arXiv:2611.30714v1 [cs.SD] 23 Nov 2026

LiveAudio：从双向模型到用于实时流式视频到音频生成的自回归扩散

Yuxuan Jiang[1] Zehua Chen[1] Yusheng Dai[2] Andong Wang[3] Weibei Dou[1] Jun Zhu[1], †[1]清华大学 [2]莫纳什大学 [3]中国人民大学

摘要

视频到音频生成能够合成在内容和时间上与视频相匹配的声音，应用涵盖电影后期制作、直播、游戏和交互式世界模型。交互式应用会持续接收视频，并需要声音实时跟随视频，但大多数现有视频到音频模型都是双向模型，必须观察完整片段后才能生成声音。我们提出 *LiveAudio*，一种流式视频到音频框架，可在视频播放时实时生成高质量且同步的音频。与从头开始训练的以往流式模型不同，*LiveAudio* 将双向模型转换为流式生成器，从而继承其音频质量和同步性能。*LiveAudio* 按块生成音频，每个块仅依赖已经播放的视频。通过渐进式因果微调和蒸馏，流式生成器由双向模型转换而来，因此每个块只需少量采样步数。*VGGSound* 上的实验表明，*LiveAudio* 在音频质量和同步方面优于现有流式模型，并且在同步方面超过了作为其基础的双向模型。在单张 *A100 GPU* 上，它以 *0.79* 的实时因子实时运行，首帧时间为 *143* 毫秒。音频样本可在 https://streamaudio.github.io/。

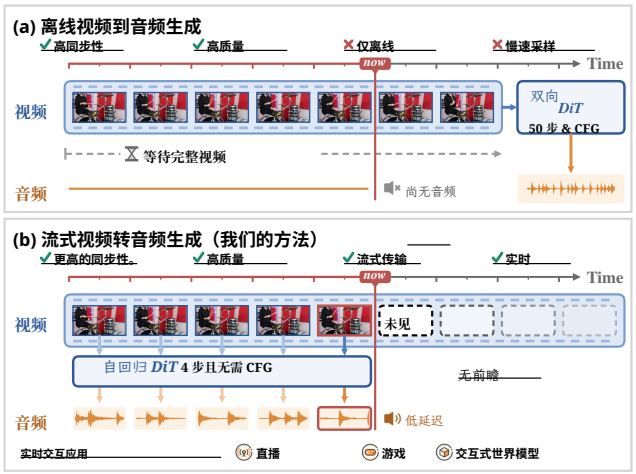

图 1. 同一时刻下，(a) 离线V2A生成与(b) 使用LiveAudio进行流式V2A生成的对比。

1. 引言

视频到音频（V2A）生成旨在合成与视频内容和时间同步的声音。近期的双向扩散 Transformer（DiT）[7, 11, 51, 64]显著提升了音频质量和音视频同步效果。这些模型以离线方式运行，因为每个音频位置都会关注整个视频和音频片段，而且只有在整个视频都准备好后才能开始生成。然而，新兴的交互式应用，例如直播 [29, 69]，游戏 [3] 和交互式世界模型 [2, 25]，逐帧传输视频，并要求声音实时跟随视频。如图 1所示，这一设置要求实时流式V2A生成：音频只能依赖截至当前已播放的视频，并且生成速度至少要与视频播放速度相同。

现有的流式V2A模型通过从头训练因果模型来应对这一设置。作为早期尝试，V-AURA [73]根据逐段提取的视频特征，自回归地生成离散音频令牌。最近，SoundReactor [63]，SwanSphere [43]，WorldSonus [19]和LiveFoley [12] 将因果自回归主干网络与局部扩散头相结合，并随着视频帧到达而生成音频。其中，LiveFoley实现了可与离线模型 [7, 11, 51, 64]，相媲美的同步效果，其同步对齐的视频条件为流式V2A生成开辟了一个有前景的方向。然而，此类模型无法利用双向模型已经实现的音频质量和同步效果。这两方面都必须重新学习，通常还需要额外的训练目标。

视频生成采取了不同的路径。一系列研究通过因果

†通讯作者。

微调和蒸馏 [28, 65, 78, 81]，以及音频驱动的数字人系统 [29, 37, 69, 79]和联合音视频生成器 [36, 68] 推动了这一路径实现实时流式传输。这一路径使流式生成器能够建立在双向模型之上，但尚未在视频转音频生成中得到探索，而这一领域面临三个挑战。首先，视频条件会随时间到达，因此生成器可能只能使用已经播放的帧，而双向视频转音频模型接收的视频特征则涵盖后续帧。其次，因果生成器无法观测后续音频或视频，我们发现这会降低视听同步性。第三，音频必须实时生成，输入视频也必须在相同的时间预算内完成编码，而仅采样就需要数十步 [53, 55, 66] 并结合无分类器引导（CFG） [27]。

为应对这些挑战，我们提出 LiveAudio，这是一种实时流式视频转音频框架，将问题拆解为因果性和效率，并从双向 V2A 模型出发逐步解决。针对因果性，LiveAudio 被构建为一种自回归扩散模型，采用块因果 DiT [61] 和因果音频 VAE [63], ，音频按块生成，每个块解码后立即播放。视频条件 [31] 也通过同步对齐的视频条件 [12], 实现因果化，使每个音频词元只能接收已经播放的视频。我们在这种条件下训练双向模型，然后通过块因果微调使其具备因果性。由此得到的因果 V2A 模型能够随着视频到达而生成音频，但每个块仍需通过 CFG 进行多步采样。为提高效率，我们通过常微分方程回归 [28, 78, 81]对因果模型进行蒸馏，将其转化为无需 CFG、可实时运行的少步生成器。最后，我们采用非对称分布匹配蒸馏（DMD） [78, 81], ，让双向模型监督因果生成器根据自身历史生成的音频，从而在不损失因果性或效率的前提下恢复同步。

我们在 VGGSound 测试集 [5],上评估 LiveAudio，结果表明，它在音频质量和同步方面均优于现有流式视频转音频模型。它的同步性能不仅优于其起点双向模型，也超过了我们对比的离线模型，同时音频质量仍具有竞争力。在效率方面，LiveAudio 可在单个 A100 GPU 上实时运行，首帧时间为 143 毫秒，实时因子为 0.79。消融研究表明，流程中的每个阶段都不可或缺；同一模型还支持流式视频文本到音频（VT2A）和文本到音频（T2A）生成。我们的贡献总结如下：

• 我们提出了 LiveAudio，这是一种实时流式 V2A 框架，可将双向 V2A 模型转换为自回归扩散生成器，与现有从零开始训练的流式模型形成对比。

• 我们通过渐进式训练流程，将这一转换拆解为因果性和效率两个方面，并表明非对称 DMD 能够恢复因果性导致的同步损失，并将同步效果提升至优于双向模型的水平。

• LiveAudio 在 VGGSound 上的音频质量和同步方面优于现有流式视频转音频模型，可在单个图形处理器上实时运行，并且也通过了流式 VT2A 和 T2A 生成的验证。

2. 相关工作

2.1. 视频到音频生成

先前的工作探索了使用潜在扩散模型进行 V2A 生成：一种方法以对比学习得到的视听特征为条件 [56] ，另一种方法则通过向 T2A 模型添加视觉条件构建 [6, 13, 32, 80]。近期的双向 DiT [7, 11, 51, 64] 通过与文本转音频数据联合训练，使单个模型支持视频、文本和联合条件。模型通过与音频时间对齐注入的视频特征来确定时间信息，通常使用来自 Synchformer 的同步特征 [31], ；语义信息则来自全局视频特征和文本。由于整个片段中的音频和视频彼此可见，这些模型都是离线模型。从 V-AURA [73]到 SoundReac-tor [63], 、SwanSphere [43], 、WorldSonus [19]和 LiveFo-ley [12], ，流式模型则选择从头开始训练因果模型，因此其音频质量和同步能力都是重新学习的，而非继承自双向模型。

2.2. 流式多模态生成

实时交互应用需要流式生成，即模型会随着输入到达而生成输出。在视频生成领域，基于 Diffusion Forcing [4], 的一系列研究已确立了如何将双向扩散模型转变为因果的少步自回归生成器 [28, 65, 78, 81]。这一方法已扩展至条件本身随时间到达的场景，包括音频驱动的虚拟形象生成 [29, 37, 69, 79], 以及音视频联合生成 [36, 68]。然而，这一方法尚未应用于 V2A 生成，而本文正是针对这一问题展开研究。

2.3. 少步扩散蒸馏

少步扩散模型主要采用基于轨迹的方法进行训练，这类方法学习在一步或少数几步内到达采样轨迹的终点 [21, 67], ，或采用 DMD [47, 76, 77], ，将学生模型的输出分布拉近教师模型的输出分布，而不模仿其轨迹。在音频生成中，基于轨迹的方法已在 T2A [1, 44, 49, 52]和 V2A [24], 中实现一步至少步采样，DMD 也已应用于音乐 [59]、声音 [71]和语音 [45], ，通常在真实音频上加入对抗项，以提升感知质量 [59, 71]。蒸馏后的学生模型甚至可以超越教师模型 [45, 76]。对抗

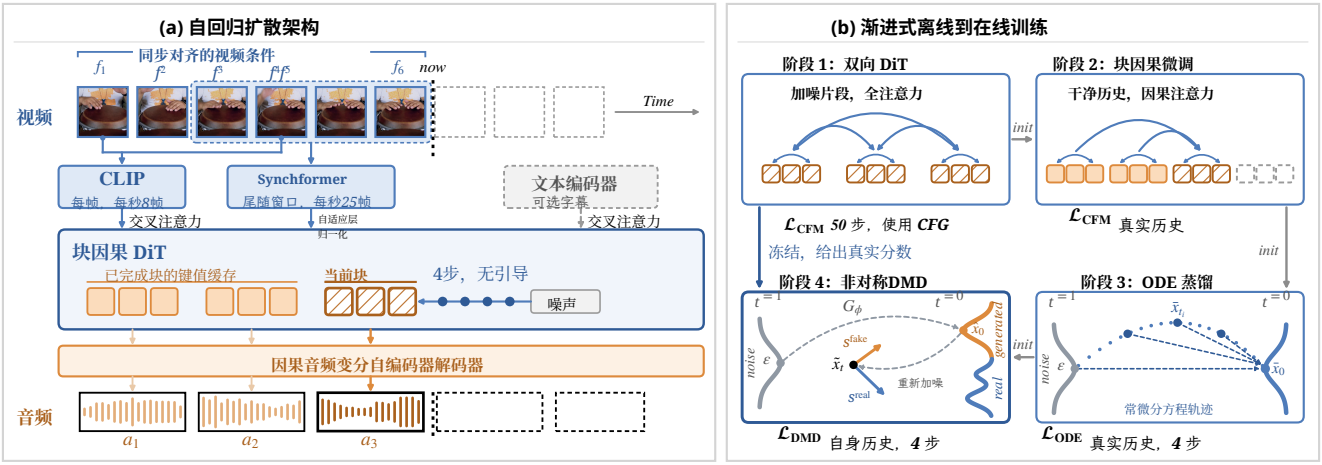

图2. LiveAudio 概述。(a) 音频按块生成，并在生成后立即解码；同步对齐的视频条件使每个块仅依赖于截至当前已播放的视频。(b) 四个渐进式训练阶段将双向模型转变为无需引导的四步自回归生成器，先使其具备因果性，再对其进行蒸馏。

性后训练还可以在没有教师模型的情况下，利用真实数据得到少步模型 [58],，并扩展到块因果流式音乐生成 [60]。然而，在这些方法中，教师模型和学生模型观察到相同的上下文，或者不使用教师模型。非对称蒸馏中，双向教师模型为因果学生模型提供监督，这种方法已在流式视频生成中得到确立 [78, 81],，但其对视听同步的影响尚未得到研究。

3. LiveAudio

3.1. 问题设定

实时流式视频到音频生成。 令 y 表示文本条件，$f_{1:T}$ 表示同一时间线上的视频帧，$x_{1:K}$ 表示音频潜变量，并将其划分为 K 个块，其中块 x_k 覆盖 $[s_k, e_k)$ 。V2A 生成对条件分布 $p(x_{1:K} \mid f_{1:T},\ y)$ 进行建模；离线模型通过双向注意力实现这一点，因此每个音频位置都能观察到所有视频帧和完整音频上下文。实时流式处理增加了两个要求。第一是因果性。音频按块生成，当视频播放至时间 τ_k 后，块 k 才会生成。它只能依赖已经到达的帧 $f_{\le \tau_k}$、文本以及先前生成的音频。该块的每个潜变量都从 τ_k 之前开始，因此声音绝不会先于其对应的视频出现。因此，流式生成器具有如下形式：

$$p_\theta(x_{1:K} \mid f_{1:T}, y) = \prod_{k=1}^{K} p_\theta(x_k \mid c_k), \tag{1}$$

其中 $c_k = (y, x_{<k}, f_{\le \tau_k})$ 是块 k 的上下文。第二点是实时运行。每个块的计算不得超过块持续时间，从而使一个块的延迟保持为其视频等待时间 $\tau_k - s_k$ 加上其计算时间，并且不会随时间累积。

条件流匹配。 流匹配学习一个将高斯噪声传输至数据的速度场 [30, 35, 46,53]。双向模型和因果模型均以其条件形式进行训练，前者在完整片段上训练，后者在带有上下文 c_k的块 x_k 上训练。对于干净样本 x 和噪声 $\epsilon \sim \mathcal{N}(0, I)$，线性路径为：

$$x_t = (1-t)\,x + t\,\epsilon, \qquad t \in [0,1], \tag{2}$$

其中 t 越大，表示噪声越多；速度模型 v_θ 通过回归该路径 [7] 的速度进行训练：

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{t,\,x,\,\epsilon}\left[\left\| v_\theta(x_t, c, t) - (\epsilon - x) \right\|_2^2\right]. \tag{3}$$

对于 CFG [27],，训练期间会将 c 中的视频和文本随机替换为空条件，从而得到上下文 $\varnothing$。在推理时，通过使用引导尺度 w求解以下常微分方程，从 $t=1$ 到 $t=0$，即可从 ϵ 中获得一个样本：

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(x_t, \varnothing, t) + w\left[v_\theta(x_t, c, t) - v_\theta(x_t, \varnothing, t)\right]. \tag{4}$$

分布匹配蒸馏。 DMD [76, 77]通过匹配输出分布而非采样轨迹，将多步教师模型蒸馏为少步生成器。生成器 G_ϕ 将噪声 ϵ和上下文 c映射为一个样本 $\hat{x} = G_\phi(\epsilon, c)$。将 $\hat{x}$ 重新加噪至噪声级别 t，根据公式 2并使用新的噪声 ϵ' 得到 $\tilde{x}_t$。DMD 最小化生成器与教师模型在噪声级别 t 下分布之间的期望反向 KL 散度，其梯度为：

$$\nabla_\phi \mathcal{L}_{\mathrm{DMD}} = \mathbb{E}_{t,\,\epsilon,\,\epsilon'}\left[\left(s_{\mathrm{fake}} - s_{\mathrm{real}}\right)(\tilde{x}_t, c, t)\,\frac{\partial \tilde{x}_t}{\partial \phi}\right], \tag{5}$$

其中 s_{real} 和 s_{fake} 分别是两个分布的分数，称为真实分数和虚假分数。教师模型被冻结，并根据公式 4的引导给出真实分数；而在生成器的

样本上使用公式 3在线训练的伪分数模型则给出虚假分数。两个分数均根据两个模型的速度预测计算得出。

3.2. 自回归扩散架构

图2 （a）展示了 LiveAudio 的自回归扩散架构。为满足因果性要求，音频由因果音频 VAE 编码，并由块级因果 DiT 生成，从而每个块一经生成即可解码和播放。视频条件也通过同步对齐的方式注入，使其具备因果性，因此每个块仅依赖于截至当前已经播放的视频。

