arXiv:2602.01340v1 [cs.CV] 1 Feb 2026

MTC-VAE：具有内容感知的多级时间压缩

Yubo Dong[1], **Linchao Zhu**[1], †[1]
ReLER、CCAI、浙江大学
†通讯作者

摘要

潜在视频扩散模型（*LVDM*）依赖变分自编码器（*VAE*）将视频压缩为紧凑的潜在表示。对于连续变分自编码器（*VAE*），更高的压缩率是理想的；然而，在不增加隐藏通道维度的情况下添加额外采样层，会显著降低效率。本文提出一种技术，可将固定压缩率的变分自编码器转换为支持多级时间压缩的模型，并提供一种简便且只需少量微调的方法，以抵消压缩率提高带来的性能下降。此外，我们研究了不同压缩级别对具有不同特征的视频片段的模型性能的影响，并通过实证证明了所提方法的有效性。我们还研究了将多级时间压缩变分自编码器与扩散模型 *DiT* 集成的方式，展示了在这些框架中成功进行并行训练的可能性及其兼容性。这项研究展现了多级时间压缩的潜在应用。

1. 引言

视频生成领域一直备受学术界和工业界关注，尤其是在 OpenAI 宣布 SORA 之后 [4]。目前，诸如 MagicTime [37],VideoComposer [31],AnimateDiff [10],Stable Video Diffusion（SVD） [3], HiGen [24], Latte [19], SORA [4],Open-Sora [41], Open-Sora-Plan [17], CogVideo [11],CogVideoX [34]和混元-DiT [18] 等潜在视频扩散模型（LVDM）凭借可靠性、性能和适应性，已在视频生成领域占据主导地位。这些 LVDM 的常见工作流程是使用变分自编码器（VAE） [15] 将原始视频压缩为潜变量。随后，训练去噪器来估计施加于这些压缩表示上的噪声。LVDM 使用的 Stable Diffusion 变分自编码器（SD-VAE） [23, 25], 专注于压缩单帧图像，却未考虑时间一致性，这会导致输入尺寸和硬件需求增加，同时降低重建的质量和准确性。先前的研究探索了在视频生成中使用向量量化变分自编码器（VQ-VAE） [29] 进行时间压缩的方法，即将视频转换为离散 token，供自回归模型使用 [9, 33, 35, 36]。

VideoVAE 是 LVDM 的关键组件。以 Stable Video Diffusion VAE（SVD-VAE） [3],OD-VAE [17], 和 CV-VAE [40], 为代表的既有方法，将时间维度的数据纳入编码，并优先通过预设的压缩率提升质量。这些方法会同时将视频片段编码为潜变量，以提高效率，并在之后将其合并用于扩散训练。在推理阶段，扩散变换器根据文本提示生成潜在表示视频。随后，它会将该潜在表示视频划分为多个片段，并使用 VAE解码器分别将其解码为视频。之后再将这些片段合并。由于视频动态各异，预设的时间压缩率可能并不适用于所有视频片段。

为探究在不影响重建质量的前提下提高时间压缩率的潜力，我们提出了**MTC-VAE**，即**多**级**时**间**压**缩变分自编码器，旨在与预训练VAE无缝集成。不同于时间压缩率固定的变分自编码器，**MTC-VAE** 会根据各视频片段的独特属性，将视频片段进行时间压缩并转换为潜在表示。它兼容多种3D变分自编码器作为基础模型，并相应地调整视频重建。基于这样的观点：稀疏或运动缓慢的片段可以比密集或变化迅速的片段采用更高的压缩率，我们提出了一种根据视频片段的时间复杂度动态分配压缩率的方法。然而，这种方法带来了一项技术难题：由于扩散变换器是基于由多个潜在片段组合而成的潜在视频进行训练的，如何才能将这些视频准确解码回其初始时间分辨率？我们设计了视频剪辑器模块，能够精准定位每个片段在隐藏特征中的初始帧（关键帧）

，从而能够按照各片段初始的压缩率准确解压。此外，我们提出了一种流引导损失，以便针对不同的视频动态，自适应地优化压缩效率与重建质量之间的平衡（表1）。

我们的MTC-VAE能否与预训练的DiT模型无缝结合？为探究这一问题，我们利用MTC-VAE对DiT进行微调。我们开展了实验，采用MTC-VAE编码器将视频转换为不同时间压缩率下的潜在表示。这些表示与相应的视频文本描述相关联，作为DiT模型的微调输入。我们调整了两个开源潜在视频扩散模型（LVDM），并观察到这些模型能够有效适应我们MTC-VAE波动的时间压缩特征。这种适配有助于在相同硬件条件下生成更长的视频，或降低相近时长视频的计算需求，从而优化质量与压缩效率之间的平衡（表2）。

我们的MTC-VAE实验表明，我们可以将压缩率显著提升92.4%，同时仍能获得几乎相同的性能，结构相似性仅相差0.0027，峰值信噪比仅相差0.15。我们以散点图呈现的详细分析比较了不同视频类型中质量指标与压缩率的关系，并揭示了一个有趣的见解：变分自编码器中较低的时间压缩率并不总能在所有视频风格上带来最佳结果。压缩率与质量之间的这种复杂关系支持了我们自适应策略的核心思想，说明对合适的视频片段选择性应用更高的压缩率，既能保持质量，又能降低整体压缩率。

2. 相关工作

2.1. 潜在视频扩散模型

LVDM 是人工智能领域的一项重大挑战 [30]。它们采用变分自编码器将视频内容封装为潜在表示，再利用去噪器进行噪声预测；这一领域正迅速发展。OpenAI 的 SORA [4], 能够生成时长一分钟、分辨率达 1080P 的视频，令世界惊叹。LVDM 根据去噪器架构分为两类。第一类采用基于U-Net的去噪器 [5, 26, 38], , 例如 MagicTime [37], 、AnimateDiff[10], 和 Stable Video Diffusion (SVD) [3]。第二类则采用基于 Transformer的去噪器[22], , 包括 Latte [19], 、SORA [4], 、CogVideoX [34],、Vidu [2], 和混元-DiT [18]。无论去噪器采用何种结构，变分自编码器都决定了去噪器的输入维度，以及潜在表示的视频重建保真度。因此，能够高效表征并确保高质量重建的变分自编码器，将显著提升性能。

2.2. 变分自编码器

变分自编码器最初旨在优化生成任务的证据下界（ELBO） [15], , 如今已成为各类生成模型的核心组件，可分为两类。第一类是向量量化变分自编码器 [29], , 它将视频转换为离散令牌，以供自回归模型使用 [9, 35, 36]。MAGVIT-v2 [36] 在视频重建方面表现出色，但其离散令牌并不适用于 LVDM。相较之下，连续VAE会创建连续视频表示，供 LVDM 使用；其中较有代表性的包括 SD-VAE [25] 和 SVD-VAE [3]。然而，这些方法仅关注空间压缩，忽略了时间冗余。近期研究开始尝试固定的时间压缩率。OPS-VAE [41] 采用两个变分自编码器分别进行空间压缩和时间压缩，CV-VAE[40]则使用时间压缩来与 SD-VAE 对齐，而 OD-VAE [6] 采用三维因果卷积神经网络实现联合压缩。我们在实验中将 MTC-VAE 与这些方法进行了比较。

3. 方法

在本文中，我们提出了**MTC-VAE**（§3.1），即**M**多级**T**时间**C**压缩变分自编码器，旨在与现有预训练VAE无缝集成。我们描述了一种核心压缩算法（§3.2），其参数按启发式方式设定；我们的分析表明，变分自编码器的压缩质量会受到视频片段属性的影响。随后，我们讨论编码器流程（§3.3）。为将片段恢复到其原始时间分辨率，我们开发了视频剪辑器（§3.4），并测试了多种配置，同时考虑到扩散Transformer是在由组合片段构建的潜在视频上训练的。此外，我们在本节详细阐述了微调策略（§3.5）以及采用所提方法所必需的关键结构改动。

