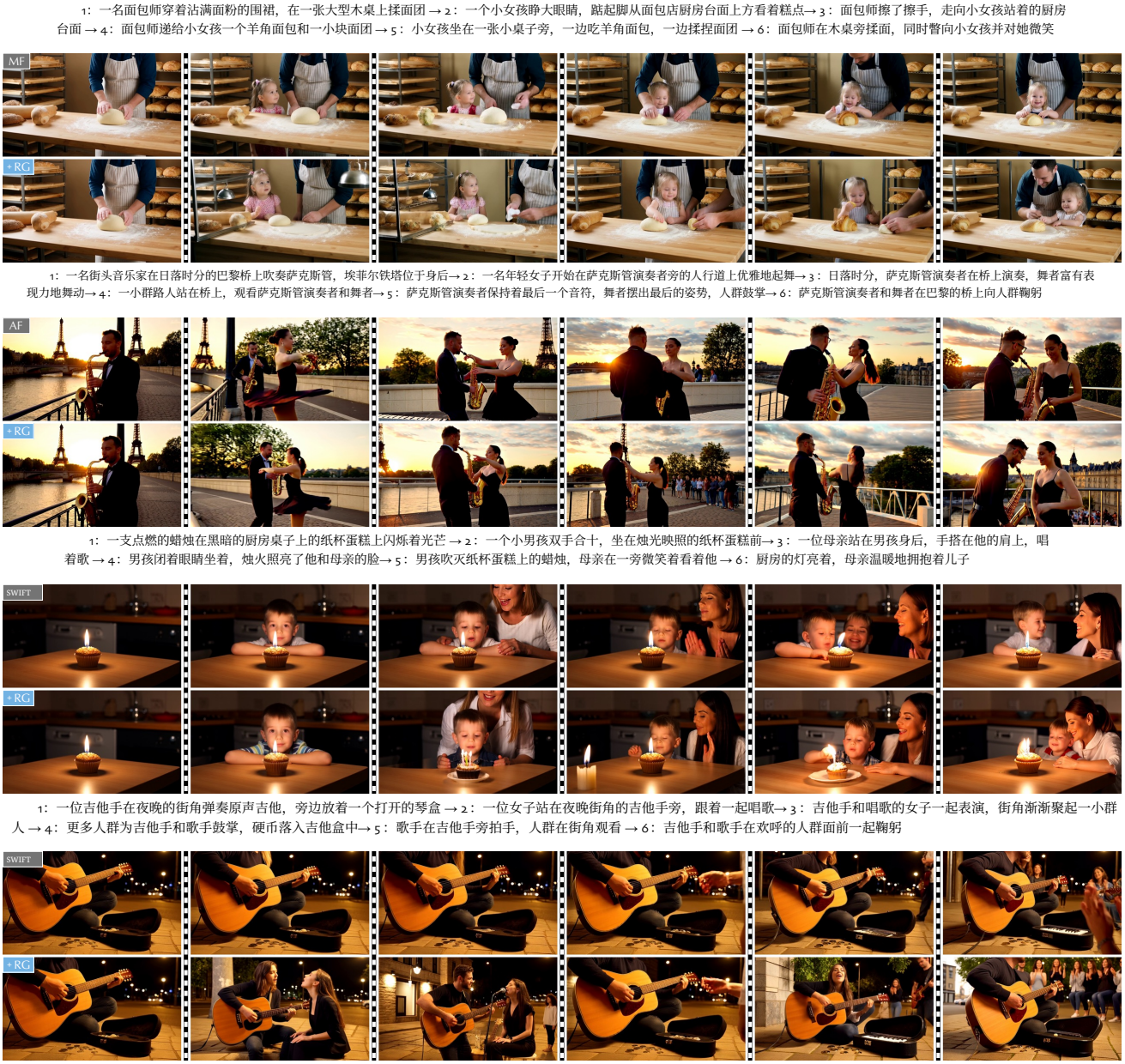


图13. **InteractiveBench 上的定性结果**。每个示例展示了基础模型（上）和应用残差引导后的同一模型（下）在 60 秒内响应六个提示词的结果，每个提示词对应一帧。由白到黑的色条标记了五次提示切换。

表8. **吞吐量与响应性**。在批量大小为 1、视频时长为 5 秒且切换一次的条件下测量。残差引导会为每个块增加一次前向传播，使吞吐量降低约 6%。

	开销（帧率 ↑）		响应性		
方法	A6000	H100	切换 ≤2 块 ↑	起始时间（秒）↓	ΔCLIP ↑
Self Forcing	4.14	16.98	0.07	0.75	0.65
+ RG	3.91	15.98	**0.35**	**0.12**	**3.91**
LongLive	3.52	14.02	0.21	0.62	1.63
+ RG	3.31	13.11	**0.54**	**0.19**	**4.22**

来衡量响应速度：在两个块内实现的切换比例（已切换 ≤2 块）、切换后相对于旧提示词，新提示词带来的 CLIP 相似度提升（ΔCLIP），以及该提升达到最终值一半所需的时间（起始时间）。在两个模型上，残差引导都使已切换比例和 ΔCLIP 提升一倍以上，并将起始时间中位数缩短至 0.2 秒以下。没有残差引导时，即使是 LongLive，起始时间仍为 0.62 秒；其切换时的缓存适配也无法消除上下文惯性。