因果音频 VAE。 现有 T2A 和 V2A 系统中的音频 VAE 使用非因果卷积 [7, 11, 18, 48],，因此解码某个潜变量需要依赖其后续的潜变量。于是，只有在生成后续块之后，才能忠实地解码某个块；若提前解码，则会产生边界伪影。相反，我们采用 SoundReactor [63]的因果音频 VAE，并在整个训练过程中将其冻结。它以 30 Hz 的频率将 48 kHz 立体声音频编码为 64 维潜变量。其编码器和有状态解码器仅依赖过去的上下文，因此，一个潜变量块一经生成即可解码为波形，无需重新解码之前的块。

块因果 DiT。 我们使用扩散 Transformer [61] 作为骨干网络，并且无需增加参数即可支持流式传输。每个潜变量对应一个音频词元，我们将连续的三个词元分组为一个时长为 0.1 秒的块。自注意力掩码采用块因果方式 [28, 78], , 使词元能够关注其所属块及更早块中的所有词元，但无法关注后续块中的词元。双向模型会在开头添加一个携带整段音频噪声级别的词元，而我们为每个块都提供一个这样的词元，因此不同块可以处于不同的噪声级别。在训练期间，对当前块添加噪声，而较早的块保持干净。在推理时，音频在相同掩码下逐块生成，已完成的块通过 KV 缓存作为后续块的条件。让单个块覆盖整段音频即可恢复双向模型，因此两者之间可以直接迁移权重，无需任何架构变更。

同步对齐的视频条件。 我们保留 MMAudio [7]和 Omni2Sound [11], 的两条视频路径：Synchformer [31] 特征通过自适应层归一化调制每个音频词元，而 CLIP [62]特征则与文本一同通过注意力机制输入。对于流式传输，我们改变两条路径的输入方式，使词元只接收已经播放过的视频。我们将连续的两帧视频称为一个切片。对于 Synchformer，我们采用 LiveFoley [12], 的尾随滑动窗口，每播放完一个切片就对最近的帧进行编码；每个音频词元则使用包含其起始位置的切片特征。对于以较低帧率采样的 CLIP，词元的交叉注意力仅涵盖在其切片结束时已经到达的帧，而文本则始终可见。块 k 会在其中所有词元对应的切片均播放完毕时生成。因此，一个块只需等待其最后一个词元对应的切片，无需进一步前瞻，所以视频等待时间 $\tau_k - s_k$ 接近一个块的时长。由此，时序提示是局部的，并随每个块的帧一同到达，而语义上下文则随着视频播放逐渐累积。

3.3. 渐进式离线到在线训练

离线双向模型可以生成高质量且同步良好的音频，但它既不具备因果性，也无法实时运行。如图2 (b)所示，我们将其分四个阶段逐步改造为实时流式生成器， [34]每个阶段（第一阶段除外）都从上一阶段的权重开始，并且只改变一项属性。我们首先训练双向模型作为起点，然后使其具备因果性，再将因果模型蒸馏为可实时运行的少步生成器，最后通过非对称 DMD 恢复其同步能力。

阶段 1：双向 DiT 预训练。我们使用流匹配（公式3），在因果 VAE [63]的潜空间中，基于视频-音频、视频-文本-音频和文本-音频数据训练双向 DiT [18] 。 [7] 在 MMAudio [11], 和 Omni2Sound 中，音频词元接收的 Synchformer 和 CLIP 特征涵盖了后续帧。我们则对两者都采用同步对齐的视频条件，使词元只接收已经播放的视频。因此，我们的双向 DiT 仅在音频词元之间的注意力上是双向的，而其视频条件本身已具备因果性，并且在后续所有阶段中保持不变。

阶段 2：块因果微调。 在这一阶段，我们使音频生成具有因果性，其他方面均保持不变。我们将双向 DiT 切换为块因果结构，并使用相同的目标函数、数据和视频条件进行微调。块 k 现在只能观察上下文 c_k ，即公式1中的内容，包含此前的块，但不包含之后的块。训练采用教师强制 [81], , 其中历史 $x_{<k}$ 为干净的真实值，只有当前块添加噪声。每个块均独立采样噪声级别，并行监督片段中的所有块。由此得到的因果 DiT 逐块生成音频，但仍然是一个多步流模型。

阶段 3：用于提升效率的因果ODE蒸馏。 因果 DiT 仍无法实现实时运行，因为每个块都需要许多采样步数，并且每一步都要对 CFG 评估模型两次。因此，我们采用 ODE回归 [28, 78,81] 将其蒸馏为少步生成器 G_ϕ ，该生成器无需引导。使用因果 DiT 求解方程 4 可得到一条从噪声到干净块的确定性轨迹，而引导已包含在该轨迹中。我们利用真实历史提取此类轨迹，并存储其

表 1。VGGSound 测试集上的视频到音频结果。离线模型只有在整个视频可用后才会生成，因此仅报告流式模型的首帧生成时间。标有 †的结果引自原论文。教师模型是我们的双向 DiT。

方法	分布匹配					音频质量		模态对齐			效率	
	FD_O↓	FD_V↓	FD_P↓	KL_S↓	KL_P↓	IS_P↑	PQ↑	不同步度↓	IB↑	CLAP↑	首帧生成时间（秒）↓	RTF↓
离线												
FRIEREN	222.58	1.35	11.73	2.95	2.74	12.12	5.90	0.849	0.228	0.227	-	0.13
MMAudio	65.85	0.76	4.57	1.93	2.10	14.20	5.90	0.457	0.327	0.308	-	0.37
AudioX	77.75	1.56	12.51	3.14	2.93	13.68	6.13	1.213	0.250	0.215	-	1.37
Omni2Sound	143.81	0.63	6.23	2.11	2.16	16.70	6.08	0.501	0.338	0.284	-	1.06
LiveAudio（教师模型）	68.17	1.29	5.11	1.86	1.99	15.81	6.26	0.484	0.324	0.260	-	0.50
流式传输												
V-AURA	82.76	2.88	14.71	2.24	2.46	10.10	6.00	0.678	0.279	0.252	7.148	2.31
SoundReactor†	-	-	-	2.37	-	-	-	1.030	0.240	-	-	-
WorldSonus	90.24	1.67	15.59	2.44	2.54	9.09	6.28	0.659	0.258	0.197	0.502	0.60
LiveFoley†	-	-	9.92	-	2.30	11.42	6.26	0.480	0.290	-	0.156	-
LiveAudio（我们的方法）	70.97	1.30	7.32	1.99	2.12	15.87	6.41	0.395	0.323	0.252	0.143	0.79

状态 $\bar{x}_{k,\,t_i}$ 在 t_i 少步采样器的步进时间点上，连同其端点 $\bar{x}_{k,\,0}$。生成器以因果 DiT 初始化，并经过训练，以根据随机步的存储状态预测端点：

$$\mathcal{L}_{\mathrm{ODE}} = \mathbb{E}_{\bar{x},\,k,\,i}\left[\left\|G_\phi(\bar{x}_{k,t_i}, c_k, t_i) - \bar{x}_{k,0}\right\|_2^2\right], \quad (6)$$

其中，$G_\phi(\cdot,\ c_k,\ t_i)$ 表示生成器的一次前向传播。该生成器现在无需引导即可通过几步生成一个块，但它从未以自身的输出作为条件。

阶段 4：用于同步的非对称 DMD。 ODE 蒸馏生成器 G_ϕ 具有因果性，只需几步即可生成每个块，但其视听同步效果仍不如双向 DiT。这是因为因果模型无法观察后续的音频或视频，而且迄今为止，生成器仅使用真实值历史进行训练，而非使用自身的输出。因此，我们采用非对称 DMD [78, 81], , 由基于真实音频训练、并对整个片段进行注意力计算的双向 DiT，来监督因果生成器的输出。DMD 匹配的是分布，而非单条轨迹，因此教师模型不必观察与生成器相同的上下文。生成器逐块生成整个片段，其中块 $\hat{}x_{<k}$ 的历史 k 来自生成器自身的输出，而非真实值 [28]。随后对生成片段重新加噪，并根据公式5的梯度更新生成器，其中真实分数由带引导的双向 DiT 提供，虚假分数则由其副本提供，该副本在生成的片段上进行在线训练。因此，双向 DiT 会整体评估每个生成片段，使生成器在保持因果性的同时，学会生成同步良好的音频。我们还加入对抗损失，通过伪分数模型特征上的判别器头，将真实音频潜变量与生成的音频潜变量区分开来 [59, 76]。

3.4. 分块式 KV 缓存推理

在推理时，生成器通过依次生成各个块来实现公式 1的因子分解。实时运行要求每个块的计算成本受其时长限制，因此不能针对每个新块重新计算历史 $\hat{}x_{<k}$。由于自注意力采用块因果机制，已完成块的键和值独立于所有后续块，因此会保留在 KV 缓存中 [28, 78]。因此，块 k 在时间 τ_k 生成如下：

$$\hat{x}_k = G_\phi\big(\epsilon_k;\ y,\ f_{\le\tau_k},\ \mathrm{KV}(\hat{x}_{<k})\big), \quad (7)$$

其中 $\epsilon_k \sim \mathcal{N}(0, I)$ 是初始噪声，而 $\mathrm{KV}(\hat{}x_{<k})$ 表示已完成块的缓存的自注意力键和值。生成器仅处理块 k的词元，这些词元会关注缓存。它运行少步采样器，根据式 2使用新的噪声，在预测干净块与将预测结果重新加噪至下一步时间之间交替进行。随后，通过对干净块进行一次额外的前向处理获得已完成块的键和值，并将其附加到缓存中，从而保持缓存历史记录的干净状态。已完成块由因果 VAE 解码并立即播放，同时持续增量计算输入视频切片的特征。

4. 实验

4.1. 实验设置

数据集。 我们在同一个语料库上训练全部四个阶段。视频-音频数据来自 SoundAtlas [11], , 其中包含来自 VGGSound 的 47 万个视频-音频-文本对 [5] 和 AudioSet [20], , 文本-音频数据来自 AudioCaps [39]和 WavCaps [57]。我们在 VGGSound 测试集上评估 V2A 和 VT2A 生成，其中 VT2A 的文本提示来自 VGGSound-Omni [11], , 并在 AudioCaps 测试集上评估 T2A生成。更多详情见附录 A。

评估指标。 我们遵循 AudioLDM [48], StableAudio Open [18] 和 MMAudio [7], 的评估协议，并评估分布匹配、音频质量、模态对齐和效率。对于分布匹配，我们报告生成音频与真实音频在 OpenL3 [9], 和 PANNs [26]嵌入上的 Fréchet 距离，分别记为 FD [40]、FD_O和 FD_V。我们还报告 PaSST [P]和 PANNs 对生成音频与真实音频所预测的类别分布之间的 KL散度，分别记为 KL [41] 和 KL_S。对于音频质量，我们报告使用 PANNs 计算的初始分数（IS_P）以及由 Audiobox Aesthetics 预测的制作质量（PQ）[P]。对于模态对齐，我们报告不同步度分数（DeSync），即由 Synchformer 预测的音频-视频偏移绝对值均值 [72]。我们还报告视频与生成音频之间的 ImageBind [31]相似度（IB），以及生成音频与字幕之间的 CLAP [23] 相似度。对于效率，我们报告首帧生成时间（TTFF），即从视频流开始到首个音频输出所测量的时间，以及实时因子（RTF），即处理时间与生成音频时长之比。我们进一步邀请 20 名参与者开展主观评估，并报告整体音频质量（MOS-Q）和时间对齐（MOS-T）的平均意见得分。所有指标的详细信息见附录 B。

实现细节。 我们使用一个 DiT [18]，其包含24层、24个注意力头，宽度为1536，约有12亿个参数，并在10秒片段上进行训练。Synchformer处理每秒25帧的视频，因此一个切片覆盖0.08秒；CLIP处理每秒8帧的视频。一个块产生的视频等待时间平均为0.11秒。可选的文本提示由冻结的 Flan-T5 base [8] 编码为最多128个词元。我们对双向 DiT训练50万步，对块因果 DiT微调10万步，通过ODE回归在131万个提取的轨迹上训练生成器3万步，并运行非对称DMD 3000次迭代。双向 DiT和块因果DiT均使用50个欧拉步和4.5的引导尺度进行采样，而生成器在无引导的情况下执行四步采样。我们在单个 NVIDIA A100 GPU上、批量大小为1的条件下测量首帧生成时间和RTF，处理时间涵盖视频解码、特征提取、采样和波形解码。更多细节见附录C。

4.2. 主要结果

视频到音频生成。我们的比较涵盖了具有代表性的离线模型，包括 FRIEREN [74], MMAu- dio [7], AudioX [70] 和 Omni2Sound [11], , 以及流式模型，包括 V-AURA [73], SoundReactor [63]和最新的 WorldSonus [19] 及 LiveFoley [12]。SoundReactor 和 LiveFoley 的结果引自其论文，其他模型则使用官方检查点和推理设置运行。我们还报告了双

表2。视频到音频生成的主观评估。我们报告了带有95%置信区间的平均意见分数。

方法	MOS-Q↑	MOS-T↑
真实值	$4.06_{\pm 0.25}$	$4.26_{\pm 0.17}$
离线		
FRIEREN	$3.28_{\pm 0.21}$	$3.47_{\pm 0.22}$
MMAudio	$3.80_{\pm 0.28}$	$3.69_{\pm 0.20}$
AudioX	$3.16_{\pm 0.21}$	$2.13_{\pm 0.23}$
Omni2Sound	$3.87_{\pm 0.26}$	$3.97_{\pm 0.19}$
流式传输		
V-AURA	$3.43_{\pm 0.24}$	$3.50_{\pm 0.18}$
WorldSonus	$3.35_{\pm 0.25}$	$3.42_{\pm 0.24}$
LiveAudio（我们的方法）	$3.66_{\pm 0.19}$	$4.06_{\pm 0.27}$

向 DiT 作为 LiveAudio（教师模型）。如表1所示，LiveAudio 的不同步度在所有模型中最低，且明显优于流式模型、离线模型及其教师模型。在语义对齐方面，LiveAudio 在流式模型中获得最高的 IB，接近离线模型的水平，其 CLAP 得分也与最佳流式模型相当。在分布匹配和音频质量方面，LiveAudio 始终优于流式模型，可与离线模型媲美，尽管它每次仅利用过去的上下文，通过四步生成一个块。在效率方面，LiveAudio 的运行速度快于实时速度，首帧生成时间也处于流式模型中的较低水平。表2中的主观评估结果与客观指标一致。LiveAudio 在所有模型中获得最高的 MOS-T，其 MOS-Q 在流式模型中最高，也接近最佳离线模型的水平。这些结果表明，由双向模型转换而来的流式生成器优于从头训练的模型。我们认为，其同步方面的优势源于非对称 DMD：双向 DiT 对因果生成器基于自身历史生成的音频进行监督。LiveAudio 还支持使用同一模型进行视频到音频生成（VT2A），此时可同时提供文本提示和视频，以指定所需的声音。完整结果见附录D.1。

文本到音频生成。我们进一步将 LiveAudio 与具有代表性的离线文本到音频模型进行比较，即 AudioLDM [48], Tango [22], AudioLDM 2 [50], Stable Audio Open [18], TangoFlux [30], MMAudio [7]以及 AudioX [70], 在表 [39] 3中 AudioCaps 测试集上的表现。我们还纳入了 AudioGen [42], 这一生成离散音频令牌的自回归模型，以及 WorldSonus [19], 这一流式传输模型。LiveAudio 在全部四个阶段均使用文本-音频对进行训练，因此同一模型还支持不含视频输入的流式传输 T2A生成，此过程中音频分四步逐块生成。尽管采用了这种因果式少步生成方式，LiveAudio 的表现仍可与离线