3.1. MTC-VAE

我们的 MTC-VAE 由一个编码器 $\mathcal{E}$ 和一个解码器组成 $\mathcal{D}$。如图1所示，视频首先被分割成固定长度的视频片段，分辨率为 T×H×W。多级压缩算法通过平衡质量与压缩来确定时间压缩率 C 。随后，编码器将视频片段转换为尺寸为 t×h×w 的潜在片段，并根据压缩率确定时间采样层的数量 C。每个片段都会额外添加一个嵌入，以指示片段区域。在解码过程中，使用线性视频预测器 $\mathcal{P}$ 来识别带标签特征，并将潜在表示划分为多个片段。我们使用

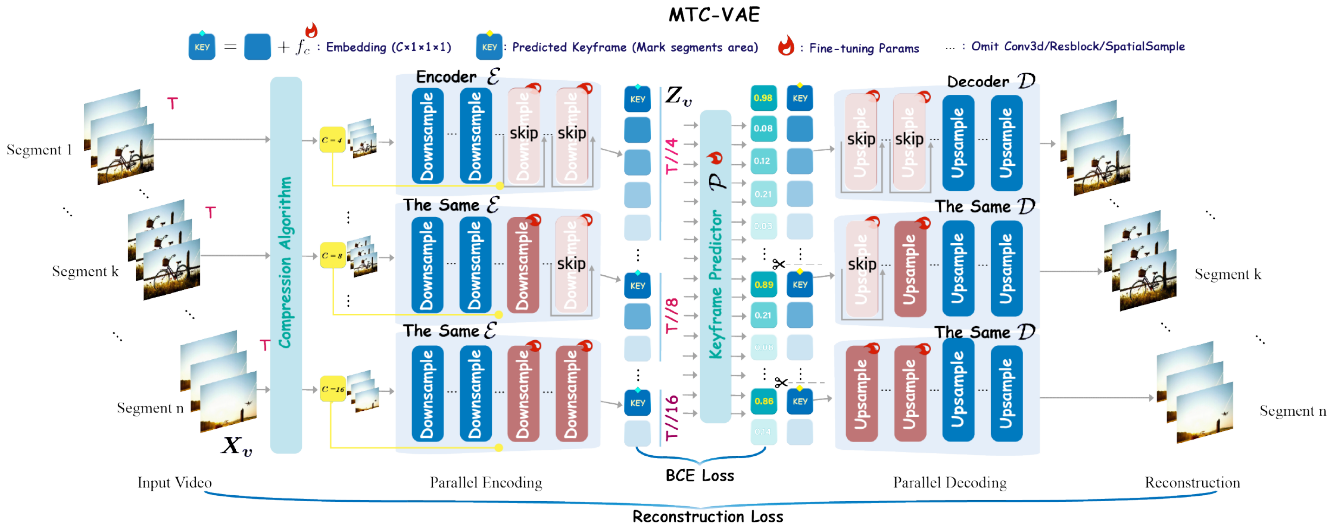


图1. 我们的 MTC-VAE 模型如图所示。该架构采用一种压缩算法，自适应地管理 **视频片段的时间压缩率** C，并决定采用的采样层数量。图中仅展示了时间采样层，新引入的采样层由现有的预训练层初始化。片段区域由嵌入 f_c表示。我们的线性关键帧预测器擅长从组合后的潜在表示中识别带标签的潜在特征，并据此划分片段。随后，这些潜在片段将被解码，以生成最终视频输出。具体过程可参见算法 21中的伪代码。

潜在表示长度和固定视频长度来计算时间压缩率并将其解码还原。诸如4×8×8 这样的比率意味着 $\frac{T}{t} = 4$, $\frac{H}{h} = 8$, $\frac{W}{w} = 8$等等。我们在原有架构中集成了新的上/下采样层（采用最近邻采样层和基础视频VAE的权重），以实现 4×8×8、 8×8×8和 16×8×8的压缩率。

3.2. 压缩算法

为了提升时间压缩效果，同时保持相当的重建效果，我们的算法所要应对的主要挑战是在重建**质量** 与**压缩**效率之间找到最佳平衡。为此，我们针对输入视频优化评分函数。考虑将一个视频划分为 N 个片段，并用张量 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{N\times T\times H\times W\times 3}$ 表示，其中每个片段 $\boldsymbol{x}_i$ 包含 T 帧。我们的目标是最大化总得分:

$$\sum_{i=1}^{N} \Big[(1-\alpha(x_i))\cdot Q(c_i, x_i) + \alpha(x_i)\cdot \log_2(c_i)\cdot w\Big], \quad (1)$$

$$\text{s.t. } c_i \in \{4, 8, 16\}, \forall i \in \{1, 2, ..., N\}. \quad (2)$$

该求和由两个部分组成：前一个片段评估每个部分在指定压缩级别下的**质量**，后一个片段则鼓励提高**压缩**率，其中 w 用于控制压缩程度增加的倾向，默认值为 $w = 1.5$。

质量评估会为每个片段生成一个质量向量 $\boldsymbol{Q}(x_i) = [Q(4, x_i)\ Q(8, x_i)\ Q(16, x_i)]$。给定片段的压缩容差因子 $\alpha(x_i)$计算如下:

$$\alpha(x_i) = \frac{1}{2}\cdot \frac{\overline{Q}(x_i) - Q_{\min}}{Q_{\max} - Q_{\min}} + \frac{1}{2}\cdot\left(1 - \frac{\sigma_Q(x_i)}{\sigma_{Q_{\max}}}\right). \quad (3)$$

其中: $\overline{Q}(x_i)$ 表示片段 x_i在不同压缩率下的平均质量 $Q_{\min}$，而 $Q_{\max}$ 表示所有压缩率和片段中的最低和最高峰值信噪比， $\sigma_Q(x_i)$ 表示片段x_i在不同压缩率下的质量标准差， $\sigma_{Q_{\max}}$ 则是所有片段中的最大标准差。表 $Q()$ 证明，使用峰值信噪比评估5更为有效。该模型会在以下情况下提高$\alpha(x_i)$（促进更高压缩率）：1. 片段在不同压缩率下的平均质量更高。2. 片段在不同压缩率下的质量变化最小。

3.3. VAE编码器

通过三个步骤对视频进行编码：首先，使用我们的算法确定每个视频片段适用的压缩率。接着，并行压缩每个片段，并为每个片段加入一个关键帧嵌入。最后，沿时间维度将它们拼接起来。

编码过程。考虑视频片段 $\boldsymbol{X}$。压缩潜在表示表示为 $\boldsymbol{Z} \in$

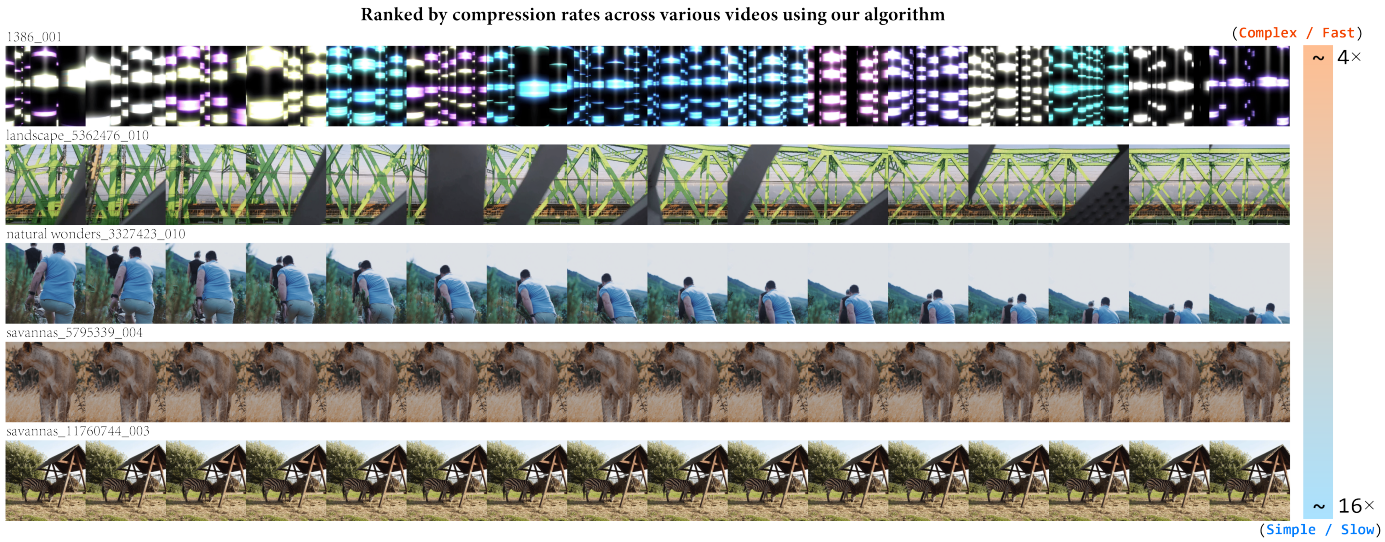


图 2. 使用我们的压缩算法，可以清楚地观察到，该算法倾向于对内容较简单、运动较慢的视频采用较高的压缩率，而对内容较复杂或运动较快的视频采用较低的时间压缩率。

$\mathcal{R}^{N\times t\times h\times w\times c}$。最初，MTC-VAE采用我们的压缩算法 3.2 来确定一系列压缩率 $\boldsymbol{C}$，定义为 $\boldsymbol{C} = [c_1, c_2, \ldots, c_N] \in \mathcal{R}^N$，其中第 i个视频片段的压缩率由c_i表示。随后，视频片段 $\boldsymbol{X}$ 由 $\mathcal{E}$并行处理，并可能根据 $\boldsymbol{C}$ 中的压缩率省略采样层，以得到潜在特征：