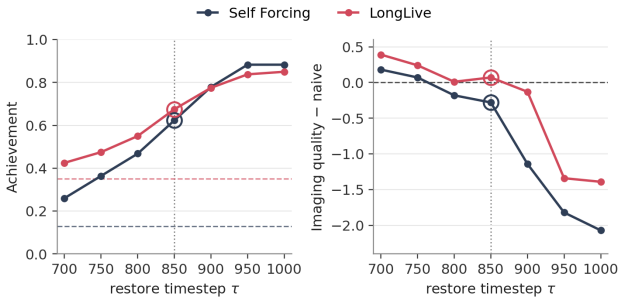


图 14。 **关于噪声水平的选择** τ。 在 100 对样本上，相对于基础模型的达成度（左）和成像质量（右）。虚线表示基础模型。达成度始终随 τ而上升，而成像质量在 $\tau = 850$ （圆圈所示）之前保持稳定，超过该值后则急剧下降；此时恢复块已无法保留主体（图 6）。

D.6. 消融实验结果

我们在 Self Forcing（SF）和 LongLive（LL）的 T2V-CompBench 动态属性集中的 100 对随机样本上评估所有消融实验。除非另有说明，我们使用$\tau = 850$， $\lambda =$ （1，0。5，0， 0 ），并使用共享噪声 ϵ，每次仅改变一个因素（表 9 ）。

噪声水平。 噪声水平 τ 控制恢复块相对于前一个块的偏离程度，从而决定新提示词覆盖上下文的强度。达成度随 τ单调上升：在 Self Forcing 上，从 τ =700 时的 0.26 上升到 $\tau = 1000$ 时的 0.88，而 LongLive 也呈现相同趋势（表 9a）。相比之下，成像质量在 $\tau = 850$之前保持稳定，但超过该值后下降超过 1 分（图 14）。图 6 展示了这一权衡：在 $\tau = 700$时，标志仍为黄色；在 $\tau = 850$时，其颜色发生变化，但形状得以保留；而在 $\tau = 1000$时，标志以不同形状重新生成，表明恢复块已无法保留主体。因此，我们使用 $\tau = 850$；在允许请求的变化的同时，主体能够在切换过程中保持不变。

重要的是，与其将 τ 视为需要针对每种配置进行调优、以使残差引导发挥作用的参数，不如将其看作一个由用户控制的旋钮。与无分类器引导中的引导尺度类似，它控制着对新提示词的响应性与当前场景保留之间的权衡。事实上，我们扫描范围内的每个取值都比基础模型提高了达成度。为简便起见并保持一致性，我们在所有模型和基准测试中使用相同的 $\tau = 850$ ，不针对每种配置进行调优。用户可以调整它，以偏向响应性或保留，而首选的平衡点可能因模型和提示词类型而异。

步数调度。 仅在第一步应用引导，可使达成度得到部分提升（0.47 和 0.63）。将全强度引导延长至第二步，可进一步提高响应性，但会略微牺牲主体一致性。在第三步之前持续应用引导，只能带来有限的额外收益，却会明显降低质量和主体一致性。我们的默认调度方案在第二步应用半强度引导，因而实现了更好的平衡：它强化了提示切换，同时将其余步骤留给生成内容进行细化。

噪声共享。 无缓存过程会用一个噪声样本重新添加噪声到前一个块 ϵ，因此重新绘制的块及其残差 r_k取决于 ϵ。如果每个块都抽取一个新的 ϵ，提示词区间中的每个块都会被拉向新提示词的不同实现，从而导致块与块之间出现主体漂移。因此，我们在切换时抽取一个 ϵ，并在提示词区间的每个块中重复使用。表9c 显示，共享噪声能提升主体一致性，在 Self Forcing 上尤为明显，并使两个模型的质量都略有提高。在 LongLive 上，达成度相近；在 Self Forcing 上则有所降低，我们认为这是因为逐块重新采样会更大幅度地改变主体。

E. 补充实验

E.1. T2V-CompBench 动作绑定

T2V-CompBench 的动作绑定提示描述了两个主体执行不同动作。我们将第一个主体用作 y_{old} ，并将第二个主体附加为 y_{new}，这样切换就会要求模型在场景中引入一个新主体，而不是改变现有主体的属性。这与动态属性切换的类型不同：必须完整保留先前的内容，并在其旁边添加一个新元素。我们在 MemFlow 和 LongLive 上进行评估，并报告该基准测试的官方动作绑定分数，以及表1中的指标。

表。 10 显示，残差引导将 MemFlow 上的达成度从 0.386 提升至 0.605，并将 LongLive 上的达成度从 0.450 提升至 0.667，官方动作绑定得分和 CLIP 也随之提升。质量与基础模型的差异保持在 0.003 以内，动态程度则有所增加，因此在不冻结视频的情况下加入了新的主体。图。 15 展示了相关示例。切换后，基础模型会保留初始场景；而采用残差引导时，新的主体会在最初的几个块内出现（例如草地上的奶牛、单板滑雪者旁边的熊、大象旁边的狮子），同时保留原始主体、其动作以及背景。绵羊示例中的热气球出现得又小又晚，这反映了添加占据画面较小部分的物体所面临的困难。

表9。**消融实验**。在 Self Forcing (SF) 和 LongLive (LL) 上，从 T2V-CompBench 动态属性集随机抽取的 100 对样本。默认设置：$\tau = 850, \lambda = (1, 0。5, 0, 0)$，并在提示词区间内共享一个 ϵ。(a) 噪声水平 τ。(b) 步数调度 λ。(c) 噪声共享。

(a) 噪声水平 τ

	成就		质量		主体	
τ	SF	LL	SF	LL	SF	LL
none	0.130	0.350	0.789	0.770	0.988	0.989
700	0.260	0.425	0.791	0.778	0.975	0.984
800	0.468	0.550	0.787	0.768	0.962	0.982
850	0.623					
900	0.779	0.775	0.790	0.771	0.954	0.979
1000	0.883	0.850	0.795	0.772	0.956	0.975

