arXiv:2603.04366v1 [cs.SD] 4 Mar 2026

低资源 面向可控潜在音频扩散的CE引导

Zachary Novack[1, 2†], *Zack Zukowski*[2], *CJ Carr*[2], *Julian Parker*[2], *Zach Evans*[2], *Josiah Taylor*[2], *Taylor Berg-Kirkpatrick*[1], *Julian McAuley*[1], *Jordi Pons*[2]

[1]加州大学圣迭戈分校, [2]Stability AI, [†]于Stability AI实习期间完成的工作

摘要

生成式音频需要细粒度的可控输出，但现有大多数方法要么需要针对特定控制信号进行模型重新训练，要么依赖推理时控制（例如，引导），而这些方法在计算上也可能要求很高。通过分析现有基于引导的控制所存在的瓶颈，尤其是解码器反向传播带来的高单步成本，我们提出一种借助选择性TFG与潜在控制头（LatCHs）的引导方法，从而以较低的计算开销控制潜在音频扩散模型。LatCHs直接在潜在空间中运行，避免了代价高昂的解码器步骤，并且所需训练资源极少（7M个参数以及 ≈4 小时的训练）。在Stable Audio Open上的实验表明，该方法能够在保持生成质量的同时，有效控制强度、音高和节拍（及其组合）。与标准端到端引导相比，我们的方法以低得多的计算成本平衡了精度与音频保真度。演示示例见 zacharynovack.github.io/latch/latch.html。

索引词— 扩散引导、可控音频生成、可控音乐生成

1. 引言

生成模型在音频领域取得了迅速发展，能够根据文本生成连贯的声音生成结果 [1, 2]。然而，随着创作工作流程对细粒度操控的需求不断增加，人们对可控音频生成的需求也日益迫切。近期的研究主要依赖于基于和弦、节奏或音高的局部条件控制 [3, 4], 、基于风格或参考音频嵌入的全局条件控制 [5, 6], 、视频条件控制 [7], 、编辑 [8], , 或根据上下文生成分轨（例如，其他分轨）[9, 10]。尽管这些方法能够实现用户可操控的生成，但此类（条件）模型需要使用难以收集的数据进行监督训练或微调。

鉴于训练或微调生成模型的成本高昂，近年来人们开始探索推理时控制方法。文本到图像领域的研究主要关注基于启发式的推理时方法 [11], 基于优化的方法 [12, 13], 以及基于引导的方法 [14, 15, 16, 17, 18, 19, 20, 21]。我们也聚焦于基于引导的方法，这类方法利用目标距离函数相对于扩散过程的梯度来引导采样。免训练引导 (TFG) [21] 在统一的超参数空间内整合了大多数基于引导的框架 (DPS [17], MPGD [18], LGD [19], UGD [20])。然而，由于采样时需要通过音频解码器进行反向传播，此类方法仅在端到端音频引导领域得到探索 [22], , 而这种引导成本高昂。读出 [14]是一个源自图像领域的有前景的思路，它通过调整引导，使其基于潜在扩散特征（而非解码后的图像），从而应对这一挑战。

我们基于这些思路，展示了使用可训练的 潜变量控制头（受读出启发）实现低资源引导（基于 TFG）的可行性。这类**潜变量控制头**（**LatCHs**）能够在潜在扩散中实现低资源引导，因为它们在潜在空间而非信号空间中运行。因此，无需通过潜在解码器进行反向传播（图 1）。此外，LatCHs 是轻量级模型（≈7M 个参数），可在单个图形处理器上于 ≈4 小时内完成训练，因此比完全条件化的生成模型更易于训练。最后，我们还提出了**选择性 TFG**，仅在少数选定的扩散步骤中应用 TFG 引导（图 1）。这一设计选择大幅提升了运行效率和整体质量，因为引导不易过度优化目标控制属性。我们将低资源引导框架应用于 Stable Audio Open（SAO） [2]，涵盖三种音乐控制属性（强度、音高和节拍），结果表明，与类似的低资源方法相比，该框架能更好地平衡控制精度与音频保真度，且比标准的端到端引导更加节省计算资源。

© 2026 IEEE。允许个人使用本材料。对于任何其他用途，无论是在当前还是未来的任何媒体中，包括出于广告或宣传目的重新印刷/再版本材料、创作新的集体作品、转售或将其重新分发到服务器或列表，或在其他作品中重复使用本作品的任何受版权保护部分，均须获得 IEEE 的许可。