$$\boldsymbol{z}_i = \mathcal{E}(\boldsymbol{x}_i, c_i), \quad i \in \{1, \ldots, N\}. \tag{4}$$

对于每个片段的第一个潜在特征 $\boldsymbol{z_i}$，位于 $t = 0$，称为**关键帧**，其中会加入额外嵌入 f_c，参见图1。关于是否在最后一帧整合 f_c，或将其排除，详见表4）：

$$\boldsymbol{z_i}[:, j] = \begin{cases} \boldsymbol{z_i}[:, j] + f_c, & \text{if } j = 0 \\ \boldsymbol{z_i}[:, j], & \text{otherwise} \end{cases} \tag{5}$$

最终的潜在表示 $\boldsymbol{Z}$ 由拼接各个片段编码而成：

$$\boldsymbol{Z} = [\boldsymbol{z}_1 \| \boldsymbol{z}_2 \| \ldots \| \boldsymbol{z}_N], \quad \boldsymbol{Z} \in \mathcal{R}^{Nt\times h\times w\times c}. \tag{6}$$

3.4. VAE解码器

鉴于各片段采用不同的时间压缩率进行压缩，首要任务是获取完整潜在特征的具体时间压缩率。确定潜在特征的时间压缩率后，即可根据各片段对应的时间压缩率并行解码。**恢复时间压缩率**。从潜在特征中提取时间压缩率颇具挑战。我们提出两种结构来解决这一问题：a)**逐潜在单元分析**。我们使用预测器评估潜在片段中每个时间索引处的时间压缩率，并利用各位置的平滑压缩率结果来近似实际片段压缩率。然而，这种方法需要训练更多参数，可能会显著降低重建能力。b)**将多分类转为二元分类**。针对前一种预测器的复杂性，我们设计了一种新型关键帧预测器，其唯一功能是识别关键帧（每个潜在片段的初始帧）。关键帧预测器 $\mathcal{P}$充当二分类器，随时间遍历 $\boldsymbol{Z}$，生成对数几率以检测潜在关键帧潜变量，用于ˆ根据固定长度片段 T计算 $\boldsymbol{C}$。随后， $\boldsymbol{Z}$在时间维度上按这些已识别的关键帧潜变量进行划分，可表示为：

$$\hat{\boldsymbol{C}} = \left\{ \frac{T}{\text{length}(\boldsymbol{Z}[i])} \mid i \in N \right\}. \tag{7}$$

解码过程。根据预测器确定的关键帧，将潜变量划分为不同的潜在片段。每个片段 z_i 随后被解码为视频部分 ˆx，并进行合并：

$$\hat{x}_i = \mathcal{D}(z_i, \hat{c}_i), \quad i \in \{1, \ldots, N\}. \tag{8}$$

3.5. 训练

我们采用了两步微调方法。首先，我们快速微调了压缩率为 8×8×8和 16×8×8的变分自编码器。通过添加两个采样层，并以预训练VAE中相邻的采样层为基础进行初始化

模型	模式	Ch	WebVid-10M VCPR ↑	峰值信噪比 ↑	结构相似性 ↑	学习感知图像块相似度 ↓	Panda-70M VCPR ↑	峰值信噪比 ↑	结构相似性 ↑	学习感知图像块相似度 ↓
VQGAN [8]	(1×8×8)	4	64	26.26	0.7699	0.0906	64	26.07	0.8295	0.0722
SD-VAE [25]	(1×8×8)	4	64	30.19	0.8379	0.0568	64	30.40	0.8894	0.0396
SVD-VAE [3]	(1×8×8)	4	64	31.15	0.8686	0.0547	64	31.00	0.9058	0.0379
TATS [9]	(4×8×8)	4	256	23.10	0.6758	0.2645	256	21.77	0.6680	0.2858
CV-VAE [40]	(4×8×8)	4	256	30.76	0.8566	0.0803	256	29.57	0.8795	0.0673
OPS-VAE [41]	(4×8×8)	4	256	31.12	0.8569	0.1003	256	31.06	0.8969	0.0666
OD-VAE [17]	(4×8×8)	4	256	31.16	0.8694	0.0586	256	30.49	0.8970	0.0454
CogVideoVAE [34]	(4×8×8)	16	256	**35.25**	**0.9323**	0.0317	256	**34.92**	**0.9422**	0.0206
MTC-VAE	(/×8×8)	4	443.7↑ +73.3%	31.25	0.8703	0.0621	371.9↑ +45.3%	30.69	0.8993	0.0464
MTC-VAE	(/×8×8)	16	**492.5**↑ +92.4%	35.10	0.9296	**0.0304**	**431.2**↑ +68.4%	34.84	0.9409	**0.0205**

表1. VAEs 在 WebVid-10M 和 Panda-70M 验证集上的视频重建结果，包括压缩率（VCPR）。在更高的时间压缩率下，我们的 MTC-VAE 仍能达到与 CogVideoVAE 模型相当的性能。“Ch”表示潜在特征通道大小。/ 表示多级时间压缩。最佳和次佳分数分别以**粗体** 和下划线标出。

并使用其参数，我们调整了这些层，使其适应新的功能。第一阶段的目标是结合重建损失、对抗损失和 KL 正则化损失，使用与 OD-VAE 相同的损失 [6]。我们使用 Adam 优化器 [14]，在第一阶段以 1×10^{-5}的学习率进行 10k 步微调，并使用 4 个 A100 GPU。VAE 训练数据集包含来自 OpenSo-raPlan 的开源视频 [17]。输入视频被划分为 17 帧片段，分辨率为 256×256 像素。

在第二阶段，我们训练了嵌入 f_c(c×1×1×1) 和线性关键帧预测器，并采用额外损失：**二元交叉熵损失**。关键帧检测器 $\mathcal{P}(\cdot)$ 用于判断潜在帧 Z_i 是否为关键帧，其依据是该损失：

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right], \quad (9)$$

其中，$p_i = \mathcal{P}(Z_i)$ 表示预测的关键帧概率，y_i 表示真实标签。**光流引导的一致性损失**。我们发现，同时提升运动和质量两个方面能够改善性能。光流引导的一致性损失为：

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{quality}} + \mathcal{L}_{\text{motion}}, \quad (10)$$

质量损失和运动损失定义如下：

$$\mathcal{L}_{\text{quality}} = \|\text{warp}(\hat{\boldsymbol{X}}, \Delta\mathbf{f}) - \boldsymbol{X}\|_1, \quad (11)$$

$$\mathcal{L}_{\text{motion}} = \|\Delta\mathbf{f}\|_1. \quad (12)$$

这里，$\Delta\mathbf{f}$ 是来自光流模型 ϒ[20], 的光流，而 warp($\boldsymbol{X}$, Δf) 是使用该光流调整后的输出。此阶段额外进行了 10k 步训练。为增强稳定性，我们按照 [27, 28]提出的方法，在训练期间采用指数移动平均（EMA），衰减率为 0.999。

