StreamingT2V:一致、动态、可扩展
从文本生成长视频

Roberto Henschel1****、 Levon Khachatryan1**** 、Hayk Poghosyan1**** 、Daniil Hayrapetyan1**** 、Vahram Tadevosyan1 ‡ 、Zhangyang Wang1,2 、Shant Navasardyan1 、Humphrey Shi1,3 Picsart AI Research (PAIR) 2UT Austin [1]3Georgia Tech https://github.com/Picsart-AI-Research/StreamingT2V

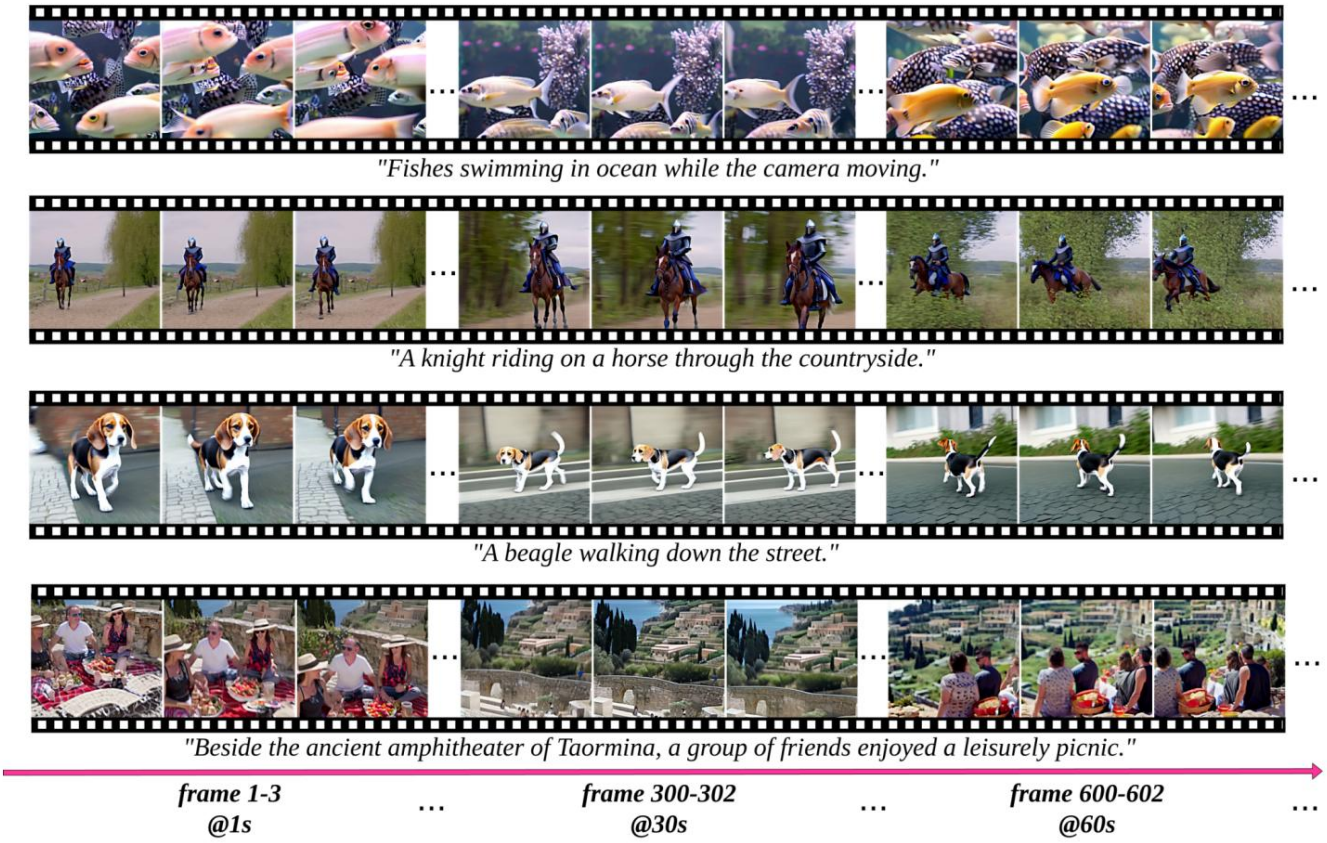


图 1. StreamingT2V是一种先进的自回归技术,用于创建具有丰富运动动态的长视频,确保时间一致性、与描述性文本对齐、高帧级图像质量且无卡顿。演示包括高达 1200 帧、时长 2 分钟的视频,并且可进一步延长。StreamingT2V 的有效性不受所用 Text2Video 模型的限制,这表明改进基础模型后,视频质量有望进一步提升。

抽象的

文本到视频的传播模型能够生成遵循文本指令的高质量视频,简化了制作多样化和个性化内容的过程。帐篷。目前的方法擅长生成短视频（最长 16 秒）,但当单纯扩展到长视频合成时,会产生硬切换。为了克服这些限制,我们提出了 StreamingT2V,这是一种自回归方法,可以生成最长 2 分钟或更长的无缝过渡长视频。其关键组件包括:(i) 一个称为条件注意模块 (CAM) 的短期记忆块,它通过一种注意机制,根据从前一个块中提取的特征来调节当前生成。

*同等贡献。 †当前所属机构:Moonvalley。 ‡当前所属机构:Superside。
联系人:Roberto Henschel <firstname@moonvalley.com>,Levon Khachatryan、Hayk Poghosyan、Daniil Hayrapetyan:<firstname.lastname@picsart.com>,以及 H. Shi。