2. 背景与我们的方法

我们旨在寻找基于引导的方法，在控制潜在音频扩散模型时，避免显著损害质量或增加运行延迟。为此，我们提出了两项方法上的贡献（图 1）：(1) 选择性 TFG（第 2.3 节），我们扩展了近期提出的 TFG [21]引导框架，使其支持仅在少数选定的扩散步骤中进行引导；(2) 潜变量控制头（LatCHs，第 2.4 节），它们将生成模型的潜在空间直接映射到目标控制，从而将延迟速度提升一个数量级。据我们所知，我们是首个使用潜变量控制头在少数选定的扩散步骤中进行引导的团队。

2.1. 潜在音频扩散背景

给定文本 $\boldsymbol{c}$和立体声音频$\mathbf{x}_0 \in \mathbb{R}^{2\times TS}$，其中 T 是长度（47.55 秒），S 是采样率（44.1 kHz），我们希望从 $p(\mathbf{x}_0|\boldsymbol{c})$ 中进行采样。实际操作中，我们首先使用预训练的自动编码器 $\mathcal{E}$将高维波形嵌入压缩的潜在空间 $\mathcal{D}$，使得$\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$。我们首先定义一个前向过程，根据$\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \epsilon$逐步破坏干净潜变量$\mathbf{z}_0$ （从数据到噪声），其中 $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$， α_t和 σ_t定义噪声调度，而$\mathbf{z}_1$ 对应完全的高斯噪声。随后，这一过程自然地导出一个反向过程（从噪声到数据），其定义为 [23]:

$$\mathrm{d}\mathbf{z}_t = \big(f(\mathbf{z}_t, t) - g(t)^2 \nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t \mid \boldsymbol{c})\big)\,\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \quad (1)$$

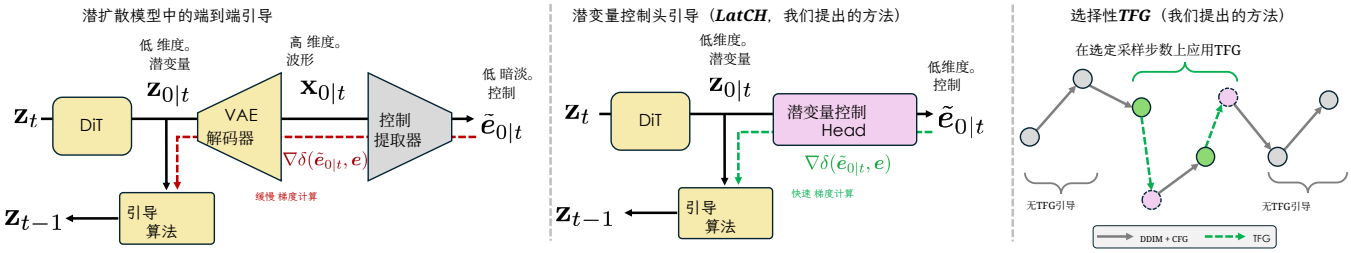


图1。*左*。端到端引导可能较慢且显存占用较高，因为它需要通过VAE解码器进行反向传播。*中*。LatCH计算效率较高，因为它直接从潜在空间预测控制特征。*右*。选择性TFG同样计算效率较高，因为它允许仅在选定的采样步数上应用TFG引导。

其中，$\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)$ 是 $\mathbf{z}$ 在时间 t处边缘分布的得分，$\mathbf{w}$ 是维纳过程，而 f 和 g 是由 α_t 和 σ_t定义的漂移系数和扩散系数。因此，基于得分的潜变量扩散旨在学习模型 $\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c}) \approx \nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t \mid \boldsymbol{c})$，以近似随时间变化的得分，并对方程 1 进行数值积分（从 $\mathbf{z}_1$开始），从而从 $p(\mathbf{z}_0 \mid \boldsymbol{c})$ 中获得样本（进而通过解码器 $\mathcal{D}$得到 $\mathbf{x}_0$）。在我们的工作中，我们采用 v-扩散参数化，并学习一个 v-预测模型 $\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c})$，使得 $\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c}) = -(\mathbf{z}_t + \frac{\alpha_t}{\sigma_t}\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c}))$。此外，对于数值积分器，我们采用 DDIM采样器 [24]:

$$\mathbf{z}_{t-1} = \alpha_{t-1}\mathbf{z}_{0|t} + \sqrt{\sigma_{t-1}^2 - \eta_t^2}\boldsymbol{\varepsilon}_t + \eta_t \epsilon, \tag{2}$$