4. 实验

首先，我们详细介绍实验配置和评估指标（§4.1）。随后，我们开展 MTC-VAE 与各种基线模型在视频重建方面的全面比较。我们探讨视频特征与压缩容忍率之间的相关性（§4.2）。此外，我们还分析定量重建结果和生成结果（§4.3）。另外，我们开展了一系列消融研究（§4.4）。

4.1. 实验设置

基线方法。为评估性能和效率，我们将 MTC-VAE 与八种用于视频重建的先进变分自编码器进行了比较。这些模型包括：(1) VQGAN、SD-VAE 和 SVD-VAE：它们是基于向量量化技术的图像变分自编码器。VQGAN [8] 是一种广受欢迎的图像向量量化变分自编码器（f8-8192 版本） [40]。SD-VAE [25] 广泛应用于 LVDM，采用稳定版 SD2.1-VAE[40]。SVD-VAE [3]通过改进解码器增强了视频变分自编码器，并与 SD-VAE 共用编码器。(2) TATS、CV-VAE 和 OPS-VAE：它们是专为视频生成设计的视频变分自编码器。TATS [9]是一种用于自回归视频生成的三维视频向量量化变分自编码器。CV-VAE [40] 和 OPS-VAE[41] 与我们的研究同期开发。(3) OD-VAE 和 OpenSora-Plan：OD-VAE [6] 是一种同期开发的视频变分自编码器，支持空间和时间压缩。OpenSora-Plan [17] 是一个使用 OD-VAE 训练的 LVDM 模型，采用 Open-Sora Plan v1.2.0 的 93×480p 模型进行比较。(4) CogVideoVAE 和 CogVideoX-5B：CogVideoVAE [34] 是来自 CogVideoX 的三维因果变分自编码器，采用空间和时间下采样。CogVideoX-5B [34]支持 720p 视频生成。此外，我们使用 MTC-VAE编码器对两个 DiT 模型 CogVideoX 和 OpenSora-Plan 进行微调，将潜变量和文本对编码为输入，并评估了它们在推理阶段的性能。

评估指标与数据集。在我们的研究中，我们采用了两个广受认可的大型开放域视频数据集：WebVid-10M [1] 和 Panda-70M [7]。验证集中的每个视频都被标准化为包含 34 帧的视频片段，分辨率为 256×256。我们在两个版本的 MTC-VAE 上进行了微调：

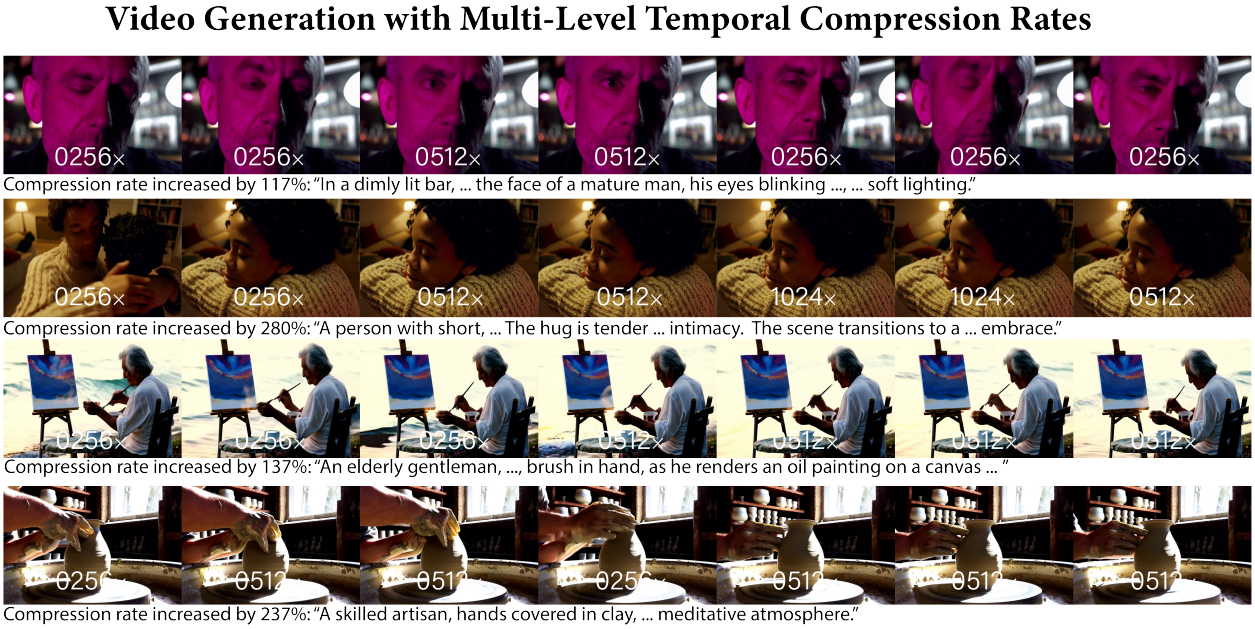


图 3. MTC-VAE 使经过精细调优的 DiT 能够为**简单**数据分配更少的**潜在空间，以实现**更高的压缩率，同时为**复杂**帧分配更多空间，采用**更低的**压缩率。我们展示了四个采用不同压缩率生成的视频： 256× (4×8×8)、 512× (8×8×8) 和 1024× (16×8×8)。该方法有助于 DiT 生成帧数可变的视频。生成过程包括以 50 个采样步数生成 720p 视频，并使用精心设计的提示词来突出时间场景转换。更多视觉内容和比较结果请参见补充材料第5 节。

模型	Ch	VCPR ↑ (倍)	延迟 ↓ (秒/帧)	加速比 ↑ (倍)	VBench ↑ 总计 (%)	分辨率	步数
OpenSora-Plan [17]	4	256.0	1.36	1.00×	74.34	512×512	30
MTC-OSP	4	**441.9**	**0.71**	**1.91**×	74.72	512×512	30
CogVideoX-5B [34]	16	256.0	4.43	1.00×	80.88	720×480	50
MTC-CogVideo	16	366.7	3.12	1.42×	80.70	720×480	50

表 2。视频生成评估结果。本表比较了 CogVideoX 和 OpenSora-Plan 在使用和未使用我们的 MTC-VAE 时的 VCPR、加速比、延迟和采样步数。"通道"表示潜在特征通道大小。

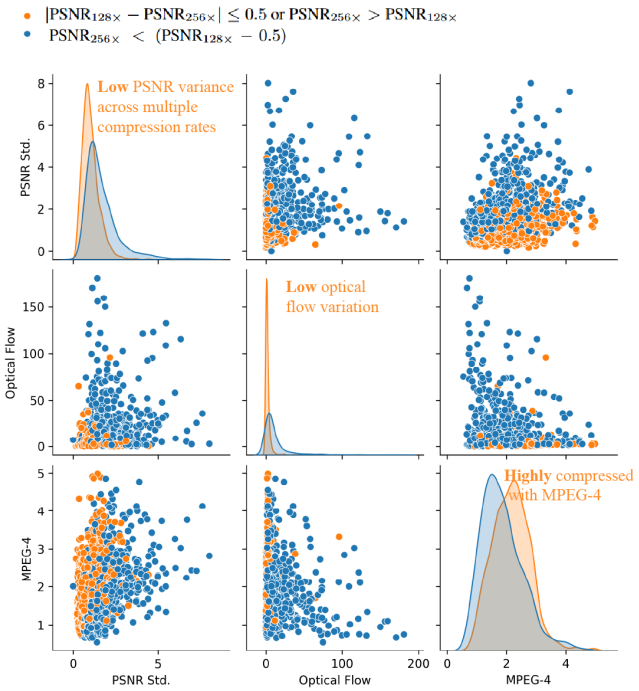
MTC-VAE(ch=4)，由 OD-VAE [6], 衍生而来；以及 MTC-VAE(ch=16)，基于 CogVideoX-5B [34]的变分自编码器进行改造。

指标。对于视频重建，我们使用视频压缩率（$VCPR = \frac{T}{t} \cdot \frac{H}{h} \cdot \frac{W}{w}$）衡量压缩效率和网络复杂度，并采用三项质量指标：峰值信噪比（PSNR）[12], 结构相似性指数（SSIM） [32] 和学习感知图像块相似度（LPIPS） [39]。关键帧预测准确率（准确率）通过正确预测的帧数除以帧总数计算得出。对于视频生成，我们使用 VBench [13], VCPR、加速比和延迟评估性能。

DiT微调。 对于视频生成评估（如伪代码所示 24），我们将我们的方法与两个领先模型 CogVideoX-5B [34] 和 OpenSora-Plan [17]。我们基于这些架构对模型的两个版本进行微调。我们的主要目标是最小化以下定义的噪声预测损失：

$$\mathcal{L}_{\text{DiT}} = \mathcal{L}_{\epsilon}(\boldsymbol{Z}; \boldsymbol{Z}_{\boldsymbol{prompt}}). \tag{13}$$

使用 MTC-VAE编码器对 DiT 进行微调，共训练 20,000 步。训练方案采用 1×10^{-5} 的学习率和 4 的批量大小。我们使用 OpenSoraPlan数据集，并将分辨率保持在 256 ×256 像素，与 [17] 中所述的 OpenSoraPlan 模型保持一致。我们的模型基于 $720p$ 版本的 CogVideoX 模型 [34]，以 $720p$ 的分辨率开发。此过程在 PyTorch 框架 [21] 中使用 4 块 NVIDIA A100 GPU 执行，每块显存为 80GB。



A VideoVAE with a 256× compression rate (4×8×8) compared to a 128× compression rate (2×8×8).