现象,导致一致的块转换,(ii)一个称为外观保存模块的长期记忆块

(APM),提取高级场景和对象特征
从第一个视频块开始,以防止模型忘记初始场景,以及 (iii) 随机混合

允许自回归应用的方法
无限长视频的视频增强器,确保
跨块一致性。实验表明,Stream-ingT2V 可以产生更多运动,而竞争方法

当单纯地以自回归方式应用时,会出现视频停滞的情况。因此,我们提出使用 StreamingT2V
高品质无缝文本转长视频生成器,在一致性和动感方面超越竞争对手。

1. 简介

扩散模型[14, 27, 29, 34]的出现
引发了人们对文本引导图像合成的极大兴趣
和操作。基于图像生成方面的成功,它们已经扩展到文本引导的视频生成
[3,4,6,10-12,16,17,20,32,37,39,44] 。

尽管生成质量和文本对齐效果令人印象深刻,但大多数现有方法,如[3、4 、

16、24、39、44、48]主要侧重于产生短
帧序列(通常为 16、24 或最近的 384 帧长度)。然而,短视频生成器在以下方面受到限制:
现实世界的用例,例如广告制作、讲故事等。
长时间训练视频生成器的简单方法
视频(例如≥1200帧)通常是不切实际的。即使对于
生成短序列,通常需要昂贵的
训练(例如 260K 步和 4.5K 批量大小,以便
生成16帧[39])。
因此,一些方法[4,16,23,48]扩展了基线
通过自回归生成基于
前一个块的最后一帧。然而,简单地将视频块的噪声潜伏期与最后一帧连接起来
前一个块的帧会导致条件不佳,场景转换不一致(参见第A.3 节)。一些

作品[3, 7, 40, 42, 47]还集成了 CLIP [26]前一个块的最后一帧的图像嵌入,这
稍微提高了一致性。然而,它们仍然容易
跨块不一致(见图A.7) ,因为
CLIP 图像编码器丢失必要的关键信息
用于准确重建条件帧。
并发工作 SparseCtrl [11]利用稀疏编码器
用于调节。为了匹配输入的大小,其架构需要连接额外的零填充帧

在插入稀疏
编码器。然而,输入的这种不一致导致
输出不一致(参见第5.2 节)。
我们的实验(见第5.2 节)表明,事实上所有评估的图像到视频方法都会产生视频停滞
或应用自回归时质量严重下降

通过调节前一块的最后一帧。
为了克服当前作品的弱点,我们
提出StreamingT2V,一种自回归文本到视频
配备长期/短期记忆块的方法
生成没有时间不一致的长视频。
为此,我们提出了条件注意模块(CAM) ,由于其注意力特性,它可以有效地
借用前几帧的内容信息
产生新的,同时不限制它们的运动
之前的结构/形状。感谢 CAM,我们的结果
流畅且无伪影的视频块转换。
当前的方法不仅表现出时间不一致和视频停滞,而且还会经历改变

在物体外观/特征方面(例如参见SEINE [6]
图A.4)以及视频质量随时间推移而下降(例如
SVD [3]如图5 所示)。这是因为只有
前面的块被考虑,从而忽略了自回归过程中的长期依赖关系。为了解决

本期我们设计了一个外观保存模块
(APM)从中提取对象和全局场景细节
初始图像,用该信息来调节视频生成,确保物体和场景特征的一致性

在整个自回归过程中。
为了进一步提高我们长期
视频生成,我们采用视频增强模型
自回归生成。为此,我们应用 SDEdit
[22]基于高分辨率文本到视频模型的方法和
增强连续的 24 帧块(与 8 帧重叠)
帧)。为了使块增强过渡平滑,我们设计了一种随机混合方法

用于无缝合并重叠块。
实验表明,StreamingT2V 可以生成长
以及由没有视频的文本生成的时间一致的视频
停滞。总而言之,我们的贡献有三方面:

- 我们引入了StreamingT2V,一种自回归方法
  用于无缝合成扩展视频内容
  短期和长期依赖关系。
- 我们的条件注意模块 (CAM)和外观保存模块 (APM)确保自然

  全局场景和对象特征的连续性
  生成的视频。
- 我们通过引入连续重叠块的随机混合方法来无缝增强生成的长视频。

2.相关工作

文本引导的视频传播模型。生成视频
使用扩散模型[14, 33]从文本说明中
引入新成立且活跃的研究领域
通过视频扩散模型(VDM) [16]。该方法可以
仅生成低分辨率视频(最高 128x128)
最多 16 帧(无自回归),施加了很大的限制,同时需要大量的训练

资源。因此,有几种方法采用空间/时间
上采样[4, 15, 16, 32],使用最多 7 个增强器模块的级联[15]。虽然这可以实现高分辨率
和长视频,生成的内容仍然受到
关键帧中描绘的内容。

为了生成更长的视频(即更多关键帧),
Text-To-Video-Zero (T2V0) [17]和 ART-V [41]利用
文本到图像的扩散模型。因此,他们可以生成
仅简单运动。T2V0 在其第一帧上通过
跨框架注意力机制和 ART-V 在锚框架上的应用。
缺乏全局推理会导致不自然或重复的动作。MTVG [23]和 FIFO-Diffusion [18]
将
文本到视频模型转化为自回归方法
一种无需训练的方法。由于它在块内和块之间使用强一致性先验,因此运动效果较低
量较大,背景大多接近静态。FreeNoise [25]
采样一小组噪声向量,重新用于
所有帧的生成,而时间注意力则在局部窗口上执行。由于时间注意力对于这种帧的混洗是不变的,因此它导致了较高的相似性