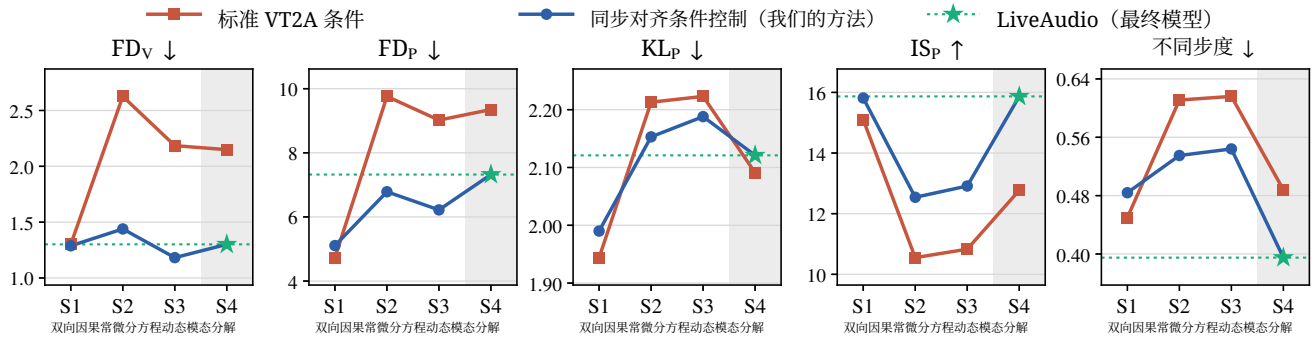

图 3. 在 VGGSound 测试集上对视频条件进行消融。比较了四个训练阶段后同步对齐条件控制与标准 VT2A 条件控制的效果。

表 3. AudioCaps 测试集上的文本到音频结果。LiveAu- dio 与表1中的模型相同，仅使用文本输入。

方法	FD_V↓	FD_P↓	KL_P↓	IS_P↑	感知质量↑	CLAP↑
离线						
AudioLDM	4.87	35.88	2.10	6.60	5.68	0.238
Tango	4.00	29.75	1.61	5.08	5.74	0.330
AudioLDM 2	2.41	33.73	1.59	7.26	5.87	0.421
Stable Audio Open	3.66	38.02	2.24	9.48	6.23	0.289
TangoFlux	2.21	21.66	1.25	11.92	5.89	0.544
MMAudio	3.32	18.33	1.44	11.52	5.61	0.489
AudioX	2.29	18.70	1.51	10.76	5.79	0.476
流式传输						
AudioGen	2.11	17.51	1.54	9.34	5.28	0.455
WorldSonus	2.46	20.65	1.66	8.63	5.81	0.442
LiveAudio（我们的方法）	2.02	19.12	1.65	10.76		0.460

文本到音频模型相当，其 FD 最低$_V$ 在所有模型中表现最佳，所有指标均优于流式传输模型 WorldSonus，并且具有与自回归模型 AudioGen 相当的竞争力。更多 T2A 结果见附录D.2。

4.3. 消融研究与分析

渐进式训练的消融。我们在表 4 中分析渐进式训练，其中 (a) 报告四个阶段各自结束后的结果，(b) 报告在相同训练预算下跳过阶段 4 之前各阶段的变体。在 (a) 中，块因果微调使模型具备因果性，但降低了音频质量和同步性。ODE 蒸馏将采样步数减少至四步且无需引导，同时保留了块因果 DiT 的质量，但同步性没有改善。随后，非对称DMD将不同步度从 0.544 降至 0.395，因此，在流式架构中，因果性造成的同步性损失不仅得到了恢复，还超过了双向 DiT 的 0.484。与阶段 3 相比，阶段 4 略微提高了 FD_V和 FD_P ，这与 DMD 的寻模行为一致 [76, 77]。与先前工作相同 [59, 71], , 其对抗损失主要提升音频

表 4. 在 VGGSound 测试集上对渐进式训练进行的消融研究。（a）每个阶段之后。（b）跳过部分阶段。

阶段	FD_V↓	FD_P↓	KL_P↓	IS_P↑	感知质量↑	不同步度↓
(a) 每个阶段之后						
阶段 1（双向）1.29		5.11	1.99	15.81	6.26	0.484
阶段 2（块因果）1.44		6.79	2.15	12.54	6.24	0.535
阶段 3（常微分方程）	1.18	6.22	2.19	12.91	6.21	0.544
阶段 4（DMD）	1.30	7.32	2.12	1		0.395
（*b*）跳过阶段						
$1 \to 4$	7.89	35.68	2.60	5.90	5.95	0.621
$1 \to 2 \to 4$	1.87	9.76	2.18	13.94	6.37	0.425
$1 \to 3 \to 4$	2.46	11.05	2.14	12.79	6.21	0.423
$1 \to 2 \to 3 \to 4$（我们的方法）	**1.30**	**7.32**	**2.12**	**1**		**0.395**

质量（附录E.3）。在 (b) 中，直接对双向 DiT 应用阶段 4 无法得到可用的生成器。跳过阶段 3（$1 \to 2 \to 4$）则是将阶段 4 直接应用于块因果 DiT；跳过阶段 2（$1 \to 3 \to 4$）则遵循 CausVid [78]和 Self Forcing [28], , 通过对双向 DiT 的轨迹进行 ODE 回归来初始化生成器。这两种方案在所有指标上都不如完整流程，因此每个阶段都是必不可少的。

视频条件控制对比。我们在图 [7, 11]中比较了同步对齐视频条件控制与标准 VT2A 条件控制3，其中两个模型均通过相同的四个阶段、采用完全一致的设置进行训练。二者的区别在于前瞻。标准条件控制为每个音频词元提供整个视频片段的特征，其中包含该词元之后的帧；而同步对齐条件控制仅使用已经播放的帧。在双向阶段，整个视频均可用，因此在音频质量相当的情况下，前瞻使标准条件控制具有略低的不同比度。在流式生成中，然而，当前块之后的帧尚未到达，因此使用标准条件控制训练的模型失去了其所依赖的前瞻。从阶段 2 开始，它在 FD、

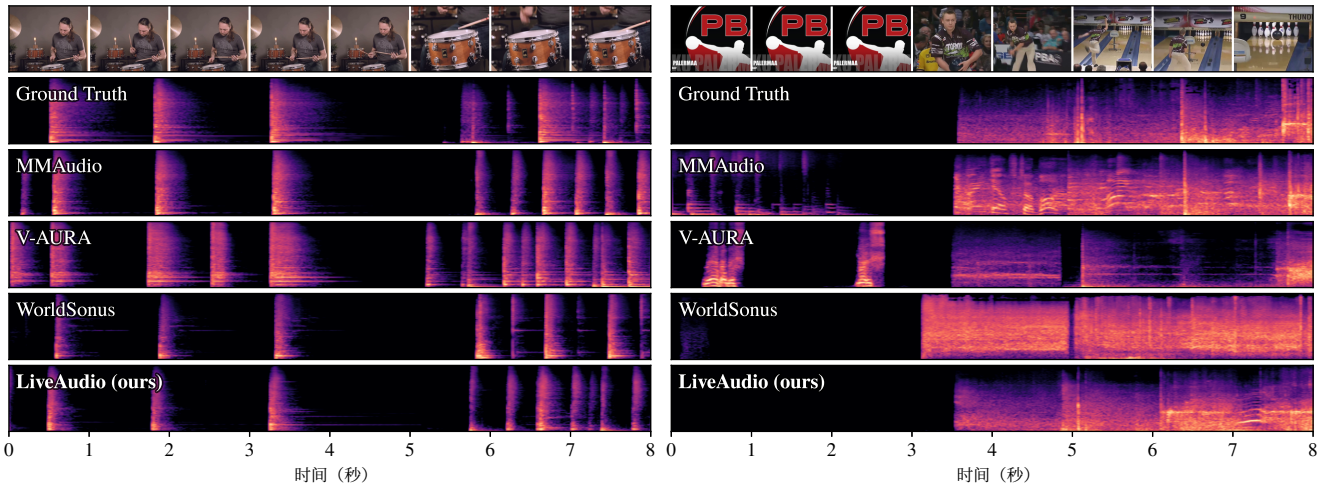

图4. 两个测试片段上的定性比较，以梅尔频谱图展示，视频帧置于同一时间轴上。LiveAudio的声音事件最接近真实值，而其他模型常常生成真实值中并不存在的声音。

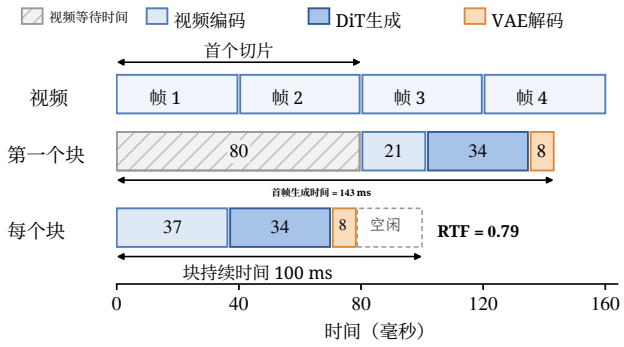

图5. LiveAudio以毫秒计的延迟分解，分别针对第一个块和每个块，其中DiT生成与VAE解码为所有块上的平均值。

条件控制方面，在 FD_V、FD_P、IS_P 和不同步度方面明显落后于同步对齐条件控制；经过非对称 DMD 后，其不同步度为 0.488，而后者为 0.395。这些结果表明，流式生成器从双向阶段开始就需要一种不带前瞻的视频条件控制。

效率分析。 图5 对LiveAudio的延迟进行了分解。对于第一个块，首帧生成时间的一半以上用于等待第一个视频切片，其余部分由视频编码、生成和解码分担。在稳态下，一个块平均跨越1.25个视频切片，并承担其相应份额的CLIP编码，因此其视频编码耗时长于第一个块。每个块仍在其持续时间内完成处理，因此LiveAudio的运行速度快于实时。当帧实时到达时，0.23秒的常数播放延迟便足以实现不间断播放。

更多研究。 受篇幅限制，我们在附录中提供了更多研究 E。我们对采样步数进行了消融，报告了视频条件对比的数值结果，并考察了非对称设计 DMD，即其对抗损失、双向教师模型以及对自生成历史的使用。我们还评估了长篇生成能力：LiveAudio 无需额外训练，即可生成超出训练长度的内容。

4.4. 案例研究

图 4 比较了两个测试片段生成的音频。在第一个片段中，LiveAudio 将小军鼓的三次独立敲击安排在与真实值相同的时间点，并跟随滚奏；而 V-AURA 则产生了额外的敲击声。在第二个片段中，视频以标题卡开场，此时真实值中没有声音。LiveAudio 在这段时间内保持静音，并在场景出现时开始播放；而 MMAudio 和 V-AURA 在标题卡期间就生成了声音，World-Sonus 则过早开始播放。更多案例见附录 F，音频样本可在演示页面上获取。

5. 结论

我们提出了 LiveAudio，这是一种将双向 V2A模型转换为自回归生成器的框架，用于实时流式V2A生成。LiveAudio 将这一转换解耦为因果性和效率两个方面。同步对齐的视频条件和块因果微调使模型具备因果性，而 ODE 蒸馏则在无需引导的情况下将采样减少到四步。我们表明，非对称DMD能够恢复模型变为因果模型后丢失的同步性，并将其提升到超过双向模型的水平。在 VGGSound 测试集上，LiveAudio 在音频质量和同步性方面优于现有流式模型，在同步性方面超过离线模型，并且在单个GPU上实现快于实时的运行速度。同时，同一模型还支持具有竞争力性能的流式VT2A和 T2A生成。

大语言模型使用声明

我们以以下方式使用了基于大语言模型的助手 Claude Code（Anthropic）和 Codex（OpenAI）。写作：协助起草和修改所有章节的文本，并搜索和总结相关工作。图表：协助编写绘图脚本；没有任何图表包含 AI 生成的图像。代码和实验：协助编写和审查训练、推理和评估代码，并启动和监控实验。构思：研究问题、方法以及要开展的实验均由作者决定，助手则提出了实现和评估方面的细节。作者审阅了所有文本、参考文献、代码和结果，并对论文内容承担全部责任。

参考文献

[1] Yatong Bai、Trung Dang、Dung Tran、Kazuhito Koishida 和 Somayeh Sojoudi。一致性tta：通过一致性蒸馏加速基于扩散的文本到音频生成。arXiv 预印本 arXiv:2309.10740，2023年。2[2]Jake Bruce、Michael D Dennis、Ashley Edwards、Jack Parker-Holder、Yuge Shi、Edward Hughes、Matthew Lai、Aditi Mavalankar、Richie Steigerwald、Chris Apps 等。Genie：生成式交互环境。载于第四十一届国际机器学习会议，2024年。1[3] Haoxuan Che、Xuanhua He、Quande Liu、Cheng Jin 和 Hao Chen。GameGen-X：交互式开放世界游戏视频生成。载于国际学习表征会议，第 37546–37593 页，2025年。1[4] Boyuan Chen、Diego Mart´ı Mons´o、Yilun Du、Max Sim- chowitz、Russ Tedrake 和 Vincent Sitzmann。Diffusion Forcing：下一词元预测遇上全序列扩散。神经信息处理系统进展，37:24081–24125，2024年。2[5] Honglie Chen、Weidi Xie、Andrea Vedaldi 和 Andrew Zisserman。VGGSound：大规模音视数据集。载于 ICASSP 2020-2020 IEEE 声学、语音与信号处理国际会议（ICASSP），第 721–725 页。IEEE，2020年。2、5、14[6] Zehua Chen、Junyou Wang、Yuxuan Jiang、Zhenying Fang、Yusheng Dai、Jianfei Chen、Ziwei Liu 和 Jun Zhu。视觉表征至关重要：利用视频到音频生成中的时间差异。arXiv 预印本 arXiv:2608.04902，2026年。2[7] Ho Kei Cheng、Masato Ishii、Akio Hayakawa、Takashi Shibuya、Alexander Schwing 和 Yuki Mitsufuji。MMAudio：驾驭多模态联合训练，实现高质量视频到音频合成。载于 2025 IEEE/CVF 计算机视觉与模式识别会议（CVPR），第 28901–28911 页。IEEE，2025年。1、2、3、4、6、7、14、15、17、19[8] Hyung Won Chung、Le Hou、Shayne Longpre、Barret Zoph、Yi Tay、William Fedus、Yunxuan Li、Xuezhi Wang、Mostafa Dehghani、Siddhartha Brahma 等。扩展指令微调语言模型。机器学习研究杂志，25(70):1–53，2024年。6、17[9] Aurora Linh Cramer、Ho-Hsiang Wu、Justin Salamon 和 Juan Pablo Bello。看、听并学习更多：设计深度音频嵌入的选择。在 ICASSP 2019-2019 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 3852–3856 页。IEEE，2019年。6, 14[10] Yusheng Dai、Chenxi Wang、Chang Li、Chen Wang、Kewei Li、Jun Du、Lei Sun、Jianqing Gao、Ruoyu Wang 和 Jiefeng Ma。用于二维长篇潜在生成的潜在交换联合扩散。在 2025 IEEE/CVF计算机视觉国际会议（ICCV）中，第 11006–11015 页。IEEE，2025年。21[11] Yusheng Dai、Zehua Chen、Yuxuan Jiang、Baolong Gao、Qiuhong Ke、Jun Zhu 和 Jianfei Cai。Omni2sound：迈向统一的视频文本到音频生成。arXiv 预印本 arXiv:2601.02731，2026年。1, 2, 4, 5, 6, 7, 14, 19, 20[12] Yusheng Dai、Vladimir Iashin、Kangdi Wang、Yuxuan Jiang、Ilpo Viertola、Weiqiang Wang、Qiuhong Ke、Min Zhao 和 Jianfei Cai。LiveFoley：面向沉浸式交互世界的动态实时视频到音频生成。arXiv 预印本 arXiv:2611.05988，2026年。1, 2, 4, 6, 17, 19, 20[13] Yusheng Dai、Kangdi Wang、Baolong Gao、Yuxuan Jiang、Weiqiang Wang、Qiuhong Ke 和 Jianfei Cai。Cine-dub：通过连贯音效将端到端视频配音扩展至多说话人对话。arXiv 预印本 arXiv:2608.15734，2026年。2[14] Konstantinos Drossos、Samuel Lipping 和 Tuomas Virtanen。Clotho：音频描述数据集。在 ICASSP 2020-2020 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 736–740 页。IEEE，2020年。19[15] Patrick Esser、Sumith Kulal、Andreas Blattmann、Rahim Entezari、Jonas Müller、Harry Saini、Yam Levi、Dominik Lorenz、Axel Sauer、Frederic Boesel 等。扩展用于高分辨率图像合成的修正流变换器。在第四十一届国际机器学习会议中，2024年。18¨Zach Evans、CJ Carr、Josiah Taylor、Scott H Hawley 和 Jordi Pons。快速时间条件潜在音频扩散。在第四十一届国际机器学习会议中，2024年。21[16] Zach Evans、Julian D Parker、CJ Carr、Zack Zukowski、Josiah Taylor 和 Jordi Pons。使用潜在扩散进行长篇音乐生成。arXiv 预印本 arXiv:2404.10301，2024年。21[17] Zach Evans、Julian D Parker、CJ Carr、Zack Zukowski、J osiah Taylor 和 Jordi Pons。Stable Audio Open。在 ICASSP 2025-2025 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 1–5 页。IEEE，2025年。4, 6, 14, 15, 16, 17, 18, 19[18] Pengjun Fang、Jingyi Fa、Kam Man Wu、Jiaming Wang、Haoyuan Huang、Yaguang Wu、Xiangjun Huang、Ziyang Ma、Weijia Chen、Hongyu Liu 等。WorldSonus：为世界带来声音。arXiv 预印本 arXiv:2610.08760，2026年。1, 2, 6, 15, 19, 21[19] Jort F Gemmeke、Daniel PW Ellis、Dylan Freedman、Aren Jansen、Wade Lawrence、R Channing Moore、Manoj Plakal 和 Marvin Ritter。Audio Set：用于音频事件的本体与人工标注数据集。在 2017 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 776–780 页。IEEE，2017年。5, 14, 15