图 4. 散点图展示了 256×(4×8×8) 的压缩在何种情况下能够达到与 128×(2×8×8) 相当或更高的质量。橙色表示某个视频片段中 $|$峰值信噪比$_{128\times}$ − 峰值信噪比$_{256\times}| \leq 0.5$ 或峰值信噪比$_{256\times}$ > 峰值信噪比$_{128\times}$ 的情况，而蓝色表示峰值信噪比$_{256\times}$ < (峰值信噪比$_{128\times}$ − 0.5)。峰值信噪比标准差表示压缩级别 128×(2×8×8)、256×(4×8×8)、512×(8×8×8) 和 1024×(16×8×8) 下峰值信噪比数值的标准差。光流指运动动态，MPEG-4 指通过 MPEG技术对视频进行的存储压缩级别。这些数据来自 Pixabay 合集中的 1,024 个视频，收录于 Open-SoraPlan 开源数据集中 [17]。

4.2. 讨论

哪些视频片段可以进行更高程度的时间压缩? 图4展示了视频能够取得相近性能的情况（$|$峰值信噪比$_{128\times}$ − 峰值信噪比$_{256\times}| \leq 0.5$），以橙色点表示。橙色点和蓝色点的分布显示两者存在一定重叠，但仍可明显区分。通过分析图中主对角线上的平均差异，可以看出：1）在 MPEG-4 [16] 下具有较高可压缩性的视频片段，通常即使经过大幅时间压缩，也能保持画质。2）光流较低、运动平滑的视频序列，往往能够承受更高的压缩级别。3）不同压缩率下的峰值信噪比相近，表明可以进行更高程度的压缩，而不会造成显著的质量损失。

4.3. 主要结果

视频重建比较。在表1中，我们报告了两个模型在两个数据集上的四项评估指标结果。主要观察如下：1）SVD-VAE 的视频压缩表现优于 SD-VAE。在 WebViD 上，其峰值信噪比达到 31.15，而 SD-VAE 为 30.19。2）CV-VAE 通过监督潜在空间提升时间一致性，将学习感知图像块相似度降至 0.0803。3）OPS-VAE 和 OD-VAE 借助因果编码器-解码器设计及有效的结构设计，在更高压缩率下取得了更好的性能。在 Panda-70M 上，当 VCPR 为 256 时，两者的峰值信噪比分别达到 31.06 和 30.49，结构相似性分别达到更高的 0.8969 和 0.8970。4）CogVideoVAE 采用更多通道，从而增强潜在表示能力。5）与上述模型相比，我们的 MTC-VAE 方法以极小的性能损失实现了更高的视频压缩率。在 WebVid-10M 验证集和 Panda-70M 上，MTC-VAE(ch=4) 的压缩率较 OD-VAE 和 CogVideoVAE 等常用模型分别提升 73.3% 和 45.3%。MTC-VAE(ch=16) 在默认 w 设置下的提升更为显著。此外，在压缩率更高的情况下，我们的 MTC-VAE 变体仍能与 OD-VAE 和 CogVideoVAE 相媲美。在 Panda-70M 验证集上，MTC-VAE(ch=4) 的峰值信噪比和结构相似性均高于 CV-VAE 和 OD-VAE。同样，在 WebVid-10M 验证集上，我们的 MTC-VAE(ch=16) 将压缩率提升了 92.4%，同时保持了与 CogVideoVAE 模型相当的性能。

视频生成对比。在表2中，我们展示了微调后的 DiT 与我们的 MTC-VAE 及其他基线模型相结合进行视频生成的结果。表2 中的结果呈现了我们对 VBench 视频生成基准的全面评估。对于 OpenSora-Plan，MTC-OSP 模型将视频压缩率（VCPR）从 256 提升至 441.9，生成速度提高了 1.91 倍，同时 VBench 分数也有所提升。对于 CogVideoX-5B，尽管 MTC-CogVideo 模型的 VBench 分数略有下降（从 80.88 降至 80.70），但它将 VCPR 从 256 提升至 366.7，并将延迟从每帧 4.43 秒缩短至 3.12 秒。因此，我们得出以下结论：1）我们的 MCT-VAE 方法兼容当前用于视频生成的 DiT 模型。2）LVDM 可通过多级时间压缩率加快视频生成过程。

视觉结果。 在分析和定性评估中，我们采用了与 CogVideoX 相同的提示词，以及 VBench 的扩展提示词，如图3所示。我们的研究结果表明，使用 MTC-VAE 进行微调后，DiT 在视觉美感方面的图像质量与之相当。此外，采用我们的 MTC-VAE 的 DiT 在

特定时刻可实现更高的压缩率，例如角色在场景中执行细微动作时（如第 3 行和第 4 行所示），或开始沉思、眺望远方时（如第 1 行和第 2 行所示），等等，从而使我们的方法能够生成更长的视频。更多视频结果请参阅补充材料。

模型	$\mathcal{L}_{\text{flow}}$	w	VCPR ↑	峰值信噪比 ↑	学习感知图像块相似度 ↓	准确率(%) ↑
OD-VAE	–	–	256.0	30.69	0.0502	–
OD-VAE	–	–	512.0	29.99	0.0565	–
MTC-VAE	✗	1.0	348.0	30.62	0.0564	99.8
MTC-VAE	✓	1.0	370.0	30.76	0.0540	99.9
MTC-VAE	✓	1.5	454.5	30.72	0.0569	99.9
MTC-VAE	✓	2.0	748.0	30.32	0.0584	99.8
MTC-VAE	✓	2.5	827.0	30.14	0.0601	99.8

表 3. 对 MTC-VAE 去除光流引导的一致性损失后的消融研究，并与压缩率为 256 和 512 时的 OD-VAE 进行比较。我们发现，光流引导的一致性损失有助于保持变分自编码器的性能。

4.4. 消融研究

使用 w 来控制时间压缩率。为了更好地了解 w 对时间压缩算法的影响，我们使用 OpenSoraPlan数据集中的视频，通过调整 w，将其从 1.0 按 0.5 的增量调整至 2.5，从而比较不同压缩率下的模型，如表3所示。我们的研究结果表明：1）将 w 从 1.0 提高到 2.5 会提升压缩率。值得注意的是，MTC-VAE 的 VCPR 最高可达 827，峰值信噪比为 30.14，超过了 OD-VAE 在 512 倍压缩率下的峰值信噪比。2）加入光流损失 L_{flow} 可提升变分自编码器的性能；使用 $w = 1$时，在第二阶段训练中使用 L_{flow} 可提高峰值信噪比，使其达到 30.76。

方法	Idx	f_c	峰值信噪比↑	学习感知图像块相似度↓	准确率↑
MTC-VAE	开始	✗	24.31	0.2390	0.41
MTC-VAE	end	✓	32.49	0.0353	1.00
MTC-VAE	开始	✓	**32.90**	**0.0338**	1.00