在帧之间,几乎总是静态的全局运动和
近乎恒定的视频。Gen-L [38]生成重叠的
短视频并通过时间协同去噪将它们组合起来,
这会导致视频质量下降和停滞。
最近的基于变换的扩散模型[24,48]
在 3D 自动编码器的潜在空间中,能够生成最多 384 帧。尽管经过大量训练,这些

模特制作的视频动作有限,通常会导致
在近乎连续的视频中。

图像引导视频传播模型作为长视频
生成器。一些作品影响着视频的生成
通过驾驶图像或视频[3、5-7、9、11、21、28、40、42、43 、
47]. 因此它们可以转化为自回归方法
通过调节前一块的帧。

VideoDrafter [21]采用锚帧(来自文本到图像模型)并调节视频扩散模型

在其上独立生成多个可共享的视频
相同的高级上下文。然而,这会导致场景切换剧烈,因为视频块之间没有强制一致性。StoryDiffusion [49]的条件是,

从关键帧开始线性传播,这导致
导致严重的质量下降。一些研究[6, 7, 43]将(编码的)条件(例如输入帧)连接起来
带有附加掩码(指示提供的帧)
到视频传播模型的输入。

除了将条件作用连接到扩散模型的输入之外,一些研究[3, 40, 47]还替换了

扩散交叉注意力中的文本嵌入
CLIP 模型[26]图像嵌入条件
帧。然而,根据我们的实验,它们在长视频生成中的适用性有限。SVD [3]显示

随着时间的推移,质量会严重下降(见图5),并且
I2VGen-XL [47]和 SVD [3]经常生成不一致的
块之间的联系,仍然表明条件
机制太弱。

一些作品[5,42],例如DynamiCrafter-XL [42] ,因此
在每个文本交叉注意力模型中添加一个图像交叉注意力模型,
这虽然可以提高质量,但块之间仍然经常出现不一致的情况。

并行工作 SparseCtrl [11]添加了 ControlNet
[45]像分支一样将条件
帧和帧掩码。根据设计,它需要附加
在条件帧中添加额外的黑帧。这种不一致性很难由模型补偿,导致

帧与帧之间场景切换频繁且剧烈。

3. 准备工作

传播模型。我们的文本到视频模型,我们称之为
StreamingT2V是一个在 VQ-GAN [8, 35]自动编码器 D(E(·))的潜在空间中运行的扩散模型,
其中 E 和 D 分别是相应的编码器和解码器
编码器。给定视频 $V \in \mathbb{R}^{\text{宽}\times\text{高}\times\text{宽}\times 3}$,和-
由空间分辨率为 H × W 的 F 帧构成,其潜在编码 $x_0 \in \mathbb{R}^{\text{高度}\times\text{高度}\times\text{宽度}\times\text{高度}}$ 通过逐帧获得
编码器的应用。更准确地说,通过识别
每个张量 $x \in \mathbb{R}^{F\times h\times w\times c}$ 作为序列 $(x_f)_{f=1}^{F}$ 和
$x_f \in \mathbb{R}^{h\times w\times c}$, 我们通过 x 获得潜在代码 $v_0^f := E(V_f)$,
对所有 f = 1, ..., F。前向扩散过程逐渐将高斯噪声 $\mathcal{N}(0, I)$ 添加到信号 x_0:

$$q(x_t|x_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I), \quad (1)$$

其中 $q(x_t|x_{t-1})$ 是给定 x_t 的条件密度
x_{t-1},以及 $\{\beta_t\}_{t=1}^{T}$ 是超参数。T 值较高
选择使得前向过程完全破坏
初始信号 x_0 得到 $x_T \approx \mathcal{N}(0, I)$。目标是
扩散模型就是学习一个向后的过程

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad (2)$$

对于 t = T, ..., 1(参见 DDPM [14]),这可以从标准高斯噪声 x_T 生成有效信号 x_0。
一旦从 x_T 获得 x_0,通过逐帧应用解码器获 生成的视频是ob-)
得:$V_f := D(x_0^f)$
对于所有 f = 1, . . . , F。然而,我们并没有学习一个预测器
等式2 中的均值和方差,我们学习一个模型 $\theta(x_t, t)$ 来
预测用于形成 x_t 的高斯噪声
输入信号 x_0 (常见的重新参数化[14])。

对于文本引导的视频生成,我们使用具有可学习权重 θ 的神经网络作为噪声预测器 $\theta(x_t, t, \tau)$
以文本提示 τ 为条件。我们对其进行训练
去噪任务:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,t,\tau,\epsilon\sim\mathcal{N}(0,I)}[\|\epsilon - \epsilon_\theta(x_t, t, \tau)\|^2], \quad (3)$$