[21] Zhengyang Geng、Mingyang Deng、Xingjian Bai、Zico Kolter 和 何恺明。用于单步生成建模的均值流。《神经信息处理系统进展》，38:75460–75482，2026年。2[22] Deepanway Ghosal、Navonil Majumder、Ambuj Mehrish 和 Soujanya Poria。使用指令引导的潜在扩散模型进行文本到音频生成。在第 31 届 ACM 多媒体国际会议论文集中，第 3590–3598 页，2023年。6, 19[23] Rohit Girdhar、Alaaeldin El-Nouby、Zhuang Liu、Mannat Singh、Kalyan Vasudev Alwala、Armand Joulin 和 Ishan Misra。ImageBind：在一个嵌入空间中绑定所有模态。在 2023 IEEE/CVF计算机视觉与模式识别会议（CVPR）中，第 15180–15190 页。IEEE，2023年。6, 15[24]Ga¨etan Hadjeres、Marc Ferras、Khaled Koutini、Benno Weck、Alexandre Bittar、Thomas Hummel、Zineb Lahrichi、Hakim Missoum、Joan Serr`a 和 Yuki Mitsufuji。Woosh：音效基础模型。arXiv 预印本 arXiv:2604.01929，2026年。2[25] Xianglong He、Chunli Peng、Zexiang Liu、Boyang Wang、Yifan Zhang、Qi Cui、Fei Kang、Biao Jiang、Mengyin An、Yangyang Ren 等。Matrix-game 2.0：开源、实时且支持流式传输的交互式世界模型。arXiv 预印本 arXiv:2508.13009，2025年。1[26] Shawn Hershey、Sourish Chaudhuri、Daniel PW Ellis、Jort F Gemmeke、Aren Jansen、R Channing Moore、Manoj Plakal、Devin Platt、Rif A Saurous、Bryan Seybold 等。用于大规模音频分类的 CNN 架构。在 2017 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 131–135 页。IEEE，2017年。6, 14[27] Jonathan Ho 和 Tim Salimans。无分类器扩散引导。arXiv 预印本 arXiv:2207.12598，2022年。2, 3, 18[28] 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》，38:167283–167308，2026年。2, 4, 5, 7, 18, 21[29] Yubo Huang、Hailong Guo、Fangtai Wu、Weiqiang Wang、Shijie Huang、Qijun Gan、Shifeng Zhang、Lin Liu、Sirui Zhao、Enhong Chen 等。实时虚拟形象：支持无限时长的实时音频驱动虚拟形象生成。在欧洲计算机视觉会议中，第 76–93 页。施普林格，2026年。1, 2[30] Chia-Yu Hung、Navonil Majumder、Zhifeng Kong、Ambuj Mehrish、Amir Ali Bagherzadeh、Chuan Li、Rafael Valle、Bryan Catanzaro 和 Soujanya Poria。TangoFlux：通过流匹配和 CLAP 排序偏好优化实现超快速且忠实的文本到音频生成。arXiv 预印本 arXiv:2412.21037，2024年。3, 6, 19[31] Vladimir Iashin、Weidi Xie、Esa Rahtu 和 Andrew Zisserman。Synchformer：从稀疏线索中实现高效同步。在 ICASSP 2024-2024 IEEE 声学、语音与信号处理国际会议（ICASSP）中，第 5325–5329 页。IEEE，2024年。2, 4, 6, 15, 17[32] Yujin Jeong、Yunji Kim、Sanghyuk Chun 和 Jiyoung Lee。阅读、观看并尖叫！从文本生成声音以及视频。在《AAAI人工智能会议》论文集，第17590–17598页，2025年。2[33] Yuxuan Jiang、Zehua Chen、Zeqian Ju、Chang Li、Weibei Dou和Jun Zhu。Freeaudio：无需训练的可控长篇文本到音频生成时间规划。arXiv预印本 arXiv:2507.08557，2025年。21[34] Yuxuan Jiang、Zehua Chen、Zeqian Ju、Yusheng Dai、Weibei Dou和Jun Zhu。Controlaudio：通过渐进式扩散建模应对文本引导、时间指示和可理解的音频生成。在《第64届计算语言学协会年会》（第1卷：长论文）论文集，第1394–1413页，2026年。4[35] Yuxuan Jiang、Mingyang Han、Yusheng Dai、Andong Wang、Tianhong Zhou、Jiaxin Ye、Dongxiao Wang、Haoxiang Shi、Boyu Li、Jun Song等。Freesonic：无需训练的时间感知解耦注意力，用于精确音频编辑。arXiv预印本 arXiv:2606.15186，2026年。3[36] Lei Ke、Jiahao Pan、Zeyue Tian、Jiaming Wang、Haoyuan Huang、Kam Man Wu、Pengjun Fang、Hongyu Liu、Chenyang Qi、Lin Wang等。Helixworld：实时交互式音频视觉世界模型。arXiv预印本 arXiv:2609.38123，2026年。2[37] Taekyung Ki、Sangwon Jang、Jaehyeong Jo、Jaehong Yoon和Sung Ju Hwang。Avatar Forcing：面向自然对话的实时交互式头部虚拟形象生成。在《IEEE/CVF计算机视觉与模式识别会议》论文集，第18074–18084页，2026年。2[38] Kevin Kilgour、Mauricio Zuluaga、Dominik Roblek和Matthew Sharifi。Fr´échet音频距离：一种评估音乐增强算法的指标。arXiv预印本 arXiv:1812.08466，2018年。14[39] Chris Dongjoo Kim、Byeongchang Kim、Hyunmin Lee和Gunhee Kim。AudioCaps：为野外音频生成字幕。在《2019年计算语言学协会北美分会会议：人类语言技术》论文集，第1卷（长论文与短论文），第119–132页，2019年。5、6、14[40] Qiuqiang Kong、Yin Cao、Turab Iqbal、Yuxuan Wang、Wenwu Wang和Mark D Plumbley。PANNs：用于音频模式识别的大规模预训练音频神经网络。IEEE/ACM《音频、语音与语言处理汇刊》，28：2880–2894，2020年。6、14、15[41] Khaled Koutini、Jan Schl¨uter、Hamid Eghbal-Zadeh和Gerhard Widmer。利用Patchout高效训练音频Transformer。arXiv预印本 arXiv:2110.05069，2021年。6、15[42] Felix Kreuk、Gabriel Synnaeve、Adam Polyak、Uriel Singer、Alexandre D´efossez、Jade Copet、Devi Parikh、Yaniv Taigman和Yossi Adi。AudioGen：文本引导的音频生成。arXiv预印本 arXiv:2209.15352，2022年。6、19、21[43] Ke Lei、Yu Zhang、Changhao Pan、Xueyi Pu、Wenxiang Guo、Ruiqi Li和Zhou Zhao。通过自回归扩散Transformer实现流式同步空间音频生成。arXiv预印本 arXiv:2605.30940，2026年。1、2[44] Xiquan Li、Junxi Liu、Yuzhe Liang、Zhikang Niu、Wenxi Chen和Xie Chen。MeanAudio：利用均值流实现快速且保真的文本到音频生成。在《

第64届计算语言学协会年会（第1卷 14393，2026 14378 2：长论文），第–页。[45] Yinghao Aaron Li、Rithesh Kumar和Zeyu Jin。DMD-Speech：通过直接指标优化实现超越教师模型的零样本语音合成蒸馏扩散模型。2024年。2[46] Yaron Lipman、Ricky TQ Chen、Heli Ben-Hamu、Maximilian Nickel和Matt Le。生成建模的流匹配。arXiv预印本 arXiv:2210.02747，2022年。3[47] Dongyang Liu、Gao Peng、David Liu、DU DU、Zhen Li、Qilong Wu、Xin Jin、Sihan Cao、Shifeng Zhang、Steven HOI等。解耦DMD：以CFG增强为矛、分布匹配为盾。在《国际学习表征会议》，第140643–140666页，2026年。2[48] Haohe Liu、Zehua Chen、Yi Yuan、Xinhao Mei、Xubo Liu、Danilo Mandic、Wenwu Wang和Mark D Plumbley。AudioLDM：利用潜在扩散模型实现文本到音频生成。arXiv预印本 arXiv:2301.12503，2023年。4、6、14、15、19、21[49] Huadai Liu、Rongjie Huang、Yang Liu、Hengyuan Cao、Jialei Wang、Xize Cheng、Siqi Zheng和Zhou Zhao。AudioLCM：利用潜在一致性模型实现文本到音频生成。arXiv预印本 arXiv:2406.00356，2024年。2[50] Haohe Liu、Yi Yuan、Xubo Liu、Xinhao Mei、Qiuqiang Kong、Qiao Tian、Yuping Wang、Wenwu Wang、Yuxuan Wang和Mark D Plumbley。AudioLDM 2：通过自监督预训练学习整体音频生成。IEEE/ACM《音频、语音与语言处理汇刊》，2024年。6、19、21[51] Huadai Liu、Kaicheng Luo、Jialei Wang、Wen Wang、Qian Chen、Zhou Zhao和Wei Xue。ThinkSound：多模态大型语言模型中用于音频生成与编辑的思维链推理。arXiv预印本 arXiv:2506.21448，2025年。1、2、19[52] Huadai Liu、Jialei Wang、Rongjie Huang、Yang Liu、Heng Lu、Zhou Zhao和Wei Xue。FlashAudio：用于快速高保真文本到音频生成的校正流。在《第63届计算语言学协会年会》（第1卷：长论文）论文集，第13694–13710页，2025年。2[53] Xingchao Liu、Chengyue Gong和Qiang Liu。Flow straight and fast：利用校正流学习生成和迁移数据。arXiv预印本 arXiv:2209.03003，2022年。2、3[54] Ilya Loshchilov和Frank Hutter。解耦权重衰减正则化。arXiv预印本 arXiv:1711.05101，2017年。17[55] Cheng Lu、Yuhao Zhou、Fan Bao、Jianfei Chen、Chongxuan Li和Jun Zhu。DPM-Solver++：扩散概率模型引导采样的快速求解器。《机器智能研究》，22(4)：730–751，2025年。2[56] Simian Luo、Chuanhao Yan、Chenxu Hu和Hang Zhao。Diff-Foley：利用潜在扩散模型实现同步视频到音频合成。《神经信息处理系统进展》，36：48855–48876，2023年。2[57] Xinhao Mei、Chutong Meng、Haohe Liu、Qiuqiang Kong、Tom Ko、Chengqi Zhao、Mark D Plumbley、Yuexian Zou和Wenwu Wang。WavCaps：由ChatGPT辅助的弱监督……

用于音频-语言多模态研究的带标注音频描述数据集。IEEE/ACM《音频、语音与语言处理汇刊》，2024年。5，14 [58] Zachary Novack、Zach Evans、Zack Zukowski、Josiah Tay-lor、CJ Carr、Julian Parker、Adnan Al-Sinan、Gian Marco Iodice、Julian McAuley、Taylor Berg-Kirkpatrick 等。基于对抗式后训练的快速文本到音频生成。载于 2025 IEEE音频与声学信号处理应用研讨会（WASPAA），第 1–5 页。IEEE，2025年。3[59] Zachary Novack、Ge Zhu、Jonah Casebeer、Julian McAuley、Taylor Berg-Kirkpatrick 和 Nicholas J Bryan。Presto! 蒸馏步数与层以加速音乐生成。载于国际学习表征会议，第 50924–50948 页，2025年。2、5、7、18、21[60] Zachary Novack、Stephen Brade、Haven Kim、Hugo Flo-res Garc´ıa、Nithya Shikarpur、Chinmay Talegaonkar、Suwan Kim、Valerie K Chen、Julian McAuley、Taylor Berg-Kirkpatrick 等。实时音乐扩散模型：交互式扩散音乐生成器的高效微调与后训练。arXiv 预印本，arXiv:2605.22717，2026年。3[61] William Peebles 和 Saining Xie。基于变换器的可扩展扩散模型。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 4195–4205 页，2023年。2、4 – 。[62]Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark 等。通过自然语言监督学习可迁移视觉模型。载于国际机器学习会议，第 8748–8763 页。PmLR，2021年。4、17[63]Koichi Saito、Julian Tanke、Christian Simon、Masato Ishii、Kazuki Shimada、Zachary Novack、Zhi Zhong、Akio Hayakawa、Takashi Shibuya 和 Yuki Mitsufuji。Soundreac-tor：帧级在线视频到音频生成。arXiv 预印本，arXiv:2510.02110，2025年。1、2、4、6、16、19[64] Sizhe Shan、Qiulin Li、Yutao Cui、Miles Yang、Yuehai Wang、Qun Yang、Jin Zhou 和 Zhao Zhong。Hunyuanvideo-foley：通过表征对齐实现高保真拟音音频生成的多模态扩散。arXiv 预印本，arXiv:2508.16930，2025年。1、2、19[65] Joonghyuk Shin、Zhengqi Li、Richard Zhang、Jun-Yan Zhu、Jaesik Park、Eli Shechtman 和 Xun Huang。Motionstream：具有交互式运动控制的实时视频生成。载于国际学习表征会议，第 7381–7406 页，2026年。2[66] Jiaming Song、Chenlin Meng 和 Stefano Ermon。去噪扩散隐式模型。arXiv 预印本，arXiv:2010.02502，2020年。2[67] Yang Song、Prafulla Dhariwal、Mark Chen 和 Ilya Sutskever。一致性模型。arXiv 预印本，arXiv:2303.01469，2023年。2[68] Yaofeng Su、Yuming Li、Zeyue Xue、Jie Huang、Siming Fu、Haoran Li、Haoyang Huang 和 Nan Duan。O mniforc-ing：释放实时音视频联合生成能力。载于欧洲计算机视觉会议，第 566–584 页。施普普林格，2026年。2