（b）步数调度 λ

	成就		质量		主体	
λ	SF	LL	SF	LL	SF	LL
(1, 0, 0, 0)	0.468	0.625	0.789	0.774	0.963	0.981
(1, 0.5, 0,0) (						
(1, 1, 0, 0)	0.753	0.838	0.778	0.771	0.942	0.969
(1, 1, 1, 0)	0.805	0.850	0.753	0.738	0.866	0.896

(c) 噪声共享 ϵ

	成就		质量		主体	
ϵ	SF	LL	SF	LL	SF	LL
每块	0.675	0.662	0.786	0.777	0.939	0.976
每次切换 0						



图 15**T2V-CompBench 动作绑定的定性结果**。白到黑的色条标示提示切换，此时提示词中加入了第二个主体。没有残差引导时，视频会保留初始场景；使用残差引导时，新主体会在切换后的最初几个块中出现，同时保留原有主体和背景。

表 10. **T2V-CompBench 动作绑定上的提示切换**。 这些提示词涉及在切换时添加第二个主体。指标遵循表 1，并以 T2V-CompBench 官方动作绑定分数（归一化至 $[0, 1]$）替代动态属性得分。

方法	**达成度** ↑	**T2V-CB 动作绑定** ↑	**CLIP** ↑	**动态** ↑	**质量** ↑	**总体** ↑
MemFlow [22]	0.386	0.524	26.39	0.680	0.836	0.746
+RG	0.605	0.624	28.00	**0.860**	**0.839**	0.793
LongLive [56]	0.450	0.564	27.24	0.665	0.838	0.761
+RG	**0.667**	**0.627**	**28.56**	0.810	0.837	**0.803**

E.2. 特殊情况：滚动强制

不同于分块模型，滚动强制 [34] 会对处于交错噪声水平的 T=4 个块组成的窗口进行去噪，并在每次前向传播中将整个窗口推进一步。在每次传播后，前端块以干净状态离开窗口并写入 KV缓存，同时一个纯噪声块从后端进入。因此，一个块会接收到它的首次预测

当其进入窗口时，且公式（4）的步骤索引对应于其从后往前的位置。残差引导按如下方式映射到此调度中。在每个切换后块 k 进入窗口之前， s_{k-1} 都取作最近离开窗口的干净块，同时也是最后写入缓存的块。随后对其重新加噪至τ=850，并在新提示词下恢复一次，而不使用

表11。 **残差引导** **滚动强制**。 100 个 T2V-CompBench 动态属性随机提示对。指标参照表 1 .

方法	达成度 ↑	T2V-CB动态属性 ↑	CLIP ↑	动态 ↑	质量 ↑	总体 ↑
滚动强制 [34]	0.214	0.042	25.58	0.925	**0.827**	0.704
+RG	**0.476**	**0.095**	**26.11**	**0.935**	0.821	**0.752**

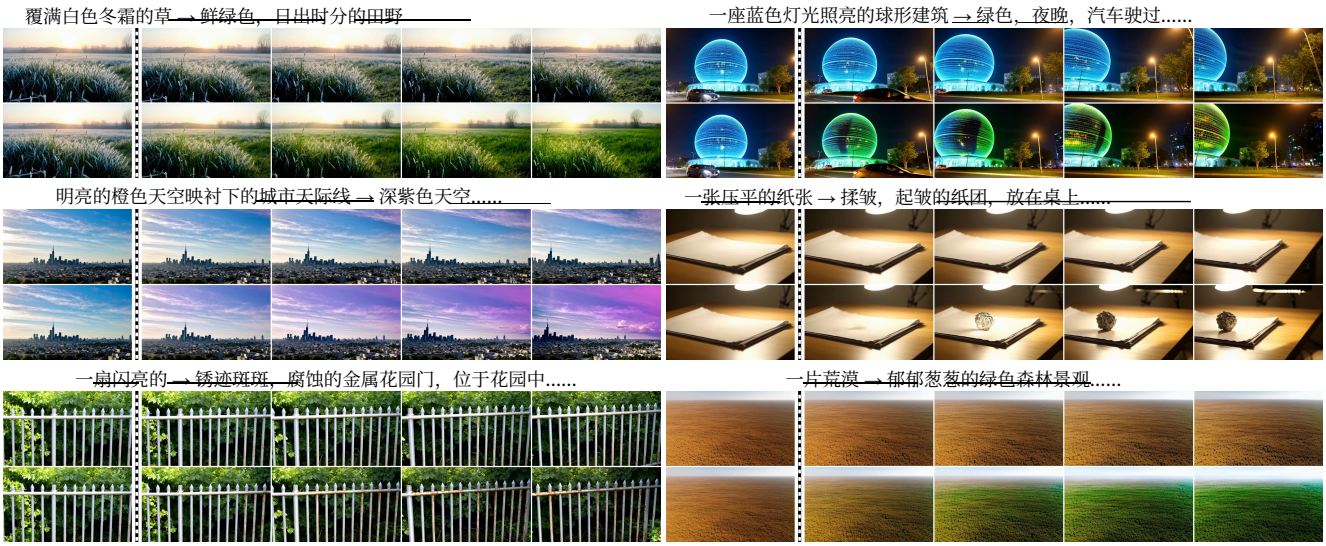


图16。 **定性结果** **关于滚动强制**。 T2V-CompBench 动态属性。在每个示例中，顶行是滚动强制 g
底行是滚动强制 搭配残差引导。黑白渐变条标示提示切换。