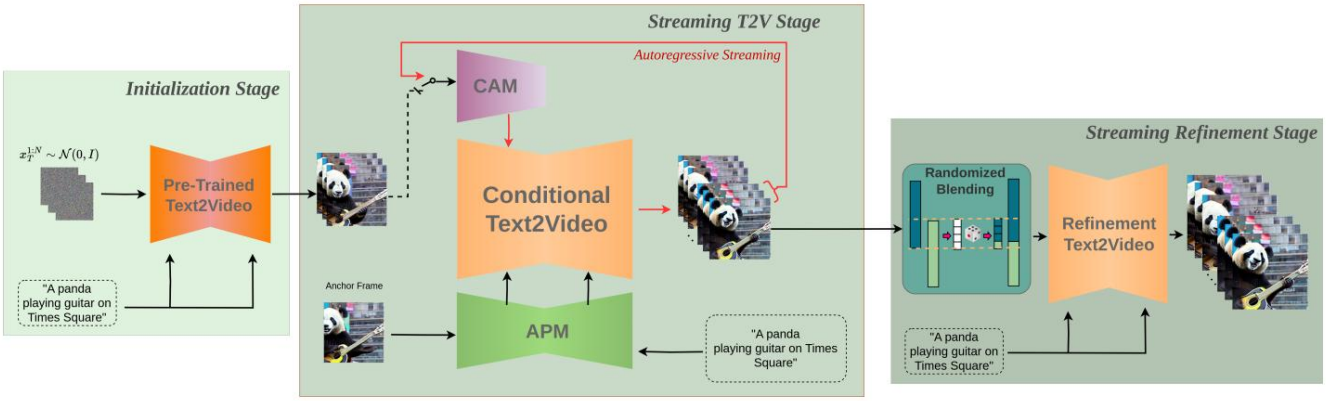


图 2. StreamingT2V 的整体流程涉及三个阶段：(i)初始化阶段:合成第一个 16 帧块
通过现成的文本转视频模型。(ii) 流式 T2V 阶段:后续帧的新内容以自回归方式生成。(iii)
流式细化阶段:使用具有随机混合方法的高分辨率文本到短视频模型对长视频（例如 240、1200 帧或更多)进行自回归增强。

使用数据分布pdata。为了简化符号,我们
将表示为 x r:s := (x j) xt的潜在序列
从帧 r 到帧 s,对于所有 r、t、s ∈ N。
文本转视频模型。文本转视频模型[4, 10, 15,
32、39]通常扩展预先训练的文本到图像模型
[27, 29]通过添加新的层来操作时间
轴。Modelscope（MS） [39]遵循这种方法,它扩展了类似 UNet 的[30]
稳定扩散架构
[29]具有时间卷积和注意层。它
在大规模设置中进行了训练,以生成 3
FPS@256x256 和 16 帧。

4.方法

在本节中,我们介绍了高分辨率
文本到长视频的生成。我们首先生成 256 ×
256分辨率长视频（240帧,或1200帧），
然后将其增强到更高的分辨率（720 × 720）。
图2给出了整个流程的概览。
长视频生成过程包括三个阶段：
初始化阶段,其中第一个 16 帧块由预先训练的文本到视频模型（例如
Mod-elscope [39])合成,流式 T2V 阶段,其中新内容

后续帧的自回归生成。为了确保块之间的无缝过渡,我们引入（参见

图3)我们的条件注意模块（CAM），它利用最后Fcond = 8 帧的短期信息
以及我们的外观保存模块（APM），它从锚帧中提取长期信息,以在自回归过程中保持物体外观和场景细节。在生成一段长视频（例如 240 帧）后，

1200 帧或更多），流式细化阶段
使用高分辨率文本到短视频模型（例如 MS-Vid2Vid-XL [47])自回归增强视频
采用我们的随机混合方法实现无缝块
处理。此步骤不需要额外的培训，
使我们的方法具有成本效益。

4.1. 条件注意模块

用于训练 Streaming T2V 中的条件网络
在阶段,我们利用预先训练的文本转视频模型（例如 Modelscope [39]）
的功能作为自回归长视频生成的先验。随后,我们

将这个预先训练的文本到（短)视频模型表示为
视频 LDM。对视频 LDM 进行自回归调节
结合前一个块的短期信息（参见
如图2 中所示,我们引入了条件注意力模块 (CAM)。CAM由 Video-LDM
UNet 中的特征提取器和特征注入器组成,其灵感来自 ControlNet
[45]。特征提取器使用逐帧图像编码器Econd,后面跟着与 ControlNet
相同的编码器层。

Video-LDM UNet 使用其中间层（初始化
与 UNet 的权重）。对于特征注入,我们让
UNet 中的每个长距离跳过连接都会通过交叉注意力关注 CAM 生成的
相应特征。
令 x 表示零卷积后Econd的输出。
我们使用加法将 x 与 CAM 第一个时间变换块的输出进行融合。为了注入 CAM 的
功能融入 Video-LDM Unet,我们考虑 UNet 的
跳过连接特征xSC ∈ R 宽×高×宽×高 （见图3）
批量大小为 b。我们应用时空组范数，
×c和xSC上的线性映射Pin 。令 x ′ ∈ R (b·w·h)×F 是
重塑后的结果张量。我们条件 x ′ 在
对应的CAM特征xCAM∈R (b·w·h)×Fcond×c
（见图3），其中Fcond是调节次数
通过时间多头注意力机制（T-MHA） [36]

即对每个空间位置（和批次)独立进行。使用可学习的线性映射PQ、 PK、PV进行查询、键、
和值,我们使用来自的键和值来应用 T-MHA
xCAM和来自 x 的查询 ′ 即 Q = PQ(x SC), K =