表4. 关于片段标记额外嵌入的消融研究。将嵌入添加到第一个索引可实现最佳性能，而移除该嵌入会显著降低准确率。

额外嵌入 f_c。可学习参数 f_c (C×1×1×1)会添加到潜在片段中，以标记每个片段区域，让关键帧预测器更容易划分片段。为分析这种嵌入的有效性，并确定每个片段的第一个位置是否更好，我们设置了两组对照：1）第一组不使用 f_c 嵌入，但仍将每个片段的第一帧视为关键帧（预测器使用关键帧对潜变量进行分段）；2）将嵌入添加到每个片段的最后一帧，并将最后一帧视为关键帧。我们随机选取了 256 个分辨率为 256、每个包含 34 帧的视频进行这项实验。消融实验（表 4）表明：1. 不使用额外嵌入时，准确率显著下降。2. 将嵌入添加到片段开头优于添加到片段末尾，这可能是因为因果变分自编码器的第一帧包含更密集的信息，且全局可见。

选择方法	峰值信噪比↑	学习感知图像块相似度↓	结构相似性↑	VCPR↑	时间（秒/迭代）↓
原始（4×8×8）	30.77	0.064	0.864	256.00	1.13
随机	29.82	0.077	0.843	598.13	**1.03**
H.264	30.51	0.070	0.852	421.23	1.07
基于峰值信噪比	**30.67**	**0.069**	**0.859**	492.00	1.22

表5。我们压缩算法的不同质量函数选择方法比较。基于峰值信噪比的方法在所有指标上都能达到最佳重建质量，同时仅小幅增加计算成本。

压缩算法的质量函数。在我们的压缩算法中，$Q()$用于评估视频质量，如散点图4 和我们的讨论所示。尽管可以使用 H.264（MPEG-4）、峰值信噪比或光流等指标，但我们观察到光流测量结果存在显著重叠，因此不适用于我们的需求。我们针对视频序列的质量—压缩权衡，评估了三种方法——随机选择、基于 H.264（MPEG-4）的选择和基于峰值信噪比的选择。与使用 H.264 分析视频压缩比相比，我们发现，利用变分自编码器视频重建的峰值信噪比作为 $Q()$ 更为合适。该方法不仅能获得更高的峰值信噪比和学习感知图像块相似度，还能提升 VCPR，同时仅需略高的计算预算。

5. 结论

在这项工作中，我们提出了 MTC-VAE，这是一种新方法，通过根据视频内容复杂度进行多级时间压缩，克服固定压缩率的局限性。MTC-VAE 能够根据视频内容复杂度动态调整时间压缩率，从而有效平衡压缩效率与高保真重建。该压缩算法可按 w 调整压缩率，从而精细控制时间分辨率。我们分析了视频压缩率与时间压缩率之间的关系，揭示了效率和性能方面的提升。我们使用 MTC-VAE 对 DiT 进行了微调，将自适应压缩框架与扩散变换器相结合。这种架构优化了潜在空间中的信息时间分布，减少了信息冗余。在视频重建与生成任务上进行的实验和消融研究表明，与固定比率压缩方法相比，我们的方法效果显著。

参考文献

[1] Max Bain、Arsha Nagrani、G¨ul Varol 和 Andrew Zisser- man。《时间冻结：用于端到端检索的联合视频与图像编码器》。载于 IEEE/CVF 国际计算机视觉会议论文集，第 1728–1738 页，2021年 5 月。[2]Fan Bao、Chendong Xiang、Gang Yue、Guande He、Hongzhou Zhu、Kaiwen Zheng、Min Zhao、Shilong Liu、Yaole Wang 和 Jun Zhu。《Vidu：采用扩散模型的高一致性、动态且娴熟的文生视频生成器》。arXiv 预印本 arXiv:2405.04233，2024年。2[3] Andreas Blattmann、Tim Dockhorn、Sumith Kulal、Daniel Mendelevitch、Maciej Kilian、Dominik Lorenz、Yam Levi、Zion English、Vikram Voleti、Adam Letts 等。《Stable Video Diffusion：将潜在视频扩散模型扩展至大型数据集》。arXiv 预印本 arXiv:2311.15127，2023年。1、2、5[4] Tim Brooks、Bill Peebles、Connor Holmes、Will DePue、Yufei Guo、Li Jing、David Schnurr、Joe Taylor、Troy Luh- man、Eric Luhman、Clarence Ng、Ricky Wang 和 Aditya Ramesh。《作为世界模拟器的视频生成模型》。2024年。1、2[5] Liuhan Chen、Yirou Wang 和 Yongyong Chen。《用于单图像盲去模糊的端到端 xy 分离》。载于第 31 届 ACM 多媒体国际会议论文集，第 1273–1282 页，2023年。2[6] Liuhan Chen、Zongjian Li、Bin Lin、Bin Zhu、Qian Wang、Shenghai Yuan、Xing Zhou、Xinhua Cheng 和 Li Yuan。《OD-VAE：用于改进潜在视频扩散模型的全维度视频压缩器》，2024年。2、5、6[7]Tsai-Shien Chen、Aliaksandr Siarohin、Willi Menapace、Ekaterina Deyneka、Hsiang-wei Chao、Byung Eun Jeon、Yuwei Fang、Hsin-Ying Lee、Jian Ren、Ming-Hsuan Yang 等。《Panda-70M：利用多个跨模态教师为 7000 万个视频生成字幕》。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 13320–13331 页，2024年。5[8] Patrick Esser、Robin Rombach 和 Bjorn Ommer。《驯服Transformer，实现高分辨率图像合成》。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 12873–12883 页，2021年。5[9] Songwei Ge、Thomas Hayes、Harry Yang、Xi Yin、Guan Pang、David Jacobs、Jia-Bin Huang 和 Devi Parikh。《借助时间无关VQGAN和时间敏感Transformer生成长视频》。载于欧洲计算机视觉会议论文集，第 102–118 页。施普林格，2022年。1、2、5[10] Yuwei Guo、Ceyuan Yang、Anyi Rao、Zhengyang Liang、Yaohui Wang、Yu Qiao、Maneesh Agrawala、Dahua Lin 和 Bo Dai。《AnimateDiff：无需特定调优即可动画化个性化文生图扩散模型》。载于第十二届国际学习表征会议。1、2[11] Wenyi Hong、Ming Ding、Wendi Zheng、Xinghan Liu 和 Jie Tang。《CogVideo：基于 Transformer 的文本到视频生成大规模预训练》。arXiv 预印本 arXiv:2205.15868，2022年。1

[12] Alain Hore与Djemel Ziou。图像质量指标：峰值信噪比与结构相似性。载于2010年第20届国际模式识别会议，第2366–2369页。IEEE，2010年。6[13] Ziqi Huang、Yinan He、Jiashuo Yu、Fan Zhang、Chenyang Si、Yuming Jiang、Yuanhan Zhang、Tianxing Wu、Qingyang Jin、Nattapol Chanpaisit、Yaohui Wang、Xinyuan Chen、Limin Wang、Dahua Lin、Yu Qiao与Ziwei Liu。VBench：面向视频生成模型的全面基准测试套件。载于IEEE/CVF计算机视觉与模式识别会议论文集，2024年。6[14] Diederik P Kingma与Jimmy Ba。Adam：一种随机优化方法。国际学习表征会议，2015年。5[15] Diederik P Kingma与Max Welling。自编码变分贝叶斯。stat，1050:1，2014年。1, 2[16]Rob Koenen。MPEG-4标准概述。国际标准化组织，ISO/IEC JTC1/SC29/WG11 N2323，活动图像和音频编码，瑞士日内瓦，2002年。7[17] 北大-袁实验室与图湛AI等。Open-Sora-Plan，2024年。1, 5, 6, 7[18]Zhimin Li、Jianwei Zhang、Qin Lin、Jiangfeng Xiong、Yanxin Long、Xinchi Deng、Yingfang Zhang、Xingchao Liu、Minbin Huang、Zedong Xiao、Dayou Chen、Jiajun He、Jiahao Li、Wenyue Li、Chen Zhang、Rongwei Quan、Jianxiang Lu、Jiabin Huang、Xiaoyan Yuan、Xiaoxiao Zheng、Yixuan Li、Jihong Zhang、Chao Zhang、Meng Chen、Jie Liu、Zheng Fang、Weiyan Wang、Jinbao Xue、Yangyu Tao、Jianchen Zhu、Kai Liu、Sihuan Lin、Yifu Sun、Yun Li、Dongdong Wang、Mingtao Chen、Zhichao Hu、Xiao Xiao、Yan Chen、Yuhong Liu、Wei Liu、Di Wang、Yong Yang、Jie Jiang与Qinglin Lu。混元-DiT：具备细粒度中文理解的强大多分辨率扩散变换器，2024年。1, 2[19] Xin Ma、Yaohui Wang、Gengyun Jia、Xinyuan Chen、Ziwei Liu、Yuan-Fang Li、Cunjian Chen与Yu Qiao。Latte：用于视频生成的潜空间扩散Transformer。arXiv预印本 arXiv:2401.03048，2024年。1, 2[20] Henrique Morimitsu、Xiaobin Zhu、Xiangyang Ji与Xucheng Yin。用于高效光流估计的循环部分核网络。载于AAAI，2024年。5[21] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga等。PyTorch：一种命令式、高性能深度学习库。神经信息处理系统进展，32，2019年。6[22] William Peebles与Saining Xie。基于Transformer的可扩展扩散模型。载于IEEE/CVF国际计算机视觉会议论文集，第4195–4205页，2023年。2 – 。[23] Dustin Podell、Zion English、Kyle Lacey、Andreas Blattmann、Tim Dockhorn、Jonas M¨uller、Joe Penna与Robin Rombach。SDXL：改进潜在扩散模型以实现高分辨率图像合成。arXiv预印本arXiv:2307.01952，2023年。1[24] Zhiwu Qing、Shiwei Zhang、Jiayu Wang、Xiang Wang、Yujie Wei、Yingya Zhang、Changxin Gao与Nong Sang。Hier-