缓存，从而得到 r_k ，如公式（3）所示。随后根据窗口位置应用残差：当 $\lambda_1 r_k$ 位于窗口末端时，将其添加到块 k 的预测中；当其处于下一个位置时，则添加$\lambda_2 r_k$ ，因此每个块在经过窗口时，都会恰好一次接收分块模型场景的调度。切换时仍位于窗口中的先前提示词块保持不变，从而保留切换前的帧。对于第一个切换后块，在新提示词下尚无块离开窗口，因此使用前端块的预测作为 s_{k-1}。表11报告了 T2V-CompBench 动态属性上的结果。残差引导将达成度从 0.214 提升至 0.476，同时质量与基础模型的差距保持在 0.006 以内，总体则从 0.704 提升至 0.752。图16 展示了定性示例：切换后，基础模型仍保持初始状态，而残差引导在前几个块内实现了变化（例如，霜融化成绿草，纸张揉皱成球），同时布局和摄像机保持连续。需要注意的是，只有当切换时已位于窗口中的块离开窗口后，切换才会实现，因此变化出现的时间比在分块模型中晚一到两个块。

表 12。**LongLive 上的 240 秒生成**。使用和不使用残差引导时，LongLive 每分钟的 CLIP 和 VBench 质量，基于 50 个脚本取平均值。

	CLIP ↑		质量 ↑	
分钟	LongLive	+RG	LongLive	+RG
0–1	27.34	**27.90**	**0.813**	0.811
1–2	26.67	**27.43**	**0.808**	0.804
2–3	25.95	**27.15**	**0.810**	0.805
3–4	25.84	**26.99**	**0.809**	0.801

E.3. 超长生成（240 秒）

为测试这种提升在长时间连续生成中是否仍然存在，我们将 50 个 InteractiveBench 脚本从 60 秒延长至 240 秒。每个脚本保留原有的六条提示词，并加入由 Qwen3-VL-32B 编写的 18 条后续提示词。这些提示词保留场景设定、角色和视觉风格，每条提示词引入一个可见事件或状态变化，并像原始基准测试一样完整地重新描述场景。因此，每段视频包含 24 条提示词，每条约 10 秒，共有 23 次切换。我们使用 LongLive，在启用和不启用残差引导的情况下，按照第5节的配置进行生成，且两种条件使用与 60 秒运行相同的随机种子。遵循 LongLive [56]，的做法，我们按分钟分组报告每个提示词片段的 CLIP 分数，并同时报告 VBench质量分数。

表 12显示，CLIP 分数的提升贯穿整段视频，并且随时间推移不断扩大：从 +0.56 第一分钟的 +1.15增加至最后一分钟的数值。随着视频变长，基础模型与每条新提示词的偏离程度进一步加大，分数从 27.34 降至 25.84；而残差引导能持续跟随提示词，分数仅降至 26.99。每分钟的质量都保持在基础模型 0.008 以内，这表明反复切换提示词对测得的质量维度造成的额外退化很小。

F. 讨论

F.1. 与先前工作的关系

与先前分析的关系。近期有多项工作报告了类似的历史诱导抑制现象，但主要通过缓存或记忆设计来应对。LongLive [56]发现，保留KV缓存会延迟或阻碍提示切换，并将其归因于先前提示留下的残余信息；为此，该工作引入了KV重新缓存。SWIFT [47] 同样指出，缓存的视频历史与更新后的提示语义之间存在不匹配，并提出语义注入和自适应记忆分配。Echo-Forcing [54] 发现过时的场景记忆会造成干扰，并通过结构化场景记忆和差异感知的记忆衰减来应对。在相关但有所不同的动作条件世界建模领域，Astra [64]研究了较长的视觉历史如何降低动作响应性，将这一现象称为视觉惯性，并通过噪声增强的历史记忆缓解这一问题。

另一类工作在固定提示词下考察自回归视频扩散的去噪步骤。Bai等人 [2] 发现，内容和运动大多在第一个去噪步骤中确立，并将此归因于历史上下文。HiAR [65]将少步学生模型的第一步与运动动力学和样本多样性联系起来。In-Context Forcing [55] 进一步表明，Self Forcing在第一步时对前一帧的关注强于对当前帧的关注，并将其与运动动力学减弱联系起来。这些研究关注的是运动、多样性或时间漂移，而非提示切换响应能力，并且主要通过学生模型再训练来应对其发现的问题。

我们的分析补充了这两条研究路线。我们剖析了保留上下文中的视觉内容如何抑制提示词引起的预测变化，即使在新提示下重新缓存后也是如此。我们还测量了每个去噪步骤的提示响应，并表明提示词作用最强的步骤，也是上下文抑制它最强的步骤，因此后续步骤无法恢复切换。即使只有一个前置块，这种抑制也已相当显著。综合来看，这些发现指出了免训练校正应当作用的位置，并为残差引导（RG）的设计提供了依据。

与基于引导的方法的关系。表13比较了残差引导与它最为相似的三种引导方案；这四种方法都可以表示为添加到切换块预测上的校正。关键区别在于构建各自校正时所用的参考：CFG 和 MAGI-1 在缓存上下文下测量提示方向，历史引导则强化上下文，而残差引导在没有缓存上下文的情况下估计新提示造成的位移。

具体而言，无分类器引导 [17] 在每一步都添加 $(w-1)[\hat{x}(\mathcal{C},y)-\hat{x}(\mathcal{C},\varnothing)]$，在上下文下测得的提示方向。历史引导 [45]添加 $(w-1)[\hat{x}(\mathcal{C},y)-\hat{x}(\varnothing,y)]$，上下文方向，从而增强上下文而非提示词；它要求使用经过历史丢弃训练的模型，以便定义 $\hat{x}(\varnothing,y)$。MAGI-1 [49]通过使用独立的缩放因子将两者结合起来，$(w_{\text{text}}-1)[\hat{x}(\mathcal{C},y)-\hat{x}(\mathcal{C},\varnothing)]+(w_{\text{prev}}-1)[\hat{x}(\mathcal{C},\varnothing)-\hat{x}(\varnothing,\varnothing)]$，并提高 w_{prev} 以减少块之间的闪烁，从而恰好增强抑制切换的上下文。其提示方向与无分类器引导中的相同，均在上下文下测量；而在蒸馏版 MAGI-1 中，由于学生模型没有空提示词，该方向被禁用。第 3.2 节表明，该方向会在上下文下缩小，因此，实现切换所需的缩放因子会在切换发生前使预测饱和。我们将在补充材料 F.2 中进一步讨论这一点。