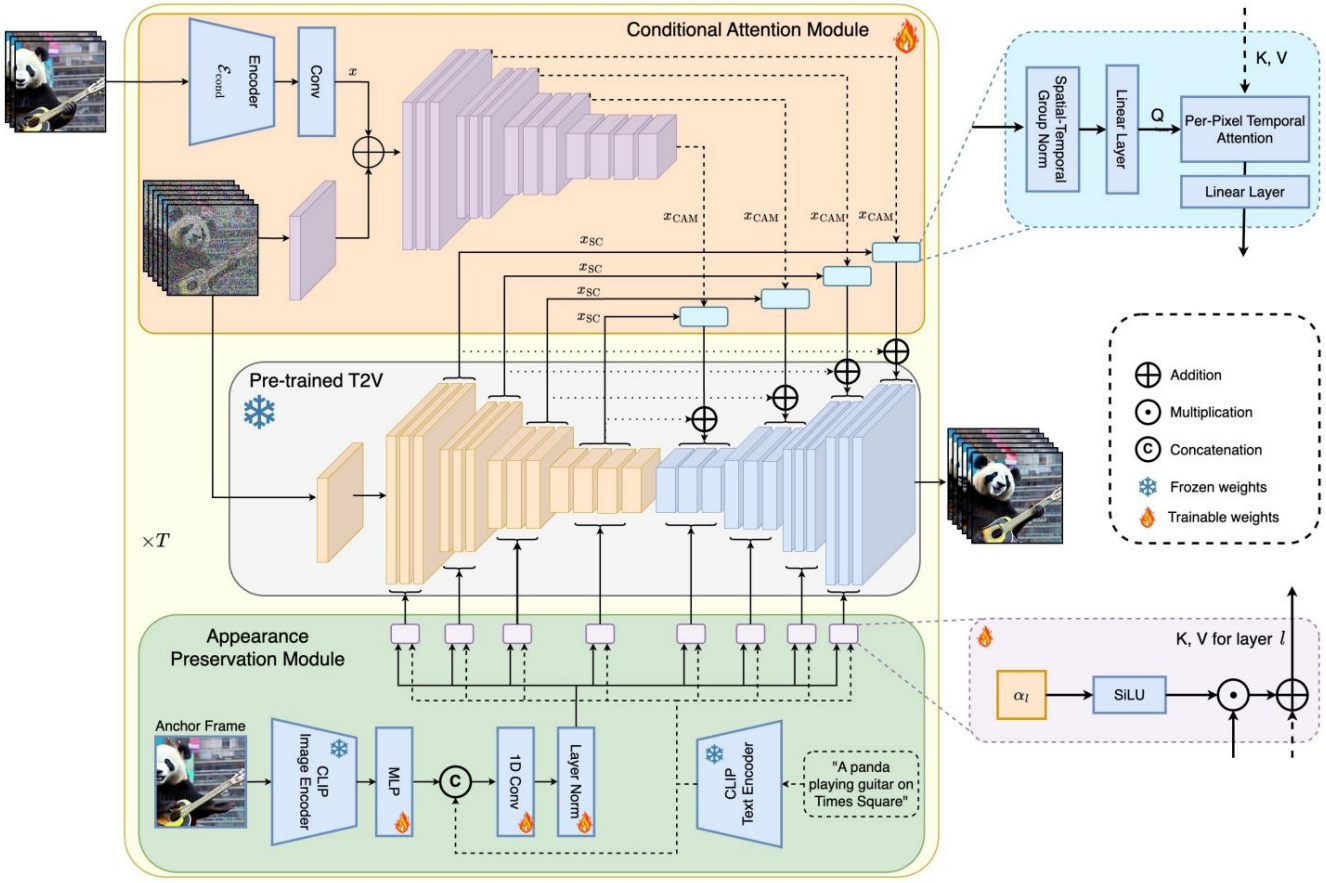


图 3. 方法概述:StreamingT2V 使用条件注意模块 (CAM) 增强视频扩散模型 (VDM),用于短期记忆,并使用外观保存模块 (APM) 增强长期记忆。CAM 使用帧编码器 (Econd) 来对前一个块上的 VDM 进行条件调整。CAM的注意机制可实现块与高运动之间的平滑过渡。APM 从锚帧中提取高级图像特征,并将其注入 VDM 的文本交叉注意模块中,从而在自回归期间保留对象/场景特征。

PK（xCAM）， V = PV （xCAM），

(4)

最后,我们利用线性映射Pout和重塑操作 R,将 CAM 的输出添加到跳过连接（类似于 ControlNet [45]）：

(5)

和 x ''' 用于 UNet 的解码器层。我们对Pout 进行零初始化,这样 CAM 最初不会影响基础模型的输出,从而提高了收敛性。

CAM 的设计使得能够根据前一个块的Fcond帧对基础模型的 F 帧进行条件处理。相比之下,稀疏编码器[11]采用卷积进行特征注入,因此需要额外的 F − Fcond 个零值帧（和一个掩码)作为输入,以便将输出添加到基础模型的 F 帧中。输入中的这些不一致性会导致输出中的严重不一致性（参见A.3.1节和 5.2 节）。

4.2. 外观保存模块

自回归视频生成器通常会遗忘初始物体和场景特征,导致严重的外观变化。为了解决这个问题,我们利用我们提出的外观保存模块 (APM),利用第一个视频块固定锚帧中包含的信息来整合长期记忆。这有助于在视频块生成过程中保持场景和物体特征（参见图A.8）。