archical时空解耦用于文本生成视频。载于IEEE/CVF计算机视觉与模式识别会议论文集，第6635–6645页，2024年。1 – 。[25]Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser与Bj¨orn Ommer。基于潜在扩散模型的高分辨率图像合成。载于IEEE/CVF计算机视觉与模式识别会议论文集，第10684–10695页，2022年。1, 2, 5[26] Olaf Ronneberger、Philipp Fischer与Thomas Brox。U-Net：用于生物医学图像分割的卷积网络。载于医学图像计算与计算机辅助干预——MICCAI 2015：第18届国际会议，德国慕尼黑，2015年10月5日至9日，论文集第III部分18，第234–241页。施普林格，2015年。2[27] Masaki Saito、Shunta Saito、Masanori Koyama与Sosuke Kobayashi。稀疏训练，密集生成：高分辨率时序生成对抗网络的内存高效无监督训练。《国际计算机视觉杂志》，128(10):2586–2606，2020年。5[28] Du Tran、Lubomir Bourdev、Rob Fergus、Lorenzo Torresani与Manohar Paluri。利用三维卷积网络学习时空特征。载于IEEE国际计算机视觉会议论文集，第4489–4497页，2015年。5 – 。[29] Aaron Van Den Oord、Oriol Vinyals等。神经离散表示学习。神经信息处理系统进展，30，2017年。1, 2[30] Shaodong Wang、Yunyang Ge、Liuhan Chen、Haiyang Zhou、Qian Wang、Xinhua Cheng与Li Yuan。Prompt2poster：仅凭提示自动进行艺术化中文海报创作。载于ACM Multimedia 2024。2[31] Xiang Wang、Hangjie Yuan、Shiwei Zhang、Dayou Chen、Jiuniu Wang、Yingya Zhang、Yujun Shen、Deli Zhao与Jingren Zhou。VideoComposer：具有运动可控性的组合式视频合成。神经信息处理系统进展，36，2024年。1[32] Zhou Wang、Alan C Bovik、Hamid R Sheikh与Eero P Simoncelli。图像质量评估：从误差可见性到结构相似性。《IEEE图像处理汇刊》，13(4):600–612，2004年。6[33] Wilson Yan、Yunzhi Zhang、Pieter Abbeel与Aravind Srinivas。VideoGPT：使用向量量化变分自编码器和Transformer进行视频生成。arXiv预印本arXiv:2104.10157，2021年。1[34] Zhuoyi Yang、Jiayan Teng、Wendi Zheng、Ming Ding、Shiyu Huang、Jiazheng Xu、Yuanming Yang、Wenyi Hong、Xiaohan Zhang、Guanyu Feng等。CogVideoX：采用专家Transformer的文本生成视频扩散模型。arXiv预印本arXiv:2408.06072，2024年。1, 2, 5, 6[35] Lijun Yu、Yong Cheng、Kihyuk Sohn、Jos´e Lezama、Han Zhang、Huiwen Chang、Alexander G Hauptmann、Ming-Hsuan Yang、Yuan Hao、Irfan Essa等。MAGVIT：掩码生成视频Transformer。载于IEEE/CVF计算机视觉与模式识别会议论文集，第10459–10469页，2023年。1, 2[36] Lijun Yu、Jos´e Lezama、Nitesh B Gundavarapu、Luca Versari、Kihyuk Sohn、David Minnen、Yong Cheng、Agrim Gupta、Xiuye Gu、Alexander G Hauptmann 等人。《语言模型胜过扩散模型——分词器是视觉生成的关键》。国际学习表征会议，2024年。1，2[37] Shenghai Yuan、Jinfa Huang、Yujun Shi、Yongqi Xu、Ruijie Zhu、Bin Lin、Xinhua Cheng、Li Yuan 和 Jiebo Luo。《MagicTime：将延时视频生成模型作为形态变换模拟器》。*arXiv* 预印本 *arXiv:2404.05014*，2024年。1，2[38] Shenghai Yuan、Jinfa Huang、Yongqi Xu、Yaoyang Liu、Shaofeng Zhang、Yujun Shi、Ruijie Zhu、Xinhua Cheng、Jiebo Luo 和 Li Yuan。《Chronomagic-bench：文本转延时视频生成的形态变换评估基准》。*arXiv* 预印本 *arXiv:2406.18522*，2024年。2[39] Richard Zhang、Phillip Isola、Alexei A Efros、Eli Shechtman 和 Oliver Wang。《深度特征作为感知度量的惊人有效性》。载于 *IEEE*计算机视觉与模式识别会议论文集，第586–595页，2018年。6[40] Sijie Zhao、Yong Zhang、Xiaodong Cun、Shaoshu Yang、Muyao Niu、Xiaoyu Li、Wenbo Hu 和 Ying Shan。《CV-VAE：面向潜空间生成式视频模型的兼容视频变分自编码器》。*arXiv* 预印本 *arXiv:2405.20279*，2024年。1，2，5[41] 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》，2024年。1，2，5

MTC-VAE ：具备内容感知能力的多级时间压缩 S

补充材料

6. 算法描述

我们提出了用于视频压缩和生成框架的两种关键算法。算法 1 详细介绍 MTC-VAE训练过程，算法 2 则描述了变分自编码器+DiT文本到视频生成流程。

算法 1 MTC-VAE训练过程

要求： 视频 X，压缩率 $C = \{4, 8, 16\}$，权重 w
将 X 分割为等长的 $\{x_1, x_2, ..., x_N\}$
对所有 $x_i \in \{x_1, x_2, ..., x_N\}$ **执行**
$\boldsymbol{Q}(x_i) \leftarrow [Q(4, x_i)\ Q(8, x_i)\ Q(16, x_i)]$ {质量向量}
$Q(x_i) \leftarrow$ 均值($\boldsymbol{Q}(x_i)$)
$\sigma_Q(x_i) \leftarrow$std($\boldsymbol{Q}(x_i)$)
$\alpha(x_i) \leftarrow \frac{1}{2} \cdot \frac{Q(x_i) - Q_{\min}}{Q_{\max} - Q_{\min}} + \frac{1}{2} \cdot (1 - \frac{\sigma_Q(x_i)}{\sigma_{Q_{\max}}})$ {自适应权重}
结束循环
$\{c_1, c_2, ..., c_N\} \leftarrow \underset{c_i \in C}{\arg\max} \sum_{i=1}^{N} [(1 - \alpha(x_i)) \cdot Q(c_i, x_i) + \alpha(x_i) \cdot \log_2(c_i) \cdot w]$ {最优速率}
编码器：
对所有 x_i **执行**
$z_i \leftarrow$ 时间下采样(x_i^{init}, c_i) {跳过采样器基于 c_i}
$z_i^1 \leftarrow z_i^1 + f_{c_i}$ {将压缩嵌入添加到第一个帧}
结束循环
解码器：
对所有 z_i **执行**
$p_i \leftarrow$ 关键帧预测器(z_i) {关键帧概率-性}
$\hat{z}_i \leftarrow$ 时间上采样(z_i, c_i) {使用真实值 c_i}
$\hat{x}_i \leftarrow$ 解码($\hat{z}_i$) {重建片段}
结束循环
$\hat{X} \leftarrow$ 拼接($\hat{x}_1, \hat{x}_2, ..., \hat{x}_N$) {最终视频输出}
返回 $\hat{X}$