残差引导在其所添加的量上有所不同。我们使用的方向 $r_k=\hat{x}(\varnothing,\ y_{\text{new}})-s_{k-1}$ 是在无上下文条件下测量的，因此并不是 CFG 和 MAGI-1 的提示方向的重新缩放，后者是在上下文条件下测量的。历史引导同样会评估无上下文预测，但它是在当前噪声样本上进行的，并将其作为偏离的目标点，从而强化上下文；而残差引导则是在重新加噪的前一块上进行评估，并将预测结果向其靠近。它参考的是前一干净块，而不是当前噪声样本，因此每个块只需计算一次，而不是在每个去噪步骤中都计算。其幅度由噪声水平 τ 决定，而其在每个去噪步骤中的贡献则由 λ_i 控制，而不是由 CFG 缩放系数控制。无上下文查询由学生模型自身获得；学生模型在每个视频的第一块上运行时不使用缓存，因此不需要无条件分支，也不需要训练。

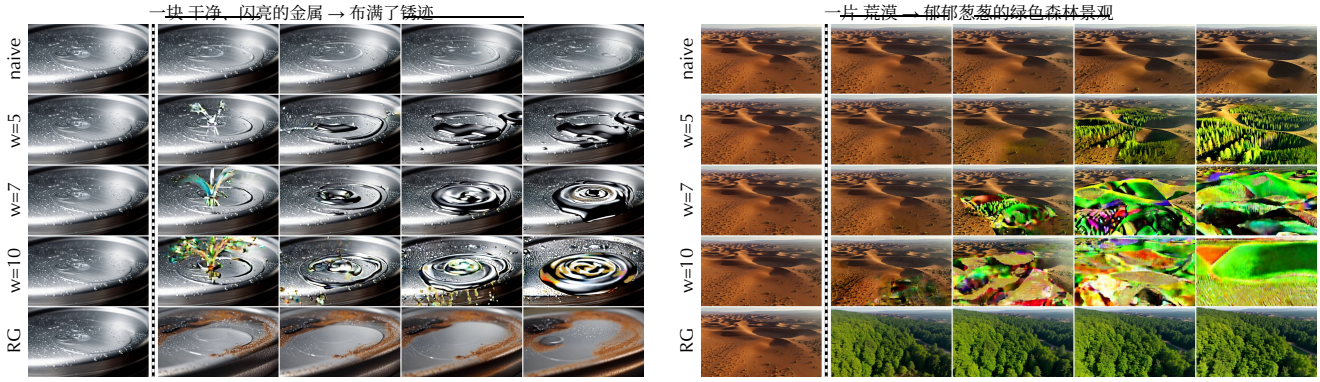
F.2. 蒸馏学生模型上的 CFG

考虑上下文后，在切换块第一步的无分类器方向幅度 $\|\hat{x}(\mathcal{C},y_{\text{new}})-\hat{x}(\mathcal{C},y_{\varnothing})\|$ 比不考虑上下文时小 3 到 6 倍（在五个模型上的平均值为 4。7×见表14）。因此，要放大这一方向，需要使用较大的尺度，提示词才能生效。

残差引导的一种自然替代方案，是放大模型已经提供的提示方向，

表 13. **与现有基于引导的方法进行对比。** 无分类器引导（CFG）、历史引导（HG）和 MAGI-1 引导都会在每一步对同一个噪声样本的预测进行对比。我们的残差引导（RG）则将无上下文预测与前一个干净块进行对比，每个块仅需一次，且无需无条件分支或训练。

	CFG [17]	HG [45]	MAGI-1 [49]	残差引导（我们的）
增强	提示词	上下文	上下文和提示词	提示词优先于上下文
对比	提示词与无	上下文与无	三个分支	无上下文 vs. s_{k-1}
参考	噪声样本	噪声样本	噪声样本	干净块
注入	每一步	每一步	每一步	早期步骤
控制	缩放 w	缩放 w	缩放比例 $w_{\text{上一项}}, w_{\text{text}}$	噪声水平 τ
需要	无条件分支	丢弃训练	无条件分支	none



(a) Self Forcing (b) LongLive

图17. **在切换块上应用无分类器引导。** 对 (a) Self Forcing 和 (b) LongLive 应用 CFG。朴素方法 表示基础模型。对于这两个模型，CFG 都能部分实现切换，但会产生伪影；而残差引导则能在不产生伪影的情况下实现切换。

表 14. **上下文会缩小无分类器方向。**$\|\hat{x}(\mathcal{C}, y_{\text{new}}) - \hat{x}(\mathcal{C}, y_{\varnothing})\| / \|s_{k-1}\|$ 在切换块的第一个去噪步骤中，不使用上下文（$\mathcal{C} = \varnothing$）与使用完整上下文时，在第 3 节的 100 个切换对上的结果（均值，自助法置信区间 95%）。LongLive 的测量是在重新缓存后进行的。3 (均值，95% 自助法置信区间)。LongLive是在重新缓存后测量的。

模型	无上下文	有上下文	比率
Self Forcing	0.907 [0.882, 0.933]	0.157 [0.140, 0.175]	5.8×
Causal Forcing	0。774 [0.746, 0.801]	0.253 [0.228, 0.277]	3.1×
LongLive	0.892 [0.869, 0.915]	0.182 [0.164, 0.200]	4.9×
MemFlow	0.926 [0.901, 0.950]	0.190 [0.171, 0.210]	4.9×
Infinity-RoPE	0.827 [0.806, 0.848]	0.176 [0.155, 0.195]	4.7×