[69] Zhiyao Sun、Ziqiao Peng、Yifeng Ma、Yi Chen、Zhengguang Zhou、Zixiang Zhou、Guozhen Zhang、Youliang Zhang、Yuan Zhou、Qinglin Lu 等。Streamavatar：用于实时交互式人类头像的流式扩散模型。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 10887–10897 页，2026年。1、2[70] Zeyue Tian、Yizhu Jin、Zhaoyang Liu、Ruibin Yuan、Xu Tan、Qifeng Chen、Wei Xue 和 Yike Guo。Audiox：用于任意内容到音频生成的扩散 Transformer。arXiv 电子预印本，第 arXiv–2503 页，2025年。6、19[71] Zeyue Tian、Lei Ke、Zhaoyang Liu、Ruibin Yuan、Liu-meng Xue、Yujiu Yang、Weijia Chen、Xu Tan、Qifeng Chen、Wei Xue 等。Audiox-Turbo：用于高效任意内容到音频生成的统一框架。arXiv 预印本，arXiv:2606.12555，2026年。2、7、21[72] Andros Tjandra、Yi-Chiao Wu、Baishan Guo、John Hoffman、Brian Ellis、Apoorv Vyas、Bowen Shi、Sanyuan Chen、Matt Le、Nick Zacharov 等。Meta Audiobox Aesthetics：面向语音、音乐和声音的统一自动质量评估。arXiv 预印本，arXiv:2502.05139，2025年。6、15[73] Ilpo Viertola、Vladimir Iashin 和 Esa Rahtu。通过自回归生成与视频时间对齐的音频。载于 ICASSP 2025-2025 IEEE 声学、语音与信号处理国际会议（ICASSP），第 1–5 页。IEEE，2025年。1、2、6、15、19[74] Yongqi Wang、Wenxiang Guo、Rongjie Huang、Jiawei Huang、Zehan Wang、Fuming You、Ruiqi Li 和 Zhou Zhao。FRIEREN：基于修正流匹配的高效视频到音频生成网络。神经信息处理系统进展，37：128118–128138，2024年。6[75] Yusong Wu、Ke Chen、Tianyu Zhang、Yuchen Hui、Taylor Berg-Kirkpatrick 和 Shlomo Dubnov。通过特征融合与关键词到字幕增强实现大规模对比式语言-音频预训练。载于 ICASSP 2023-2023 IEEE 声学、语音与信号处理国际会议（ICASSP），第 1–5 页。IEEE，2023年。6、15[76] Tianwei Yin、Micha¨el Gharbi、Taesung Park、Richard Zhang、Eli Shechtman、Fredo Durand 和 William T Freeman。改进的分布匹配蒸馏，用于快速图像合成。神经信息处理系统进展，37：47455–47487，2024年。2、3、5、7[77] Tianwei Yin、Micha¨el Gharbi、Richard Zhang、Eli Shecht-man、Fredo Durand、William T Freeman 和 Taesung Park。通过分布匹配蒸馏实现一步扩散。载于 2024 IEEE/CVF计算机视觉与模式识别会议（CVPR），第 6613–6623 页。IEEE，2024年。2、3、7、18[78] Tianwei Yin、Qiang Zhang、Richard Zhang、William T Free-man、Fredo Durand、Eli Shechtman 和 Xun Huang。从缓慢的双向视频扩散模型到快速的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），第 22963–22974 页。IEEE，2025年。2、3、4、5、7[79] Jintao Zhang、Kai Jiang、Jintao Chen、Xu Wang、Yang Luo、Yuji Wang、Dechuang Chen、Jungang Li、Chengyang Ye、Marco Chen 等。Vidu S1：实时交互式视频生成模型。arXiv 预印本，arXiv:2607.03118，2026年。2

[80] Yiming Zhang、Yicheng Gu、Yanhong Zeng、Zhening Xing、Yuancheng Wang、Zhizheng Wu、Bin Liu 和 Kai Chen。Foleycrafter：让无声视频通过逼真且同步的声音焕发生命力。*International Journal of ComputerVision*，134(1):46，2026年。2[81] Hongzhou Zhu、Min Zhao、Guande He、Hang Su、Chongxuan Li 和 Jun Zhu。Causal Forcing：用于高质量实时交互式视频生成的自回归扩散蒸馏。*arXiv* 预印本 *arXiv:2602 02214*，2026年2，3，4，5，18。。

目录

1 引言 **1**

2 相关工作 **2**
2.1 视频到音频生成 .22.2 流式多模态生成 .22.3 少步扩散蒸馏 .2

3 LiveAudio **3**
3.1 问题设定 .33.2 自回归扩散架构 .43.3 渐进式离线到在线训练 .43.4 分块式 KV 缓存推理 .5

4 实验 **5**
4.1 实验设置 .54.2 主要结果 .64.3 消融研究与分析 .74.4 案例研究 .8

5 结论 **8**

A 数据集详情 **14**
A.1 训练数据 .14A.2 评估数据 .14

B 评估指标 14
B.1 分布匹配 .14B.2 音频质量 .15B.3 模态对齐 .15B.4 效率.15B.5 主观评估 .16

C 实现细节 **16**
C.1 模型架构 .16C.2 训练细节 .17C.3 推理细节 .18

D 补充结果 **19**
D.1 视频文本到音频生成 . 19
D.2 文本到音频生成 . 19

E 附加消融研究与分析 **20**
E.1 采样步数.20E.2 视频条件.20E.3 非对称DMD.20E.4 长篇音频生成.21

F 案例研究 **22**

A. 数据集详情

我们在同一语料库上训练全部四个阶段。视频到音频（V2A）和视频文本到音频（VT2A）任务在SoundAtlas上进行训练 [11], , 而文本到音频（T2A）任务则在AudioCaps上进行训练 [39]以及WavCaps上 [57]。我们在VGGSound测试集上评估V2A和VT2A [5], , 并在AudioCaps测试集上评估T2A。

A.1. 训练数据

SoundAtlas。 该数据集来自 Omni2Sound [11] ，提供了来自 VG-GSound [5] 和 AudioSet [20], 两个10秒 YouTube 视频集合中片段的 47 万个视频 - 音频 - 文本三元组，这些视频均标注了声音类别。每个片段的字幕描述其声音，由一种同时利用片段音频和视频的智能标注流程生成。VT2A训练使用带字幕的片段，而 V2A 训练使用所有去除字幕的片段。

AudioCaps。 该音频描述数据集 [39] 将来自 AudioSet 的10秒片段与通过众包由人工标注者撰写的字幕配对。每个训练片段有一条字幕，而每个验证片段或测试片段有五条字幕。其测试集是 T2A生成 [18, 48]的标准基准。我们使用其中的5万个训练片段进行 T2A 训练。

WavCaps。 该数据集 [57] 由从 FreeSound、BBC S ound Effects、SoundBible 和 AudioSet 收集的40万个音频片段组成，其字幕由 ChatGPT 根据片段的原始描述生成。我们也将其用于 T2A 训练，并将其中的长录音划分为不超过30秒的片段。由于发布的音频被下采样至32千赫，我们遵循 MMAudio [7] 的做法，将来自 BBC Sound Effects 的片段替换为其原始录音；这些原始录音的采样率为44.1或48千赫。

过滤与采样。 我们会舍弃短于 1 秒的片段和分段，并移除所有与 AudioCaps 的验证集或测试集，或 MMAudio 使用的 VGGSound 验证集或测试集共享源视频的训练片段 [7]。每个训练批次仅包含一个任务，且四个阶段均使用相同的任务混合比例。遵循 Omni2Sound [11], , VT2A、V2A 和 T2A 批次的采样概率分别为 0.35、0.55 和 0.10（图 6）。所有样本的时长均为 10 秒。视频片段从其开头开始，较长的文本-音频分段从随机位置裁剪，而较短的片段则使用掩码进行填充。

A.2. 评估数据

VGGSound 测试集。 V2A 和 VT2A 均在 VGGSound 的测试划分上进行评估 [5], , 其中每个片段最长为 10 秒。该数据集经过收集，使得声音来源在视频中清晰可见，并且每个片段都被标注

每个声音类别对应一种声音。对于 VT2A，文本提示是

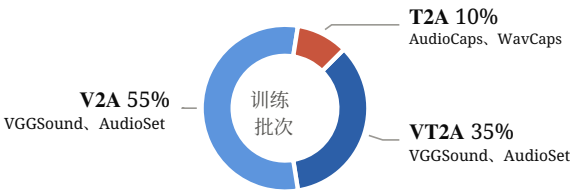

图 6. 训练批次中三项任务的概率，这些训练批次遵循 Omni2Sound [11]。

VGGSound-Omni 的字幕。VGGSound-Omni 是 Omni2Sound [11] 中引入的基准，用于描述每个测试片段的声音。

AudioCaps 测试集。 T2A 在 AudioCaps [39] 中仍可用的测试片段上进行评估。每个片段时长为 10 秒，并配有五条人工撰写的描述，其中一条用作文本提示。

B. 评估指标

B.1. 分布匹配

FD_O。 该指标使用 Stable Audio Open [18]的评估代码计算，用于比较生成音频与真实音频在 OpenL3 [9]嵌入空间中的分布。所有音频均进行峰值归一化至 −1dBFS，并重采样至 44.1 千赫；单声道音频则复制到两个声道。OpenL3 的环境声音版本每 0.5 秒从每个声道提取一个 512 维嵌入，并将两个声道的嵌入进行拼接。FD_O 是对所有生成音频和所有真实音频的嵌入分别拟合两个高斯分布后计算得到的 Fr´echet 距离。由于该指标在 44.1 千赫下计算，FD_O 还反映了 8 千赫以上的频率，而在 16 千赫下计算的指标会去除这些频率。

FD_V。 该指标比较生成音频与真实音频在 VG-Gish 模型 [26], 的嵌入空间中的分布，并使用 AudioLDM [48]的评估代码进行计算。它也被称为 Fr´echet 音频距离（FAD） [38]。所有音频均重采样至 16 千赫，VGGish 每 0.96 秒音频提取一个 128 维嵌入。FD_V 是对所有生成音频和所有真实音频的嵌入分别拟合两个高斯分布后计算得到的 Fr´echet 距离。

FD_P。 该指标衡量生成音频与真实音频在 PANNs CNN14 分类器嵌入空间中的分布距离 [40], , 并使用 AudioLDM [48]的评估代码计算。所有音频均重采样至 16 千赫，每个片段都由 CNN14 的倒数第二层映射为一个 2048 维嵌入。FD_P 是对所有生成片段的嵌入和所有真实音频片段的嵌入分别拟合两个高斯分布后，计算得到的 Fr´echet 距离。

$\mathrm{KL_S}$。 使用 PaSST分类器 [41], 该指标衡量生成片段中的声音事件与真实音频中的声音事件的匹配程度，并使用 Stable Audio Open 的评估代码计算。所有音频均转换为 32 千赫单声道，并进行峰值归一化至 -1 dBFS。短于 10 秒的片段会进行零填充，以达到 PaSST 的 10 秒输入长度。PaSST 对 AudioSet 的 527 个 [20] 类别预测出的 logits 经 softmax 转换为类别分布。以 p_{gen} 和 p_{gt} 分别表示生成片段及其真实音频的分布，$\mathrm{KL_S}$ 为所有片段的 Kullback-Leibler（KL）散度 $\mathrm{KL}(p_{\mathrm{gen}} \,\|\, p_{\mathrm{gt}})$ 的平均值。

$\mathrm{KL_P}$。 使用 PANNs CNN14 分类器 [40], , 该指标衡量生成片段中的声音事件与其真实值中的声音事件的匹配程度，并使用 AudioLDM [48]的评估代码进行计算。所有音频均重采样至 16 千赫，CNN14 对整个片段在 527 个 AudioSet 类别上预测的 logits 通过 softmax 转换为类别分布。设 p_{gen} 和 p_{gt} 分别表示生成片段及其真实值的分布，则 $\mathrm{KL_P}$是所有片段上 $\mathrm{KL}(p_{\mathrm{gt}} \,\|\, p_{\mathrm{gen}})$ 的平均值，其中散度方向与 $\mathrm{KL_S}$相反。

B.2. 音频质量

$\mathrm{IS_P}$。 $\mathrm{IS_P}$ 是生成音频的初始分数，使用 PANNs CNN14 分类器 [40] 和 AudioLDM 的评估代码 [48] 获得，且不使用真实值。所有音频均重采样至 16 千赫，CNN14 对片段 x在 527 个 AudioSet 类别上预测的 logits 通过 softmax 转换为类别分布 $p(y \mid x)$。对于一组片段，该分数是 $\mathrm{KL}(p(y \mid x) \,\|\, p(y))$ 平均值的指数，其中 $p(y)$ 是该组中 $p(y \mid x)$ 的平均值。当每个片段都以较高置信度被分类，且这些片段覆盖多样的类别时，该分数较高。生成片段使用固定种子进行打乱，并划分为十个子集，$\mathrm{IS_P}$ 是这些子集分数的平均值。

PQ. PQ 是 Audiobox Aesthetics 预测的制作质量 [72], 。Audiobox Aesthetics 从制作质量、制作复杂度、内容愉悦度和内容实用性四个维度对音频进行评分，每项评分范围为 1 到 10。制作质量关注音频的技术层面，包括清晰度和保真度、动态范围、频率内容和空间化，而非其主观吸引力。我们使用官方预测器和检查点：它们会将音频转换为 16 千赫单声道，以 10 秒为窗口进行评分，再根据各窗口的时长加权平均窗口分数。PQ 是所有生成片段的平均分数。

B.3. 模态对齐

不同步度。 该不同步度分数用于衡量生成音频相对于视频的时间偏移程度，并使用 MMAudio [7] 的评估代码计算得出。视频的前 8 秒按每秒25帧提取，音频的前 8 秒则转换为 16 千赫单声道。Synchformer [31] 将视听偏移预测为 21 个类别之一，范围为 -2 至 2 秒，步长为 0.2 秒，并分别针对该区间的前 4.8 秒和后 4.8 秒进行预测。不同步度是在两个窗口及所有片段上取平均后的绝对偏移值，短于 8 秒的片段将被排除。

IB. IB 是 ImageBind 分数，用于衡量视频与生成音频之间的语义一致性。该分数通过 ImageBind [23], 计算，并使用 MMAudio [7]的评估代码。取视频前 8 秒中每秒 0.5 帧的四帧，视频嵌入则对三个空间裁剪结果以及三个由连续两帧组成的片段取平均。音频会转换为 16 千赫单声道，并截取前 8 秒；其嵌入则对音频中分布的三个 2 秒片段取平均。IB 是两个嵌入之间的余弦相似度，并在与不同步度相同的片段上取平均。