算法 1：MTC-VAE训练过程。该算法详细介绍了我们针对动态潜帧率的变分自编码器训练方法。我们针对每个视频片段的 4×、 8×和16× 压缩率确定质量指标，以得出自适应权重 $\alpha(x_i)$，从而平衡重建质量与效率。编码器根据内容复杂度选择最优压缩率，并将该速率数据嵌入潜在空间。解码器包含一个关键帧预测器，用于识别需要更高精度的帧。

算法 2变分自编码器+DiT文本到视频生成

要求： 文本提示 y，噪声调度 $\beta_1, ..., \beta_T$，数量扩散步数 T，片段长度 L
$z_T \sim \mathcal{N}(0, I)$ {采样初始噪声}
$\hat{y} \leftarrow$ 文本编码器(y) {编码文本提示}
扩散去噪：
对于 $t = T$ **到** 1 **执行**
$\epsilon_\theta \leftarrow$ DiT($z_t, t, \hat{y}$) {使用DiT预测噪声}
$\mu_t \leftarrow \frac{1}{\sqrt{\alpha_t}}(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta)$ {预测均值}
如果 $t > 1$**则**
$z_{t-1} \sim \mathcal{N}(\mu_t, \frac{\beta_t (1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} I)$ {采样下一个状态}
else
$z_0 \leftarrow \mu_1$ {最终去噪潜变量}
结束条件判断
结束循环
VAE 解码：
$p \leftarrow$ 关键帧预测器(z_0) {预测关键帧概率}
$K \leftarrow \{i \mid p_i >$ 阈值$\}$ {确定关键帧位置}
划分 z_0 into 片段 $\{z_1, z_2, ..., z_M\}$ 基于 on 关键帧位置 K
对所有片段 z_i **执行**
$l_i \leftarrow$ Length(z_i) {当前潜在片段的长度}
$c_i \leftarrow L / l_i$ {计算压缩率}
$\hat{z}_i \leftarrow$ 时间上采样(z_i, c_i) {基于压缩率}
$x_i \leftarrow$ VAE解码($\hat{z}_i$) {将潜在表示解码到像素空间}
结束循环
$X \leftarrow$ 拼接(x_1， x_2，.....， x_M) {合并片段为最终视频}
返回 X

算法 2：变分自编码器+DiT 文本到视频生成。该算法概述了我们使用扩散变换器（DiT）结合 MTC-VAE 框架进行文本到视频生成的流程。从随机噪声和文本提示开始，我们使用以文本嵌入为条件的 DiT 模型进行迭代去噪。随后，我们的关键帧预测器会处理去噪后的潜在表示 z_0，以确定最佳片段边界。每个片段 z_i 都会根据计算出的压缩率 c_i进行上采样，该压缩率通过比较潜在片段长度与目标输出长度得出。最后，VAE解码器会分别重建各个片段，再将它们拼接成最终视频输出。

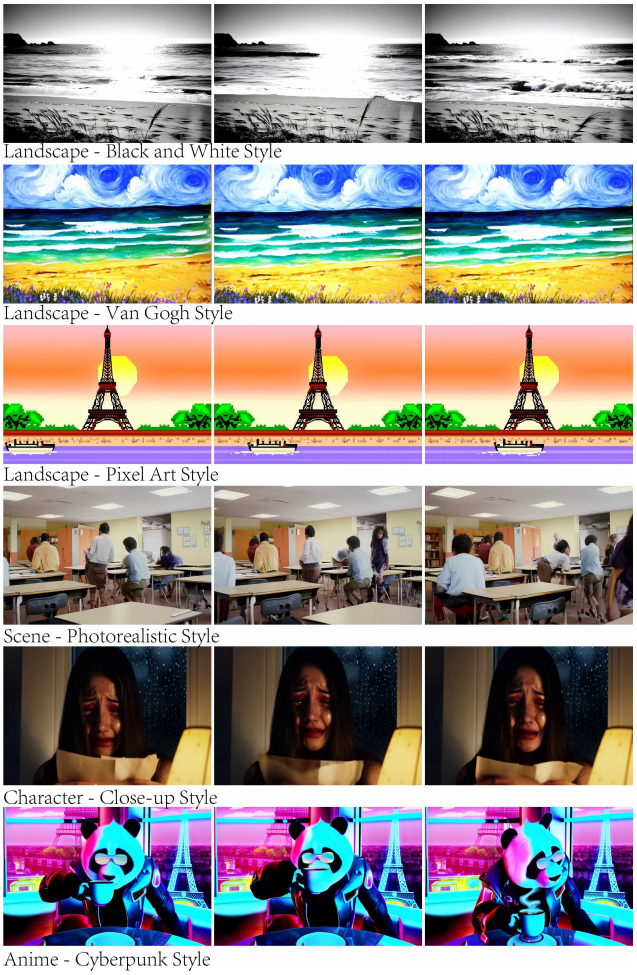


图5. 我们的 MTC-VAE 模型在不同场景中的表现对比。尽管压缩率各不相同，每一行均展现出一致的风格保留效果，证明模型能够在多种内容类型中保持稳定的视觉质量。

7. VBench 上的其他结果

我们在 VBench 基准上进行了额外评估，以考察模型在不同压缩率下保持视觉一致性的能力。图5展示了 MTC-VAE 在采用不同时间压缩率时保持视觉外观的表现。

如图所示，我们的模型能够在多种场景中成功保持视觉风格一致，从黑白海滩景观到色彩搭配鲜艳的风格化绘画、照片级写实建筑结构，以及具有艺术风格的角色动画。这些结果证实，我们的自适应压缩方法在降低计算需求的同时，不会牺牲视觉质量或风格一致性，展现出在 VBench 多样化评估标准下的稳健性。

8. 影响声明

本文旨在推动机器学习领域的发展，尤其是视频生成领域。通过引入 MTC-VAE（多级时序压缩变分自编码器），我们提出了一种易于迁移的框架，可根据视频内容复杂度调整潜在帧率，同时保持较高的重建质量。这项创新有望提高视频生成的效率和可扩展性，使得在减少计算资源的情况下合成更长、分辨率更高的视频。

8.1. 局限性与未来工作

我们承认当前方法存在一些局限。例如，为了更好地利用现有的采样参数，我们采用了三种离散压缩率。未来工作可以探索使用注意力块训练完整的变分自编码器和 DiT 模型，以实现任意压缩率，从而支持更精细的压缩。这将使各组件之间的集成更加顺畅，并有望让 DiT 根据文本输入自主控制视频压缩率。

我们的关键帧预测器将压缩率预测视为二元分类问题，因此准确率较高，但在生成过程中仍可能出现预测错误。对于预测置信度低于 0.6 的帧，我们默认采用前一个片段的压缩率（若为第一个片段，则采用 256× ）。我们认为，这些局限部分源于预训练不足和计算资源受限，并鼓励未来工作通过更广泛的训练方案应对这些挑战。

8.2. 更广泛的影响

我们的研究有助于让先进的视频生成技术更易于获取，同时更具环境可持续性。通过降低计算需求，MTC-VAE 使资源有限的研究人员也能参与前沿视频合成研究。这种技术普及有望推动创意艺术、科学可视化等多个领域的创新。

我们认识到，高效的视频生成技术具有双重用途的潜力。虽然这类技术能够用于改进教育内容和医学模拟等有益应用，但也可能助长用于欺骗的合成媒体制作。我们倡导持续开发检测工具和水印技术，并推行透明的披露实践，以降低潜在的滥用风险。我们的方法具备较高的计算效率，也契合人们日益关注的人工智能环境足迹问题，有望降低视频生成系统的碳影响。我们将继续致力于负责任的创新，在技术进步与伦理考量之间实现平衡。