$\hat{x}(\mathcal{C}, y_{\text{new}}) - \hat{x}(\mathcal{C}, y_{\varnothing})$，缩放系数 w 与无分类器引导中的相同。该方向是在上下文条件下测量的。在切换块的第一步中，其幅度比无上下文对应项小 3 至 6 倍（4。7×在五个模型上的平均值，表14），因此，要将预测移动到切换所要求的程度，所需的缩放系数很大。图 17 展示了 $w \in \{5, 7, 10\}$ 在 Self Forcing 和 LongLive 上的结果。在 w=5 切换仅部分实现，因为金属仍大多保持洁净，而森林只出现在沙漠边缘。在 w=7 和w=10预测被推得足够远而发生改变，但相同的尺度会放大该方向的每个分量，包括与切换无关的分量；在到达新提示之前，帧中就会出现饱和色彩和几何结构破损。由于方向中有用的部分与伪影会一同增长，因此不存在能够干净利落地实现切换的中间尺度。相反，残差引导会添加一个在没有上下文的情况下测量得到的方向，因此无需放大就足够大，从而无需大引导尺度所带来的伪影即可实现切换，同时基本保留场景。

F.3. 替代方案

设置。 第3 节提出了一个问题：除了添加残差之外，是否还有其他方法可以削弱上下文。我们在 T2V-CompBench 动态属性集随机选取的 100 对样本上，将各种替代方案与 Self Forcing 和 LongLive 进行比较（图18）。除非另有说明，所有替代方案都与残差引导（RG）作用于相同的块，即从切换块到视频结尾，并在每个去噪步骤中执行。除了达成度和质量，我们还报告切换前后的场景变化：计算切换前后帧之间的 DreamSim 距离，并遮蔽变更对象，因此方法会因改变背景而受到惩罚，而不会因实现提示词而受到惩罚。

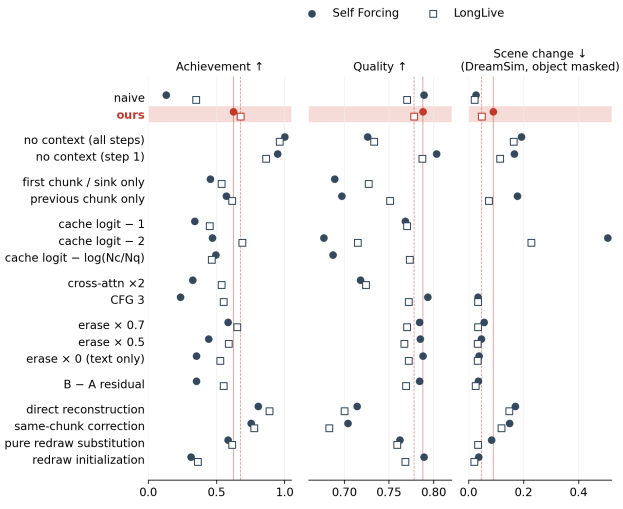


图18. **残差引导的替代方案**。 在基础模型、残差引导（阴影部分）以及通过削弱上下文、增强提示词、改变擦除强度或以其他方式使用无缓存重建（提示词差分（B–A）残差、直接重建、同块校正、纯替换、重绘初始化）的替代方案之间切换时，使用 Self Forcing（圆圈）和 LongLive（方块）对100对动态属性进行评估，指标包括达成度、质量和场景变化（物体掩码 DreamSim）。竖线标示残差引导。右侧面板中未标记的变体未测量场景变化。朴素方法 表示基础模型。

削弱上下文。 无上下文 会移除对缓存的自注意力，可以在每一步执行，也可以仅在第一步执行。仅第一块 和仅前一块将缓存缩减为第一块（LongLive 中的汇点），或当前块之前的那一块。注意力降权会从缓存键的注意力对数几率中减去一个常数 b，其中 $b \in \{1, 2\}$ 和 $b =$ 为 $\log(N_{\text{cache}}/N_{\text{cur}})$，从而平衡缓存与当前块的注意力权重。

增强提示词。 交叉注意力 $\times 2$ 将文本交叉注意力的输出加倍。*CFG3* 对蒸馏学生模型应用无分类器引导，$\hat{x}(\mathcal{C}, y_{\text{new}}) + 2\,[\hat{x}(\mathcal{C}, y_{\text{new}}) - \hat{x}(\mathcal{C}, y_{\varnothing})]$。

残差引导的变体。令 $\hat{s}(y) = \hat{x}_\theta(\tilde{x}^{\tau}_{k-1} \mid \varnothing, y)$ 表示在提示词 y 下对前一个块进行的无缓存重建，因此 RG 会添加 $r_k = \hat{s}(y_{\text{new}}) - s_{k-1}$（式（3））。擦除缩放将此残差拆分为文本项和擦除项，$r_k = [\hat{s}(y_{\text{new}}) - \hat{s}(y_{\varnothing})] + \gamma\,[\hat{s}(y_{\varnothing}) - s_{k-1}]$，其中 $\gamma \in \{0, 0.5, 0.7\}$。设定 $\gamma = 1$ 即可恢复 RG，而 $\gamma = 0$ 则仅保留文本项。提示差分残差（B–A）将 s_{k-1} 替换为使用先前提示词进行的重建，$r_k = \hat{s}(y_{\text{new}}) - \hat{s}(y_{\text{old}})$。最后四种变体使用无缓存重建 $\hat{s}(y_{\text{new}})$，但不将其作为残差。直接重建将 $\hat{s}(y_{\text{new}})$ 本身作为引导块输出。同分块校正将 r_k 添加到使用上下文可见状态生成的重建中，而不是添加到当前块中。纯替换 使预测朝重建结果移动，$\hat{x} + \lambda_i\,[\hat{s}(y_{\text{new}}) - \hat{x}]$ 与 $\hat{x} = \hat{x}_\theta(x_k^{t_i} \mid \mathcal{C}_{<k}, y_{\text{new}})$，使用 RG 的步数调度。重绘初始化 从 $\hat{s}(y_{\text{new}})$ 开始，将其重新加噪至 τ，同时保留缓存且不使用残差。