为了使 APM 能够平衡锚框架和文本指令的指导,我们建议（见图3）：(i) 我们将锚框的CLIP [26]图像标记与文本指令中的 CLIP 文本标记结合起来,方法是使用 MLP 层将剪辑图像标记扩展为 k = 16 个标记,在标记维度上连接文本和图像编码,并利用投影块,得到xmixed ∈ R b×77×1024； (ii) 对于每个交叉注意层 l,我们引入一个权重αl ∈ R（初始化为 0)来

使用来自的键和值执行交叉注意
加权和xmixed,以及通常的 CLIP 文本编码
文本说明xtext：

(6)

A.3.2节中的实验表明,轻量级
APM模块有助于保留场景和身份特征
整个自回归过程（见图A.8）。

4.3. 自回归视频增强

为了进一步提高文本转视频结果的质量和分辨率,我们使用高分辨率
（1280×720)文本转（短)视频模型（Refiner Video-LDM,见图2）,例如

MS-Vid2Vid-XL [40, 47],自回归改进 24-
帧视频块。为此,我们为每个视频添加噪声
并使用 Refiner Video-LDM（SDEdit 方法[22])进行去噪。具体来说,我们将
每个低分辨率
将 24 帧视频块使用双线性插值[2] 转换为 720 × 720 分辨率,使用零填充转
换为 1280 × 720 分辨率,并使用
图像编码器 E 得到潜在代码x0,应用 T
前向扩散步骤（见公式1）,因此xT'仍然包含
信号信息,并用 Refiner Video-LDM 进行去噪。

天真地独立地增强每个块会导致
不一致的过渡（见图4 (a)）。为了克服这个问题
缺点,我们引入共享噪声和随机
混合技术。我们将低分辨率长视频分成
分成 m 个块V1, ..., Vm ,每个块包含 F = 24 帧,每帧
连续块之间有 O = 8 个帧重叠。对于
每个去噪步骤,我们必须对噪声进行采样（见公式2）。我们
将该噪声与已经采样的噪声相结合
前一个块的重叠帧形成共享
噪声。具体来说,对于块Vi ,i = 1,我们采样噪声

(7)

在扩散步骤 t（从 T 开始 ）,我们执行一个
使用 i进行噪声处理,并获取块Vi 的潜在代码
xt−1(i).尽管做出了这些努力,但过渡错位仍然存在（见图4 (b)）。

为了显著提高一致性,我们引入了随机混合。考虑潜在编码xL :=

xt−1(i − 1) 和xR := xt−1(i)两个连续块

Vi−1, Vi在去噪步骤 t−1 时。块的潜在代码xL
Vi−1从第一帧到
重叠帧,而潜在代码xR拥有
从重叠帧到后续帧的平滑过渡。因此,我们将两个潜在代码结合起来

通过在随机选择的重叠位置进行连接,

方法	↓MAWE	↓SCuts	↑CLIP
稀疏Ctrl [11]	6069.7	5.48	29.32
I2VGenXL [47]	2846.4	0.4	27.28
DynamiCrafterXL [42]	176.7	1.3	27.79
塞纳河[6]	718.9	0.28	30.13
SVD [3]	857.2	1.1	23.95
自由噪音[25]	1298.4	0	31.55
OpenSora [48]	1165.7	0.16	31.54
OpenSoraPlan [24]	72.9	0.24	29.34
StreamingT2V （我们的）	52.3	0.04	31.73

表 1. 与最先进的开源软件的定量比较
文本转长视频生成器。性能最佳的指标以红色突出显示,其次以蓝色突出显示。
我们的方法在
MAWE 和 CLIP 得分。仅在 SCuts 中,StreamingT2V 得分
第二好的,因为 FreeNoise 可以生成近乎恒定的视频。

通过从 {0, . . . , O} 中随机采样帧索引fthr
根据该公式,我们合并两个潜在向量xL和xR：

(8)

然后,我们更新整个长视频的潜在代码
xt−1在重叠帧上进行降噪,并执行下一步降噪步骤。因此,对于帧 f ∈ {1, ..., O}
重叠部分,块Vi−1的潜在代码与概率f一起使用
能力 1 − 。 这种潜在的概率混合
重叠区域有效地减少了不一致性
块之间（见图4（c））。在消融研究中,进一步评估了随机混合的重要性

附录（见 A.3 节）

5.实验

我们进行了定性和定量评估。附录中提供了实施细节和消融研究,以展示我们
贡献的重要性（第 3 节）。

A.3和 Sec. A.4)

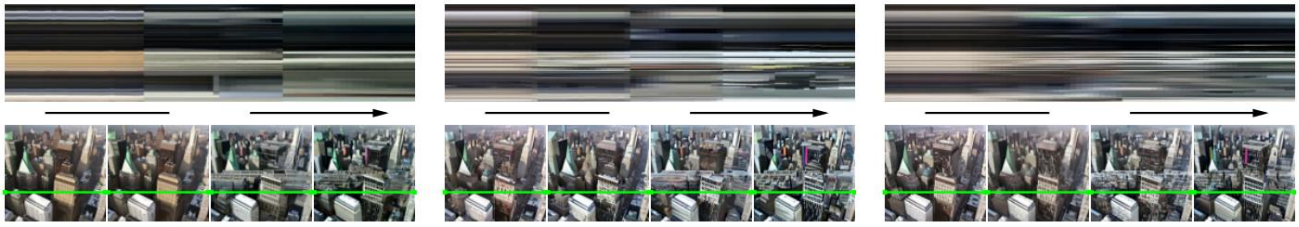
5.1. 指标

为了进行定量评估,我们测量时间一致性、文本对齐和每帧质量。

为了实现时间一致性,我们引入了 SCuts,它
使用以下方法计算视频中检测到的场景切换次数
AdaptiveDetector [1]使用默认参数。此外,我们提出了一种称为运动感知扭
曲的新指标
误差（MAWE）,它连贯地评估运动量
和扭曲误差,当视频同时表现出一致性和大量运动时,会产生较低的值