CLAP。 CLAP 用于衡量生成音频与字幕之间的一致性，计算时使用 Stable Audio Open [18] 的评估代码。生成的音频会转换为 48 kHz 单声道，并进行峰值归一化至-1 dBFS；LAION-CLAP 模型 [75] 采用特征融合，对音频和字幕进行嵌入。CLAP 是这两个嵌入之间的余弦相似度，并对所有片段取平均。对于 VT2A 和 T2A，字幕即文本提示；对于 V2A，字幕则是该片段的 VGGSound-Omni 字幕，模型不会获得此字幕。

B.4. 效率

两项指标均在单台 NVIDIA A100 GPU 上测量，批量大小为 1，且 GPU 上没有运行其他进程。每个模型均采用其质量评估中使用的采样步数、引导尺度和数值精度。计时前，会加载并在适用时编译每个模型，然后使用四个片段进行预热。随后再使用 20 个片段测量，并对指标取平均值。

首帧生成时间。 首帧时间（TTFF）是指从视频流开始到生成的首批音频样本可供 CPU 内存读取所需的时间。测量时，每个视频都会作为实时流回放，帧会以预先解码的形式提供，并且只有在其显示间隔结束时才会变得可用。模型的流式传输会话会在视频流开始前开启。因此，流式模型的 TTFF 包括等待第一个块所需视频的时间，以及对该视频进行编码、生成块并将其解码为波形所需的时间。在生成第一个块之前，LiveAudio 会等待 0.08 秒的视频，WorldSonus [19] 会等待 0.1 秒，V-AURA [73] 会等待 2.56 秒。离线模型只有在整个视频到达并处理完毕后才会生成音频，因此其 TTFF 超过视频时长，故不予报告。

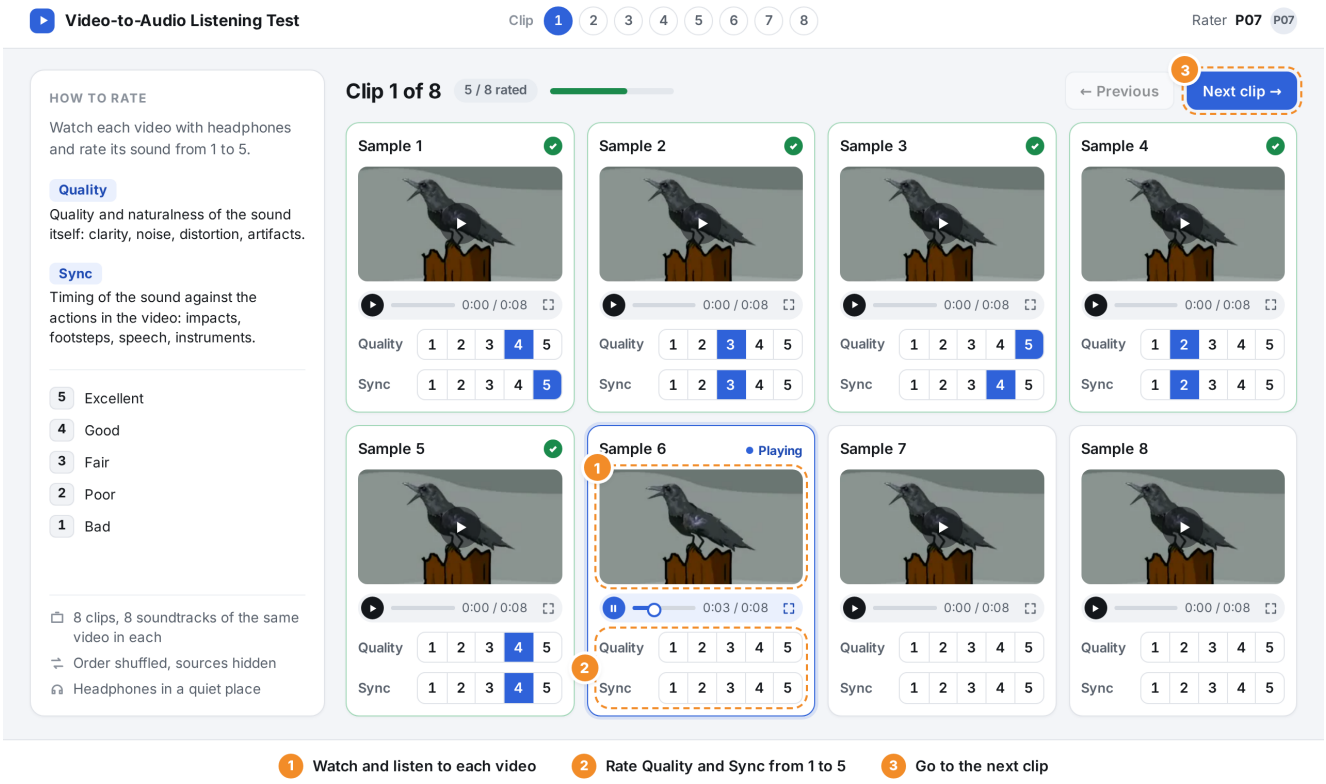

图7. 主观评估界面。每页显示一个视频、其真实音频以及七个模型按随机顺序生成的输出。参与者 (1) 观看每个视频，(2) 按 1 到 5 分评定质量（MOS-Q）和同步（MOS-T），(3) 进入下一个片段。

实时因子。 RTF 是实时因子，即视频处理时间与生成音频时长的比值；数值低于 1 表示模型的运行速度快于实时。测量时，完整视频文件从一开始就已可用，因此模型会尽可能快地处理视频。处理时间从视频文件到达开始，直到最后一批音频样本可供 CPU 内存读取为止。该时间涵盖视频解码、特征提取、采样和波形解码，但不包括模型加载或将音频写入文件的时间。

B.5. 主观评估

V2A 和 VT2A 的主观评估均由 20 位参与者完成，他们观看 VG-GSound 测试集中的八段视频。每段视频均配有其真实音频，以及各个对比模型生成的音频（V2A 见表 2，VT2A 见表 7），并针对每位参与者以随机顺序呈现，且不显示模型名称。V2A 的评分界面如图 7 所示。每段音频均保留其模型的采样率，截取前 8 秒，并与同一段 8 秒视频轨道配对；其响度归一化为 −23LUFS。建议参与者在安静的环境中使用耳机，并从以下方面对每段音频按五点评分量表打分，1 分（差），5 分（优）。

每个模型的结果均报告平均意见得分（MOS），即其评分的平均值，以及对应的 95%置信区间。

MOS-Q. MOS-Q 评定声音本身的质量和自然度，依据其清晰度以及是否存在噪声、失真和伪影进行判断。参与者评分时不考虑视频。

MOS-T. MOS-T 评定声音与视频中动作的时间同步程度，依据声音是否在事件发生的时刻出现进行判断，例如撞击声、脚步声、语音开始以及乐器演奏声。

MOS-A. MOS-A 评定声音与文本提示的匹配程度，依据提示中描述的声音事件是否出现在音频中进行判断。该评分仅针对 VT2A，因为每个视频都会显示文本提示。

C. 实现细节

C.1. 模型架构

VAE。 我们采用 SoundReactor [63]，的因果变分自编码器（VAE），[18] 它遵循 Stable Audio Open 的 Oobleck 架构，并在所有

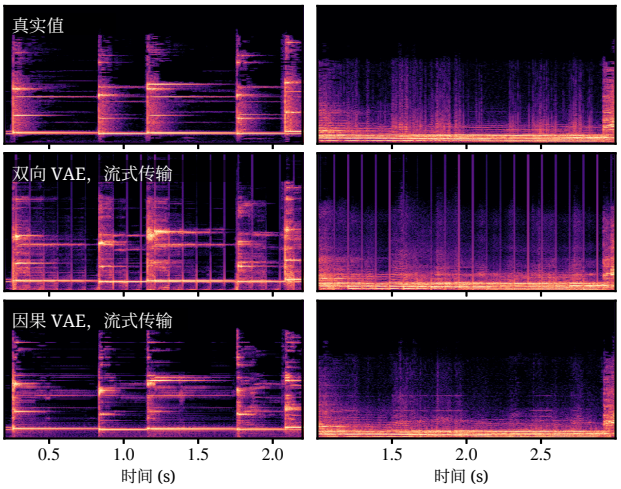

图 8. 真实值及其分别使用 Stable Audio Open 的双向 VAE 和因果 VAE 进行的流式解码，其中每个约 0.1 秒的块均在没有后续潜变量的情况下被解码。

阶段中保持冻结。其编码器将 48 kHz 立体声音频依次按 2、4、5、5 和 8 的步幅进行下采样，总计下采样 1600 倍，生成 30 Hz、64 维的潜变量；其解码器则通过转置卷积实现与编码器相反的操作。所有卷积均为因果卷积，仅在左侧进行填充，因此每个潜变量只依赖当前及过去的音频，而每个音频样本只依赖当前及过去的潜变量。SoundReactor 使用多分辨率 STFT 损失、对抗损失、特征匹配损失，以及针对潜变量的 KL 惩罚来训练 VAE。训练时，每个潜变量都从 VAE 后验分布中采样，该分布的均值和方差会针对每段音频预先计算。对于流式传输，解码器会转换为随 VAE 一同发布的流式解码器，其卷积层会在调用之间保留过去的输入，因此每生成三个潜变量组成的块，就会将其解码为 0.1 秒的音频。相比之下，双向 VAE 只有在与后续潜变量一同解码时，才能忠实地还原某个潜变量。图8使用 Stable Audio Open 的双向 VAE，逐块解码真实潜变量；解码时使用所有更早的潜变量，但不使用后续潜变量，因此每个块的边界处都会产生咔嗒声。使用因果 VAE 时，流式解码得到的波形与解码整个音频片段得到的波形相同。

DiT。 扩散 Transformer（DiT）基于 Stable Audio Open [18], 的架构构建，包含 24 层、24 个注意力头、宽度为 1536，以及不含文本编码器和 VAE 的 11.7 亿参数。每个 VAE 潜变量被投影为一个词元，因此一段 10 秒片段产生 300 个词元，构成 100 个由三个词元组成的块。每一层由带旋转位置嵌入的自注意力、对条件词元的交叉注意力以及前馈网络组成。双向模型与块因果模型共享该架构，仅在注意力掩码和噪声级别令牌上有所不同。

表 5. 四个阶段的训练超参数。LR 表示学习率，FP16 表示混合精度，EMA 表示权重的指数移动平均。

	阶段 1	阶段 2	阶段 3	阶段 4
迭代次数	500k	100k	30k	3k
LR	10^{-4}	2×10^{-6}	2×10^{-6}	2×10^{-6}
虚假分数的学习率	-	-	-	4×10^{-7}
学习率调度	平方根倒数	常数	常数	常数
批量大小	256	64	64	64
A100 GPU	32	8	8	8
精度	FP16	FP16	FP16	FP32
EMA 衰减	0.9999	-	-	-
条件丢弃	0.1	0.1	-	-
片段长度	10 s			
优化器	AdamW [54]			
AdamW β	(0.9，0.999)			
权重衰减	0.001			
梯度裁剪	None			

视频条件。 视频特征的提取方式与 MMAudio [7]相同。CLIP [62] 对以每秒8帧采样并缩放至 384×384 的帧进行编码，得到 1024 维特征；这些特征接收绝对位置嵌入，并与文本一同进入交叉注意力。一个 CLIP 帧在其时间戳之后 0.04 秒、即在每秒25帧视频中其显示区间结束时变为可用；仅当该帧在包含某音频词元起始位置的切片结束前可用时，该音频词元才会对其施加注意力。Synchformer [31] 在以每秒25帧采样、短边缩放至 224 并居中裁剪的帧上运行。遵循 LiveFoley [12], , Synchformer 在每个切片播放完成后，将以该切片结尾的尾随窗口编码为一个 768 维特征，因此一段 10 秒片段可得到 125 个特征。包含某音频词元起始位置的切片特征会被投影为该词元自适应层归一化的调制参数。

文本和时长条件。冻结的 Flan-T5 base [8]将文本编码为最多 128 个词元。与 Stable Audio Open 一样，开始时间和总时长的嵌入会输入交叉注意力，并添加到每个噪声水平标记中。任务未提供的条件将留空；对于 T2A, Synchformer 特征则设为零。

C.2. 训练细节

通用设置。所有阶段均使用附录A中描述的数据和任务混合进行训练，其超参数列于表5。阶段 2 从阶段 1 权重的指数移动平均（EMA）初始化，之后每个阶段均从前一阶段的权重初始化。VAE 和条件编码器保持冻结，仅训练 DiT 以及用于嵌入条件的层。

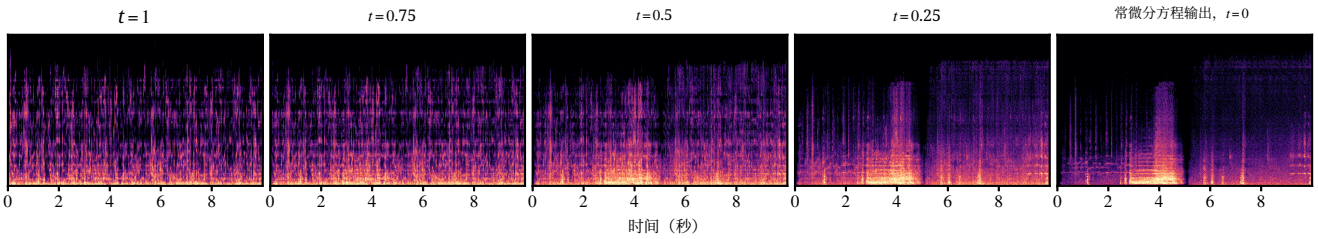

图 9。使用阶段 2 模型提取的训练片段的常微分方程轨迹。每幅频谱图都通过 VAE 解码某一时间步存储的潜变量，最后一幅是最终输出，阶段 3 模型学习根据其他各个状态预测该输出。

阶段 1。噪声级别从 logit-正态分布中抽取 [15]，$t = \text{sigmoid}(n)$ 其中 $n \sim \mathcal{N}(0, 1)$。对于无分类器引导（CFG），[27],以 0.1 的概率同时丢弃样本的视频和文本条件。学习率采用 Stable Audio Open 的调度方案 [18],：先进行指数预热，随后按平方根倒数衰减，$10^{-4}\,(1 - 0.99^{s+1})(1 + s/10^6)^{-1/2}$，在第 s 步时达到相应状态。衰减率为 0.9999 的权重 EMA 用作所有后续阶段的双向 DiT。

阶段 2。 我们通过将每个片段的干净副本和含噪副本打包到同一序列中来实现教师强制，其中每个副本的每个块都带有一个噪声水平标记。每个含噪块都会关注自身的词元以及较早块的干净副本，因此所有块都能利用真实历史并行训练。每个块都会独立采样自身的 logit-正态噪声级别，并保留阶段 1 的条件丢弃。

阶段 3。 使用阶段 2 模型，通过求解公式 4来提取轨迹：在从 $t = 1$ 到 $t = 0$的均匀网格上执行 48 个欧拉步，使用 4.5 的引导尺度和全精度。片段的所有块都结合真实历史进行联合去噪，48 个步将四步采样器的步时间 1、0.75、0.5 和 0.25 放置在该网格上。对于从训练数据的任务混合中抽取的 131,072 个训练片段，我们存储这四个时间点的状态以及最终输出。图 9 展示了一个片段的存储状态。在训练期间，每个样本都会为其所有块抽取四个步时间中的一个，并且不会丢弃任何条件，因为引导已包含在轨迹中。