结果。 移除上下文后，达成度高于残差引导，但代价是场景保留效果变差。每一步都移除上下文时，质量会下降，场景变化幅度约为残差引导的 2 至 3.5 倍。仅在第一步移除上下文时，质量得以维持，但场景变化仍约为残差引导的两倍。无论保留第一个块还是前一块，达成度和质量都不及残差引导；而只保留前一块时，场景变化也比残差引导更大。注意力降权不稳定。使用 $b = 1$时，达成度几乎没有变化；使用 $b = 2$ 时，在 LongLive 上与残差引导相当，但在 Self Forcing 上会破坏场景；而平衡值则介于两者之间，且会损失质量。将交叉注意力加倍和应用 CFG 都只能使达成度提升不到 0.2；此外，将交叉注意力加倍还会损失质量。

降低擦除项的权重会使达成度单调下降。仅使用文本项时（$\gamma = 0$），Self Forcing 上的达成度仅为 0.351，LongLive 上仅为 0.525，只恢复了残差引导相较于基础模型所带来增益的大约一半（表 3）。提示差分残差（B–A）同样只保留提示方向，其表现也远低于残差引导。因此，即便不借助缓存来衡量，单独使用提示方向仍然不够；消除上下文施加的影响，其作用与增强提示词同样重要。较小的 γ 能在质量不变的情况下减少场景变化，而 $\gamma = 0.7$ 的达成度接近残差引导，但任何 $\gamma \neq 1$都会引入额外的超参数和一次额外的空提示词传递。在 $\gamma = 1$ 时，空提示词项会相互抵消，因此残差引导只需额外传递一次，且不需要额外权重。

重建如何进入块，与重建本身同样重要。直接输出无缓存的重建，或校正一个上下文可见的重建，都会提高达成度，但质量损失与移除上下文一样多，场景变化也相应增加。从重建开始能够保留场景，却几乎不会改变内容，因为缓存会将该块拉回前一状态。纯替换在达成度和质量两方面都不如残差引导，而加入相同信息作为残差，则能保留上下文感知预测的细节。因此，残差引导是最简单的变体，既能显著提高达成度，又能将质量维持在.....的水平



图19。 **替代方案的定性比较**。 对于每次切换，第一行显示切换前的帧，第二行显示切换后最后一块中的帧；上方为 Self Forcing，下方为 LongLive。只有残差引导（RG）在保留场景的同时呈现了新提示词。移除上下文或直接重建区块会重绘整个场景，前一块只会改变场景或错误的属性，交叉注意力缩放和同分块校正会引入伪影，而缓存logit、CFG 和重绘初始化会保留旧状态。定量比较请参见图 18 。

基础模型，且场景变化较小。

定性比较。 图 19 展示了两次切换：在 Self Forcing 中，一面光秃的岩石峭壁应变成白雪覆盖的峭壁；在 LongLive 中，一朵绿色玫瑰和一朵紫罗兰色郁金香应变成棕色和亮橙色。残差引导在同一山坡上添加了积雪，云层此时覆盖山脊，并为同样的两朵花重新着色，同时摄像机和布局保持切换前的样子。无论是在第一步还是每一步移除上下文，都能达到新提示词的要求，但画出的会是另一座山和另一束花。直接重建也会重新绘制场景，而且会调换花朵的位置。仅使用前一块，就会把这座山替换成另一座雪峰，并将花朵染成黄色和洋红色。将交叉注意力和同分块校正加倍会留下明显的伪影。将缓存logit降低 1 并重新初始化会使两段视频都保持旧状态，而 CFG 在 w=3只改变玫瑰的中心。在这些变体中，只有残差引导遵循新提示词并大体保留场景，这与图 18 中展示的权衡相符。

F.4. 场景切换

我们基准测试中的一些提示词已经涉及场景切换，就像电影剪辑中的场景切换一样。例如，在图12中，洒出的水形成了一条河流，随后一艘船在河流上航行。在这里，我们通过保持主体不变并替换整个场景的提示词，直接检验这一场景。我们将残差引导应用于启用 RoPE Cut 的 Infinity-RoPE；RoPE Cut 是其专为场景切换设计的机制。如图20所示，即使启用了 RoPE Cut，Infinity-RoPE 也经常停留在前一个场景中，而残差引导则会在每次切换时转移到新场景，同时保留主体。由于场景切换所需的变化幅度大于场景内切换，因此我们使用更高的噪声水平 $\tau = 950$，我们发现，与默认值 $\tau = 850$相比，这样能更可靠地替换场景。

F.5. 局限性与未来工作

图 21 展示了残差引导（RG）的代表性失败案例，可分为两类。

提示响应不足。 在左列中，无论是否使用残差引导（RG），切换都未能实现。云朵仍然留在天空中，房间只变得略微更暗，狗也没有把头放在爪子上。这些提示词要求移除一个物体、改变整体照明，或改变细粒度姿态，同时场景的布局保持不变。我们将这些失败归因于残差的获取方式。前一块只被重新加噪到中间水平，因此其粗略结构和总体颜色在重建过程中得以保留，而对于这类变化，所得残差很小。特别是，被移除后没有其他物体取代其位置的物体属于这种粗略结构的一部分，因此往往会保留。更高的噪声水平会增强响应，但也会改变场景（第 5.3节）。第三个示例来自 InteractiveBench，其中连续的提示词共享大部分描述内容，这进一步减小了新提示词能够引起的差异。