（具体定义见附录A.6 节）。对于涉及光流的度量,计算如下：

将所有视频调整为 720 × 720 分辨率。



(a)朴素连接 (b)共享噪音 (c)随机混合

图 4. 我们对视频增强器改进的消融研究。XT 切片可视化显示,随机混合可平滑的块转换,而两个基线都有清晰可见的块之间严重的不一致。

对于视频文本对齐,我们采用 CLIP [26] 文本图像相似度得分(CLIP)。CLIP 计算视频序列所有视频帧的 CLIP 文本编码与 CLIP 图像编码的余弦相似度。

所有指标首先按视频计算,然后对所有视频进行平均,所有视频均使用 240 定量分析框架。

5.2. 与基线的比较

基准。为了评估 StreamingT2V 的有效性,我们创建了一个由 50 个不同提示组成的测试集动作、物体和场景(列于 A.5 节)。我们与近期可用的方法进行了比较,包括:图像转视频方法 I2VGen-XL [47]、 SVD [3]、DynamiCrafter-XL [42]、OpenSoraPlan v1.2 [24]和 SEINE [6] ,视频转视频方法 SparseControl [11],

OpenSora v1.2 [48]和 FreeNoise [25]。

对于所有方法,我们使用它们发布的模型权重和超参数。对自回归生成方法的性能进行公平的比较和深入的分析,并使分析独立于

在所使用的初始帧生成器上,我们使用相同的 Video-LDM 模型生成第一个块,包含 16帧,给出一个文本提示并将其增强到720x720 使用相同的 Refiner Video-LDM 分辨率。然后,我们生成视频,同时我们通过对该块的最后几帧进行条件处理来启动所有自回归方法。对于在不同空间分辨率上工作的方法,我们应用对初始帧进行零填充。所有评估均基于 240 帧视频生成进行。

自动评估。我们对测试集显示,StreamingT2V 在无缝块转换和运动一致性方面表现最佳

(见表1)。我们的 MAWE 得分显著优于所有竞争方法(例如,比 OpenSoraPlan 的第二好成绩低了近 30%)。同样,我们的方法在所有竞争对手中,SCuts 得分第二低。只有 FreeNoise 得分略低,为满分。然而,FreeNoise 拍摄的视频几乎是静态的

(参见图5),这自然导致 SCuts 分数较低。OpenSoraPlan 经常出现场景切换,导致 SCuts 得分比我们的方法高 6 倍。SparseControl 遵循 ControlNet 方法,但可实现 100 倍以上的与 StreamingT2V 相比,场景切换次数有所增加。这显示了我们的注意力 CAM 块相对于 SparseControl 的优势。其中条件帧需要用零填充,因此输入的不一致导致严重的场景切换。

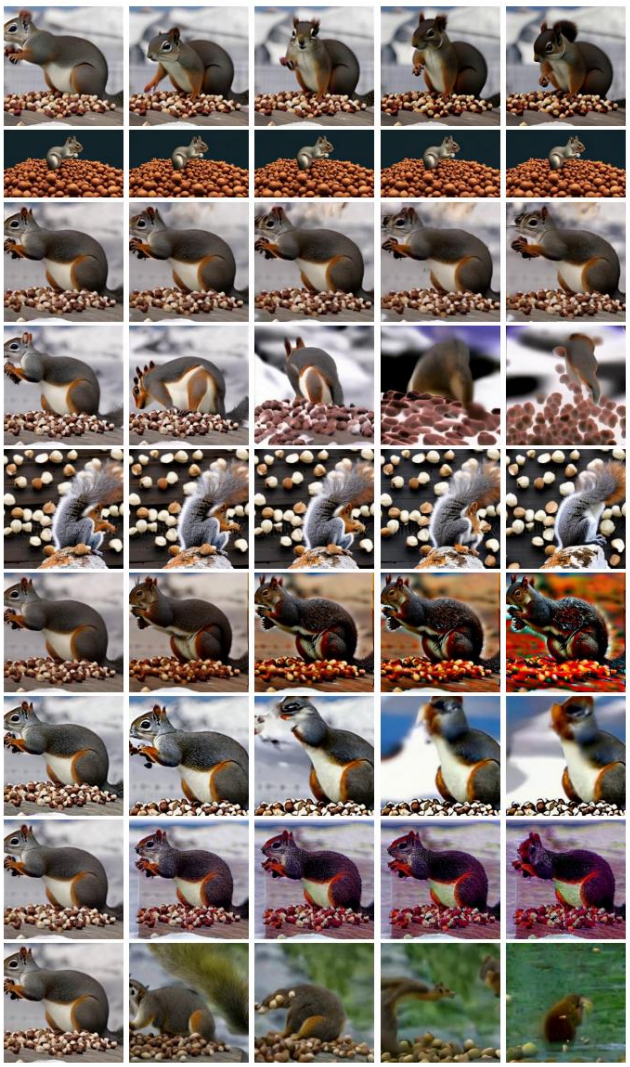
有趣的是,所有竞争方法都包含 CLIP 图像编码容易出现错位(以较低的 CLIP 分数衡量),即 SVD 和 DynamiCrafterXL 和 I2VGen-XL。我们假设这是由于域移位造成的;CLIP 图像编码器是在自然图像,但在自回归设置中,它应用于生成的图像。凭借我们的长期记忆,APM 提醒网络关于真实图像的领域,因为我们使用固定锚框架,使其不会降解,并保持与文本提示完美对齐。因此,StreamingT2V 获得所有评估方法中 CLIP 得分最高。