阶段 4。 真实分数由阶段 1 的 EMA权重以 4.5 的引导尺度给出，伪分数模型则使用相同的权重进行初始化。每次迭代都会更新伪分数模型，而生成器每五次迭代更新一次，因此在 3k 次迭代中会进行 600 次生成器更新。遵循 Self Forcing [28], 生成器使用四步采样器，根据自身历史逐块生成每个片段；每个片段的退出步均匀抽取，并由所有块共享，且只有该步的输出接收梯度。如 Causal Forcing [81], 所示，分数在 $t = 5u/(1 + 4u)$ 处评估，$u \sim \mathcal{U}[0, 1)$并截断至 $[0.02, 0.98]$，且公式 5中的 DMD 梯度通过生成片段与真实分数 [77] 的预测之间的平均绝对差进行归一化。判别器头遵循 Presto [59], ，其中四个带有组归一化和 SiLU 的卷积块，后接一个 1×1 卷积，将伪分数模型第 12 个块之后的特征映射为每四个词元对应一个分数。在进入判别器头之前，真实潜变量和生成潜变量会在 logit-normal 噪声级别下加入噪声；该判别器头使用最小二乘损失进行训练。对抗损失在生成器损失中的权重为 0.01，在伪分数损失中的权重为 0.005；判别器头与伪分数模型一起以 10^{-4} 的学习率进行更新。

C.3. 推理细节

多步采样。 双向 DiT 和块因果 DiT 在从 $t = 1$ 到 $t = 0$ 的均匀网格上，使用欧拉方法进行 50 步采样。对于 CFG，无条件预测会以与训练中条件丢弃相同的方式移除条件，引导尺度为 4.5。双向 DiT 使用 EMA权重，一次性对整个片段进行去噪。块因果 DiT 则按顺序生成各个块，每个块采样 50 步；每个已完成的块都会写入两种预测各自的 KV 缓存中。

少步采样。 阶段 3 和阶段 4 的模型使用四步采样器，步进时间为 1、0.75、0.5 和 0.25，且不使用引导。KV 缓存会保留片段中所有较早的块，因此每个块都能关注完整历史。

时长条件。 由于无法预先知道流的长度，LiveAudio 默认生成 10 秒音频，并将其裁剪至与视频时长一致。

流式处理管线。 LiveAudio 的质量与其延迟采用相同的流式处理管线进行评估。视频按显示顺序仅解码一次，并以每秒8帧的速度为 CLIP 采样帧、以每秒25帧的速度为 Synchformer 采样帧；随着帧的到达，逐步计算其特征。每个块通过两个 CUDA 图运行：一个用于四个采样步骤，另一个用于写入 KV 缓存的过程；同时采用 BF16自动混合精度，并预先将权重转换为 BF16。一个块处理完毕后，流式 VAE解码器便会将其中的三个潜变量转换为 0.1 秒的 48kHz音频。

表 6。使用来自 VGGSound-Omni 的字幕，在 VGGSound 测试集上进行的视频文本到音频结果。如表 1所示，首帧生成时间（TTFF）仅针对流式模型进行报告。标有 † 的结果引自原论文。教师模型是我们的双向 DiT。

方法	分布匹配					音频质量		模态对齐			效率	
	FD_O↓	FD_V↓	FD_P↓	KL_S↓	KL_P↓	IS_P↑	PQ↑	不同步度↓	IB↑	CLAP↑	首帧生成时间 (s)↓	RTF↓
离线												
MMAudio	69.01	0.83	4.22	1.41	1.62	14.90	5.99	0.447	0.331	0.373	-	0.38
AudioX	73.47	1.24	8.01	1.55	1.57	15.14	6.29	1.235	0.266	0.409	-	1.35
ThinkSound	47.12	1.15	8.89	1.55	1.72	10.51	6.06	0.555	0.255	0.358	-	0.71
混元视频拟音	80.65	1.90	10.30	1.44	1.78	11.39	6.19	0.534	0.319	0.364	-	1.52
Omni2Sound	142.61	0.58	5.01	1.19	1.40	15.84	6.12	0.521	0.334	0.385	-	1.06
LiveAudio（教师模型）	67.66	1.28	4.33	1.33	1.44	14.87	6.25	0.498	0.323	0.327	-	0.52
流式传输												
WorldSonus	89.96	1.92	8.91	1.45	1.49	10.19	6.29	0.715	0.263	0.330	0.554	0.62
LiveFoley†	-	-	7.34	-	1.61	11.21	6.38	0.490	0.290	-	-	-
LiveAudio（我们的方法）	69.63	1.19	5.78	1.38	1.51	15.20	6.41	0.421	0.320	0.324	0.145	0.81

D. 补充结果

D.1 视频文本到音频生成

表 6报告了在 VGGSound 测试集上的 VT2A 结果，采用 VGGSound-Omni 的字幕作为文本提示（附录A.2）。我们保留了表 1 中的基线模型，这些模型接受视频和文本提示作为输入，包括 MMAudio [7],AudioX [70], Omni2Sound [11], WorldSonus [19] 和 LiveFoley [12], , 并另外与 ThinkSound [51]和 HunyuanVideo-Foley [64], 进行比较；这两种离线模型均接受两种输入。V-AURA [73]和 SoundReactor [63] 仅支持流式视频到音频生成，不接受可选的文本输入。对于 ThinkSound，我们使用字幕替代多模态语言模型通常会生成的思维链描述。LiveFoley 的结果引自其论文，该论文针对这项任务训练了一个文本条件变体；其他所有基线模型均采用其官方检查点和推理设置运行，与表1一致。与 V2A 一样，LiveAudio 在所有模型中实现了最低的不匹配度和最高的感知质量。它在大多数指标上优于 WorldSonus，在所有报告的指标上优于 LiveFoley；在分布匹配、音频质量和模态对齐方面，则与离线模型相当。在效率方面，LiveAudio 的运行速度快于实时处理，且首帧生成时间远低于 WorldSonus。表7 报告了主观评估结果，其中 MOS-A 用于评定音频与文本提示之间的对齐程度（附录B.5）。LiveAudio 在所有模型中取得了最高的 MOS-T；其 MOS-Q 和 MOS-A 在流式模型中最高，也接近最佳离线模型的水平。

D.2 文本到音频生成

除表 3 中的 AudioCaps 测试集外，表 8还在 VGGSound-Omni 上评估了 T2A [11] 以及在评估-

表 7：视频文本到音频生成的主观评估。我们报告了具有 95% 置信区间的平均主观评分。

方法	MOS-Q↑	MOS-T↑	音频文本对齐主观评分↑
真实值	$4.38_{\pm 0.17}$	$4.43_{\pm 0.12}$	$4.39_{\pm 0.16}$
离线			
MMAudio	$3.66_{\pm 0.17}$	$3.98_{\pm 0.17}$	$3.61_{\pm 0.17}$
AudioX	$3.51_{\pm 0.18}$	$3.02_{\pm 0.18}$	$3.21_{\pm 0.20}$
ThinkSound	$3.38_{\pm 0.21}$	$3.70_{\pm 0.15}$	$3.36_{\pm 0.29}$
混元视频拟音	$3.60_{\pm 0.16}$	$3.92_{\pm 0.14}$	$3.53_{\pm 0.18}$
Omni2Sound	$3.80_{\pm 0.14}$	$4.10_{\pm 0.12}$	$3.81_{\pm 0.18}$
流式传输			
WorldSonus	$3.55_{\pm 0.16}$	$3.73_{\pm 0.18}$	$3.48_{\pm 0.19}$
LiveAudio（我们的方法）	$3.74_{\pm 0.19}$	$4.11_{\pm 0.17}$	$3.62_{\pm 0.25}$

ation set ofClotho v2 [14], 这两者在训练中均未使用。我们与相同的基线模型进行比较，即 Au- dioLDM [48], Tango [22], AudioLDM 2 [50], Stable Au-dioOpen [18], TangoFlux [30], MMAudio、AudioX、Au- dioGen [42] 和 WorldSonus。在 VGGSound-Omni 上，每个模型都根据测试片段的字幕生成音频，不使用视频。Clotho 中的片段时长为 15 至 30 秒，每段片段配有五条人工撰写的描述；每个模型根据第一条描述生成 10 秒音频，并与真实值的前 10 秒进行比较。WorldSonus [19]未报告 T2A 结果，因此我们调整了其官方代码以完成这项任务。其结果以 ‡ 标记，使用其采样器的无视频分支生成；该分支使用学习得到的空视频词元，并以离线方式运行，而非流式运行，其 RTF 也在此分支上测量。在两个基准上，LiveAudio 的表现仍可与离线模型媲美，且优于流式模型，这与 AudioCaps 上的结果一致。为评估效率，我们还报告了生成 10 秒片段的 RTF；LiveAudio 在 T2A 任务中也实现了实时生成。

表 8. 文本到音频结果　　在 VGGSound-Omni 和 Clotho v2 的评估集上。WorldSonus‡ 改编自其官方代码。

方法	VGGSound-Omni						Clotho v2						效率
	FD_V↓	FD_P↓	KL_P↓	IS_P↑	感知质量↑	CLAP↑	FD_V↓	FD_P↓	KL_P↓	IS_P↑	感知质量↑	CLAP↑	RTF↓
离线													
AudioLDM	3.55	24.89	2.32	7.02	5.93	0.389	3.64	19.48	2.33	7.48	5.82	0.336	1.94
Tango	3.71	21.36	1.96	6.61	6.07	0.395	5.20	27.33	2.43	6.18	6.24	0.281	1.43
AudioLDM 2	2.78	26.91	1.95	11.17	6.33	0.289	3.75	28.76	2.42	8.50	6.25	0.298	2.38
Stable Audio Open	3.72	27.76	2.20	11.78	6.60	0.362	3.20	29.90	2.52	9.49	6.47	0.379	2.93
TangoFlux	2.09	16.54	1.74	10.47	6.12	0.346	2.51	21.76	2.54	10.94	6.01	0.327	0.38
MMAudio	1.56	8.98	2.03	14.62	5.99	0.377	2.37	15.58	2.50	9.34	6.11	0.381	0.17
AudioX	1.57	10.62	1.60	15.75	6.29	0.382	2.86	21.39	2.61	11.38	6.08	0.336	1.01
流式传输													
AudioGen	2.37	13.91	2.10	11.00	5.49	0.287	2.42	19.82	2.61	7.43	5.34	0.288	1.75
WorldSonus‡	2.47	8.38	1.51	10.65	6.39	0.339	3.38	19.71	2.61	8.17	6.13	0.271	1.86
LiveAudio（我们的模型）	1.73	8.87	1.69	15.41	6.65	0.328	4.19	18.90	2.52	9.39	6.46	0.299	0.42

表 9. 视频到音频生成中采样步数的消融实验，评估数据为 VGGSound 测试集。每一行代表一个独立的生成器：它们均在阶段 4 中从同一个第 3 阶段模型蒸馏而来，采样步数分别为 1、2 或 4 步；其中，四步生成器即为 LiveAudio。

步数	分布匹配					音频质量		模态对齐		效率		
	FD_O↓	FD_V↓	FD_P↓	KL_S↓	KL_P↓	IS_P↑	PQ↑	不同步度↓	IB↑	CLAP↑	首帧生成时间（秒）↓	RTF↓
1	83.31	2.13	10.18	2.04	2.13	13.96	6.22	0.405	0.298	0.246	0.123	0.57
2	73.42	1.32	7.70	1.96	2.12	15.16	6.27	0.408	0.318	0.252	0.129	0.64
4（我们的方法）	70.97	1.30	7.32	1.99	2.12	15.87	6.41	0.395	0.323	0.252	0.143	0.79

E. 附加消融研究与分析

E.1. 采样步数

表9 研究了生成器的采样步数。我们以一步、两步和四步采样重复阶段 4，其中每个生成器均在相同设置下由同一个第 3 阶段模型蒸馏而来。使用两步时，生成器在分布匹配和模态对齐方面仍接近四步生成器，但音频质量略低。使用一步时，不同步度与两步时相当，而分布匹配、音频质量和 IB 均明显下降，因此单步主要牺牲生成质量，而非同步性。步数减少后，首帧生成时间和 RTF 仅小幅降低，因为无论采样步数多少，视频编码耗时都相同（图5）。由于四步采样的运行速度已快于实时，我们默认采用四步。

E.2. 视频条件

表10 给出了图 3 的数值结果，其中 (a) 在全部四个阶段采用标准条件控制，(b) 则采用同步对齐条件控制。标准条件控制是 offlineV2A 模型的视频条件控制，我们遵循 Omni2Sound [11], 的做法，即在生成前提取整个视频的特征，而帧音频词元接收的内容不局限于已经播放的帧。Synchformer 将视频编码为每段 16 帧、步长为 8 帧的片段，因此调制音频词元的特征涵盖整个片段，并包含该词元之后最多约 0.64 秒的视频；在双向阶段，每个词元都能看到所有帧的 CLIP 特征。同步对齐条件控制通过相同的两条路径注入特征，但每个词元只能接收已经播放的帧（附录 C.1）。在双向阶段，两者表现相当，标准条件控制因具备前瞻而获得略低的不 $_V$步$_P$度$_P$ 。从阶段 2 开始，流式生成不再具备前瞻，此时同步对齐条件控制在 FD [12] 、FD、IS 和不同步度方面明显更优，这证实了同步对齐条件控制 对流式生成有效。这与第 4.3 节的结论一致。

E.3. 非对称DMD

非对称 DMD 是恢复因果生成器同步的阶段（第4.3节），因此我们在表11 中进一步考察了其设计。我们测试了三种变体，每种都只更改一个组件，并在相同设置下从同一个第 3 阶段模型蒸馏而来。第一种变体移除对抗损失，以考察这种损失在分布匹配之外的作用。

表 10. 在 VGGSound 测试集上，经过四个训练阶段中的每个阶段后，视频条件在视频到音频生成中的比较。（a）所有阶段均采用标准条件控制。（b）所有阶段均采用同步对齐条件控制。

阶段	分布匹配					音频质量		模态对齐		
	FD_O↓	FD_V↓	FD_P↓	KL_S↓	KL_P↓	IS_P↑	PQ↑	不同步度↓	IB↑	CLAP↑
（*a*）标准										
阶段 1（双向）	66.50	1.30	4.73	1.77	1.94	15.10	6.23	0.450	0.324	0.264
阶段 2（块因果）	76.20	2.63	9.76	2.14	2.21	10.55	6.35	0.610	0.282	0.225
阶段 3（常微分方程）	70.09	2.18	9.02	2.07	2.22	10.83	6.35	0.616	0.278	0.227
阶段 4（DMD）	75.68	2.15	9.35	1.91	2.09	12.77	6.43	0.488	0.303	0.248
（*b*）同步对齐										
阶段 1（双向）	68.17	1.29	5.11	1.86	1.99	15.81	6.26	0.484	0.324	0.260
阶段 2（块因果）	68.19	1.44	6.79	2.08	2.15	12.54	6.24	0.535	0.307	0.235
阶段 3（常微分方程）	68.17	1.18	6.22	2.04	2.19	12.91	6.21	0.544	0.301	0.236
阶段 4（DMD）	70.97	1.30	7.32	1.99	2.12	15.87	6.41	0.395	0.323	0.252

表 11。不对称性的消融　　在 VGGSound 测试集上的视频到音频生成中进行 DMD 对比。每个变体仅更改一个组件。

变体	分布匹配					音频质量		模态对齐		
	FD_O↓	FD_V↓	FD_P↓	KL_S↓	KL_P↓	IS_P↑	PQ↑	不同步度↓	IB↑	CLAP↑
LiveAudio（我们的方案）	70.97	1.30	7.32	1.99	2.12	15.87	6.41	0.395	0.323	0.252
无对抗损失	85.25	2.15	9.23	2.06	2.14	14.81	6.36	0.403	0.308	0.246
对称*DMD*	94.84	2.71	11.96	2.25	2.31	12.49	6.38	0.502	0.269	0.219
真实历史	191.82	8.12	44.49	3.60	3.08	5.76	6.02	0.711	0.147	0.103