非预期的变化。 在右栏中，切换虽然实现了，但同时伴随着提示词未要求的变化。冰淇淋变成了坚实的冰淇淋球，但勺子和一只手进入了帧中。果汁变成了绿色，但玻璃杯多了一个把手。行星出现了，但杯子本身变成了一颗带环行星。由于残差是在没有缓存的情况下估计的，新提示词未明确说明的内容不会受到前面帧的约束，模型的先验信息可以通过残差进入下一个块中

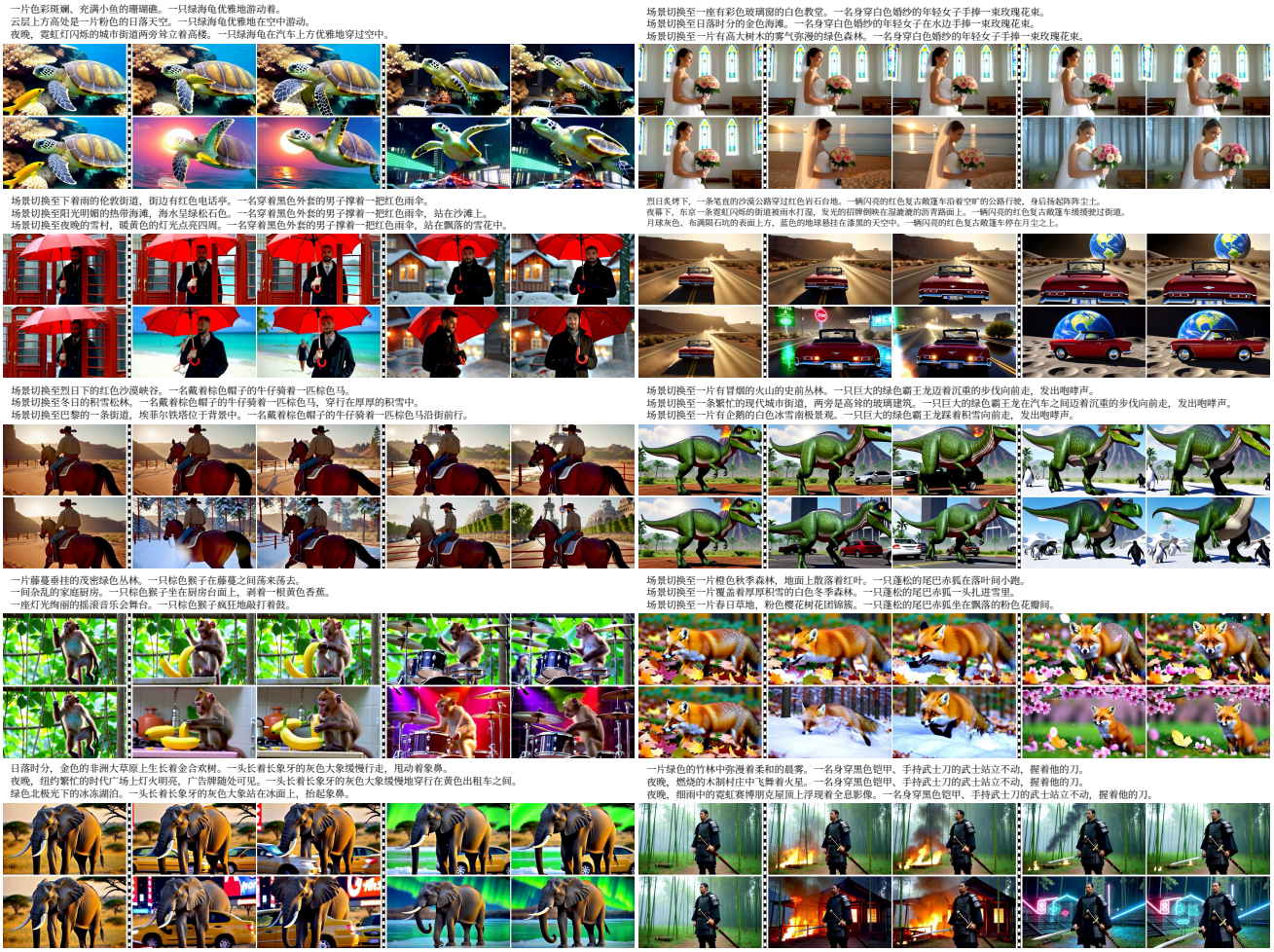


图 20. **Infinity-RoPE 上的场景切换**。每个面板展示了三个连续的提示词，这些提示词保留主体并替换场景。上排为 Infinity-RoPE，下排为加入残差引导（RG）的 Infinity-RoPE，二者使用相同的随机种子生成。虚线标示提示切换。Infinity-RoPE 往往停留在第一个场景中，或仅部分更新场景；而残差引导（RG）会在切换时转向每个新场景，同时保留主体。

这一估计中，而模型的先验信息可以通过残差进入下一个块中。这在第三个示例中最为明显，因为新提示词没有提到杯子。

综合来看，这些示例表明，响应性与保留之间仍存在权衡：校正力度较弱可能无法捕捉细微变化，而选择性不足的校正则可能改变提示词之外的内容。未来工作可探索对残差的强度和空间范围进行提示感知自适应，以更好地平衡这两个目标。



图 21. **失败案例**。每个示例中，上排为基础模型，下排为残差引导。第一帧是切换前的最后一帧，虚线条标示提示切换。**左：提示响应不足**。请求移除物体（LongLive）、改变光照（SWIFT）或改变姿势（MemFlow、InteractiveBench）的操作均未实现。**右：非预期变化**。切换已实现，但出现了勺子和一只手（Anchor Forcing），玻璃杯长出了把手（Self Forcing），或者杯子本身变成了一颗行星（Anchor Forcing、StoryEval）。