为了评估这些指标随时间的稳定性,我们以 20 帧为步长,从 120 帧到 220 帧对它们进行了计算。结果如下:MAWE 得分:(43.25、46.92、46.79、45.79、45.84、45.84)和 CLIP 得分:(32.45、32.30、32.16、32.02, 31.89, 31.79)。这些结果表明,随着时间的推移保持相对稳定。

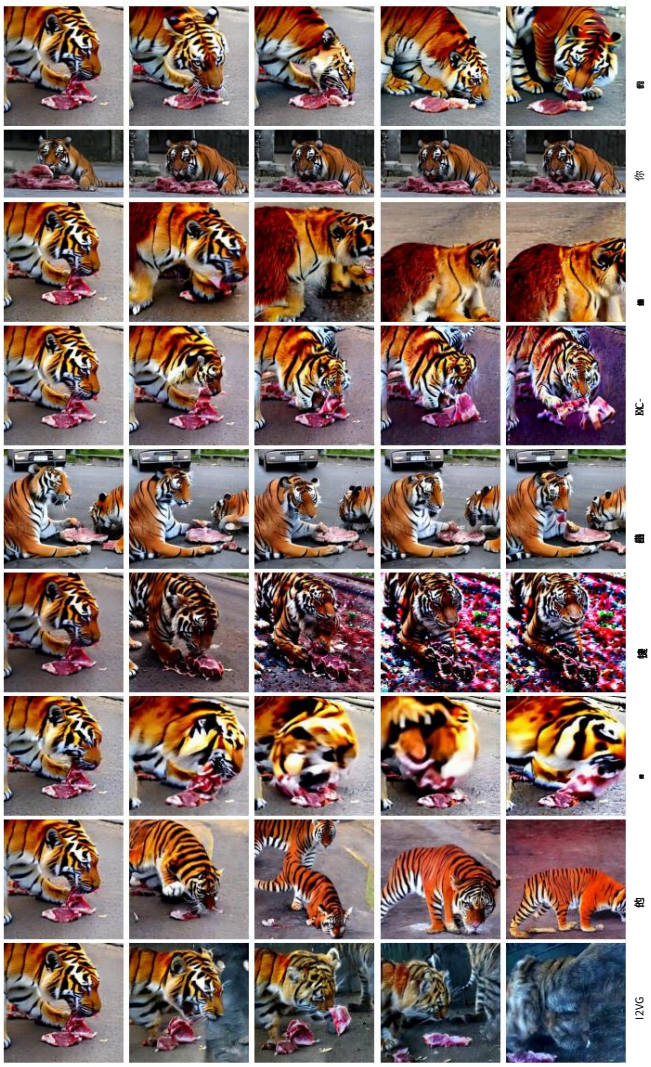
定性评估。最后,我们提出相应的图5 (以及A.2节)中的测试集上的视觉结果。为竞争对手描绘的帧高度相似表明所有竞争方法都存在视频停滞的问题,即背景和摄像头冻结,并且几乎没有产生物体运动。我们的方法是生成流畅一致的视频,而不会导致停滞。I2VG、SVD、SparseCtrl、SEINE、OpenSoraPlan 和 DynamiCrafter-XL 容易出现严重的质量下降,例如错误的颜色和扭曲的帧,以及不一致,表明它们通过 CLIP 图像进行调节

编码器和级联太弱,会严重放大误差。相比之下,得益于更强大的 CAM

机制,StreamingT2V 可以实现平滑的块转换。APM 条件依赖于固定的锚帧,因此 StreamingT2V 不会受到错误累积的影响。



（a)南极洲的一只松鼠在一堆榛子上。



(b) 一只老虎在街上吃生肉。

图 5. StreamingT2V 与最先进的方法在 240 帧长度的自回归生成视频上的视觉比较。在
与其他方法相比,StreamingT2V 可以生成长视频,而不会出现运动停滞的情况。

图 6.使用 OpenSora 作为基础模型的 StreamingT2V 结果。

6.结论和未来工作

在本文中,我们解决了生成长
视频来自文本提示。我们观察到所有现有的
方法制作的长视频要么存在时间不一致,要么严重停滞直至静止。为了克服

针对这些限制,我们提出了 StreamingT2V,它结合了短期和长期依赖关系,以确保平稳
连续高运动量的视频块
同时保留场景特征。我们提出了一种随机
混合方法允许使用视频增强器
自回归过程。实验表明,Stream-ingT2V 的表现优于竞争对手,能够生成长视频

文字提示,无内容停滞。

我们还注意到我们的方法可以推广到
DiT 架构也是如此,例如 OpenSora (OS) [48]，
我们添加了 CAM 模块,允许 OS 的最后 14 个转换块通过 CAM 的注意力机制
来关注前一个块的信息。APM 模块

与交叉注意力机制相关,就像在 StreamingT2V 中一样。将我们的框架添加到 OS 之后，
结果证实了该方法的泛化能力
（见图6)使得未来的研究可以集中对此方向进行详细的分析。