第二种变体是对称 DMD，它使用阶段 2 的块因果 DiT 而非双向 DiT 提供真实分数，并以相同权重初始化伪分数模型，用以检验双向模型的监督是否必不可少。第三种变体在训练时，像阶段2和阶段3那样，根据较早块的真实音频生成每个块，而不是根据生成器的输出生成，用以检验生成器是否必须基于自身历史进行训练。移除对抗损失后，不同步度几乎没有变化，而 FD_O、FD_V、FD_P 和 IS_P 则明显下降。这与 Presto [59]和 AudioX-Turbo [71], 的做法一致：它们在 DMD 中加入针对真实音频的对抗损失，以提升音频的感知质量。这也表明，在音频模型的 DMD 中，对抗损失主要影响音频质量。采用对称 DMD 后，所有指标均下降，而不同步度仅略优于第 3 阶段（表10），因为块因果 DiT 同样无法看到后续音频或视频，这证实了同步是通过双向 DiT 的监督恢复的。使用真实历史时，所有指标都远差于第 3 阶段，这表明分布匹配只有应用于生成器根据自身历史生成的音频时才有效 [28]。

E.4. 长音频生成

在流式应用中，流的长度事先无法确定，并且可能远超10秒片段的长度

表12. Audio- Caps 测试集上的长时（60秒）文本到音频生成。WorldSonus‡ 基于其官方代码进行适配。

方法	FD_V↓	FD_P↓	KL_P↓	IS_P↑	感知质量↑	CLAP↑	RTF↓
AudioGen	2.81	19.68	1.73	8.76	5.37	0.421	1.79
AudioLDM	5.50	34.39	1.96	7.26	5.81	0.399	0.67
AudioLDM 2	6.17	36.49	1.87	8.55	5.96	0.376	0.62
WorldSonus‡	4.97	27.84	2.10	7.87	5.79	0.377	1.83
LiveAudio（我们的方法		22.94	1.95	10.14	5.91	0.404	0.42

LiveAudio 的训练方式。因此，我们在表12中进一步探索了 LiveAudio 的长度外推能力：无需额外训练即可生成 60 秒的音频。LiveAudio 作为流持续逐块生成，并且在 KV 缓存中仅保留最近 5 秒的音频，使所有块都处于训练时见过的范围内。AudioLDM [48]通过外绘延长片段，这是一种离线过程：对每个 10 秒窗口整体进行去噪，其中前半部分固定为已生成的音频，后半部分追加到片段中。AudioLDM 2 [50] 一次性生成音频，而 AudioGen [42]和 WorldSonus [19] 则借助滑动上下文继续自回归生成。我们遵循长音频生成的通用评估协议 [10, 16, 17, 33]。LiveAudio 获得了最高的 IS_P，在 FD 方面仅次于 $AudioGen_V$ 和 FD_P，并且在所有指标上都优于 WorldSonus，其 KL_P 接近离线模型的水平。这表明，无需额外训练，LiveAudio 也能超越其训练长度。

F. 案例研究

我们在演示页面https://streamaudio.github.io/上提供了 V2A、VT2A 和 T2A 的生成样本及其与现有方法的对比。

下面，我们提供了更多 V2A 的频谱图可视化（图10和11）、VT2A（图 12）、T2A（图 13）以及长篇 T2A（图 14）。

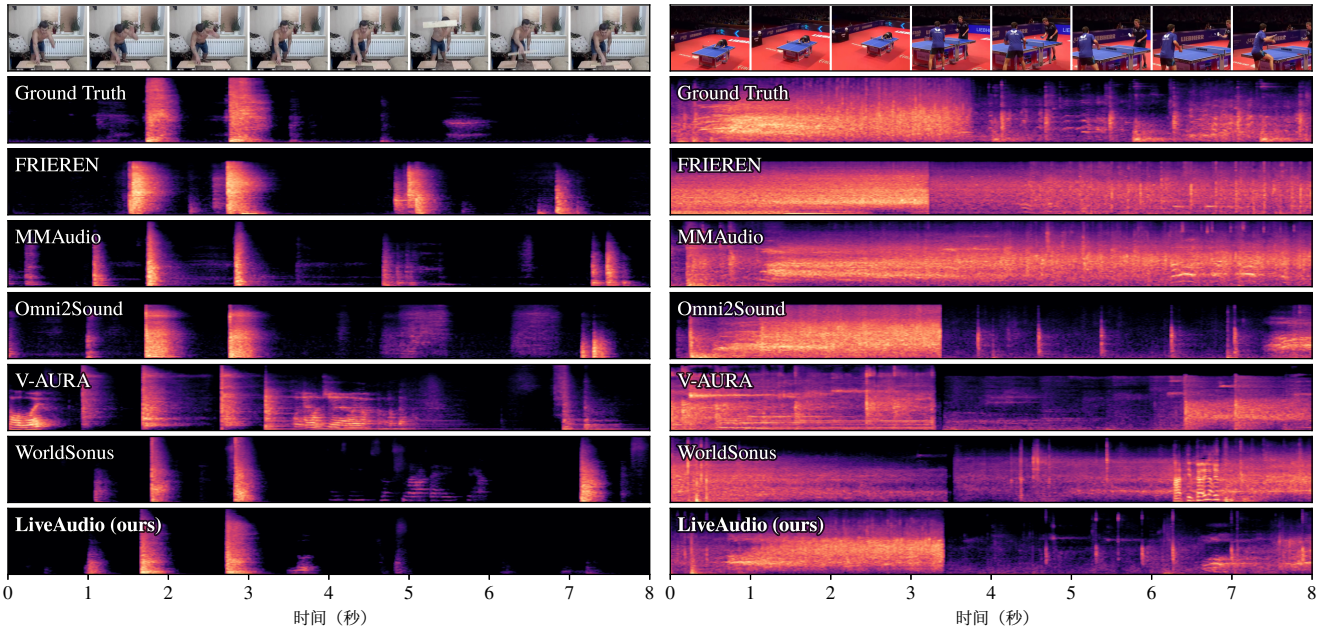

图 10. 两个测试片段上的视频到音频生成定性比较。

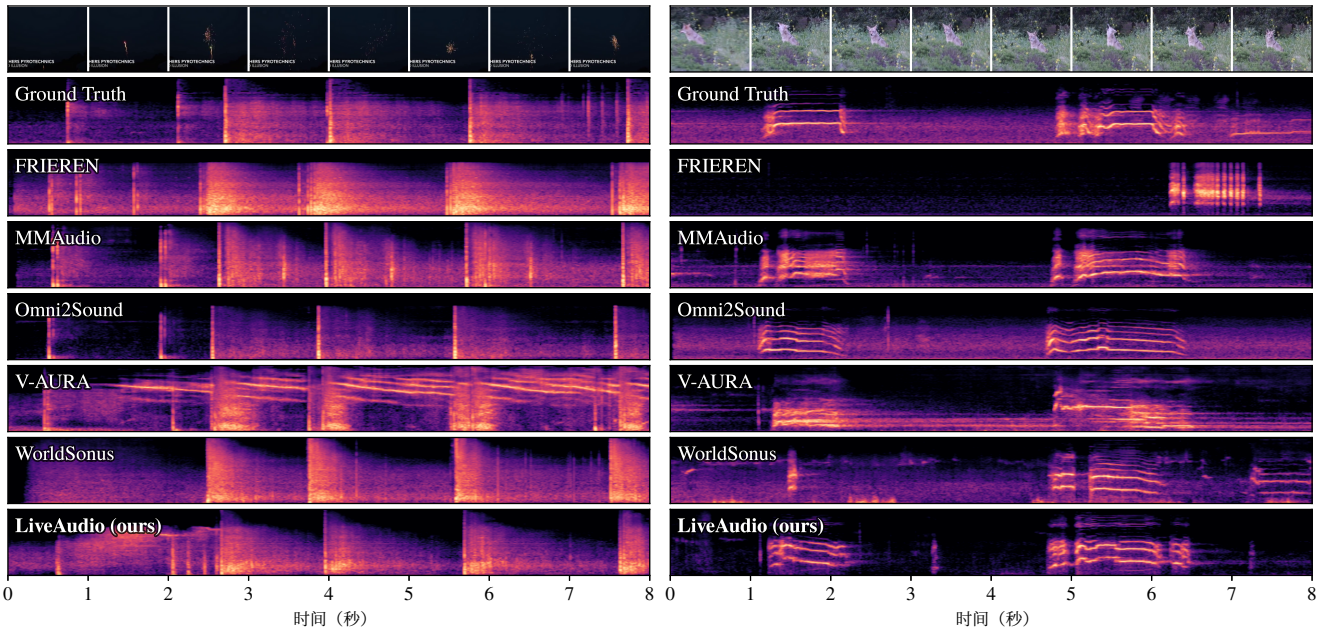

图 11. 另外两个测试片段上的视频到音频生成定性比较。

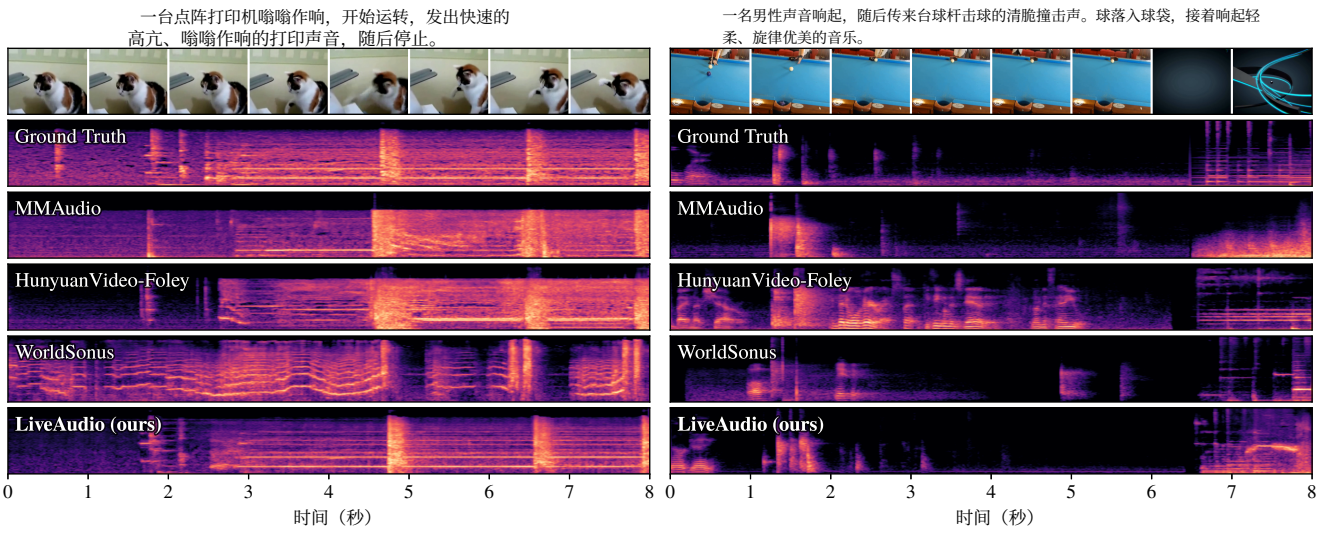

图12. 视频文本到音频生成的定性比较，展示两个测试片段．每个片段上方均显示文本提示．

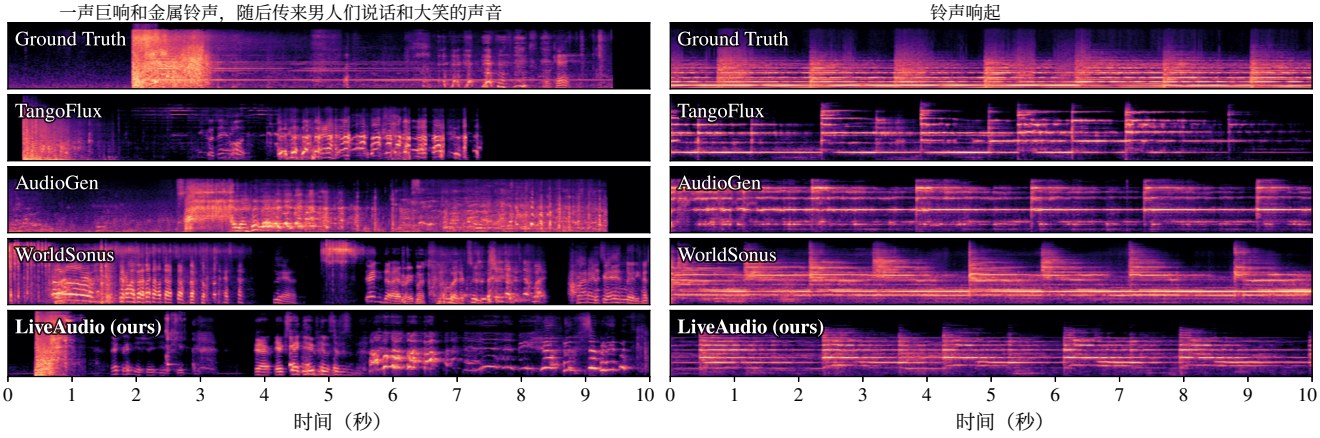

图13. AudioCaps 测试集两个字幕的文本到音频生成定性比较。每个示例上方均显示文本提示。

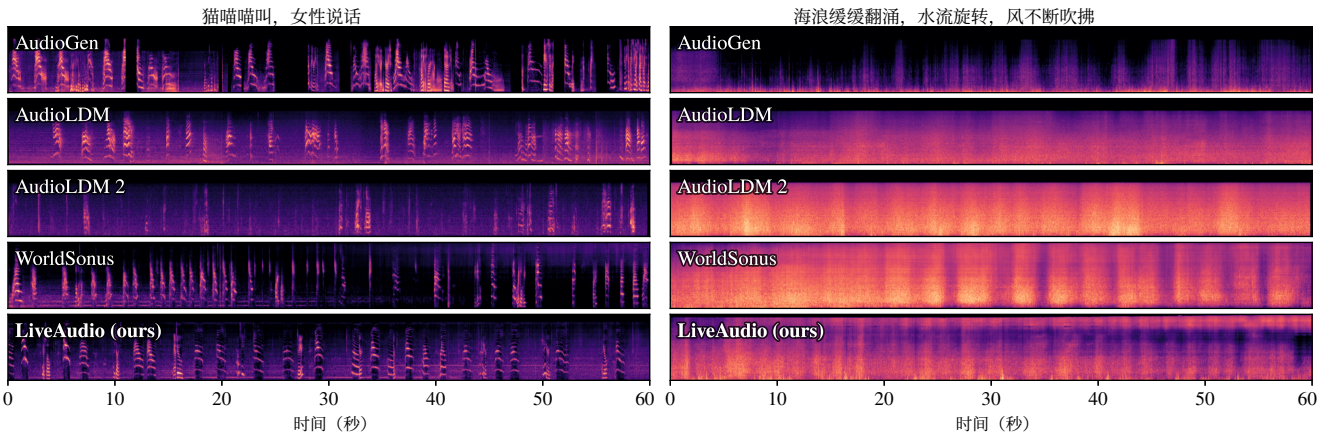

图14. AudioCaps 测试集两个字幕的长篇文本到音频生成定性比较，其中每个模型均生成 60 秒音频。每个示例上方均显示文本提示。