其中$\mathbf{z}_{0|t} = \alpha_t \mathbf{z}_t - \sigma_t \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c})$是时刻 t的预测干净数据，$\boldsymbol{\varepsilon}_t = \sigma_t \mathbf{z}_t + \alpha_t \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c})$是时刻 t的预测噪声，$\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$，并且 η_t 是与时间相关的噪声参数。

2.2. TFG背景

TFG [21] 旨在给定预训练的潜变量扩散模型 $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{c})$ 的情况下，从 $p(\mathbf{x}_0 \mid \boldsymbol{c}, \boldsymbol{e})$ 中进行采样，其中 $\boldsymbol{e}$ 是扩散模型未曾训练过的某种额外控制信号（例如 RMS能量）。此外，令 $\mathbf{C}(\cdot): \mathbb{R}^{2\times TS} \to \mathbb{R}^d$ 表示一个可微的 d 维特征提取器，使得 $\hat{\boldsymbol{e}} = \mathbf{C}(\mathbf{x}_0)$，并令$\delta(\hat{\boldsymbol{e}}, \boldsymbol{e})$表示某个选定的距离函数，用于评估我们生成的样本 $\mathbf{x}_0$ 在多大程度上遵循控制信号 $\boldsymbol{e}$。在（自编码器潜在空间中）评估该概率的得分时，我们有:

$$\begin{aligned}\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t \mid \boldsymbol{c}, \boldsymbol{e}) &= \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t \mid \boldsymbol{c}) + \nabla_{\mathbf{z}_t} \log p(\boldsymbol{e} \mid \mathbf{z}_t, \boldsymbol{c}) \\ &\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, \boldsymbol{c}) - \rho_t \nabla_{\mathbf{z}_t} \delta(\tilde{\boldsymbol{e}}_{0|t}, \boldsymbol{e}) - \mu_t \nabla_{\mathbf{z}_{0|t}} \delta(\tilde{\boldsymbol{e}}_{0|t}, \boldsymbol{e}),\end{aligned} \tag{3}$$

其中，$\tilde{\boldsymbol{e}}_{0|t} = \mathcal{N}(\mathbf{C}(\mathcal{D}(\mathbf{z}^{0|t})), \gamma_t)$ 是在时刻 t，根据我们对输出 $\mathbf{x}_{0|t} = \mathcal{D}(\mathbf{z}^{0|t})$ 的估计，与方差为 γ_t 的高斯噪声 $\mathcal{N}(\cdot)$ 卷积后得到的预测特征 $\hat{\boldsymbol{e}}_{0|t} = \mathbf{C}(\mathbf{x}_{0|t})$。我们将 ρ_t、μ_t、γ_t 分别分解为 $\rho \cdot s(t)$、$\mu \cdot s(t)$、$\gamma \cdot s(t)$，其中 $s(t) = \alpha_t / \sum_{t=1}^{T} \alpha_t$ 定义了它们的时间结构 [21]。式 3 中的第一项梯度称为“方差引导” [21]，并由 ρ_t 控制；该导数是对 $\mathbf{z}_t$ 求取的，并会经过扩散模型。第二项梯度称为“均值引导” [21]，并由 μ_t 控制；该导数是对预测均值 $\mathbf{z}^{0|t}$ 求取的，值得注意的是，它不会反向传播回扩散模型，因此计算速度快得多。在实践中，式 3 中的损失梯度可以直接从 $\mathbf{z}_{t-1}$（针对“方差引导”）和 $\mathbf{z}^{0|t}$（针对“均值引导”）中减去，分别在应用式 2 的更新之后和之前进行，从而对每个优化目标执行一步梯度下降。TFG 引入了两项额外的算法修改:

- 迭代于 $\mathbf{z}^{0|t}$ ——由于“均值引导”是对 $\mathbf{z}^{0|t}$ 的标准优化步骤，它可以在每个采样步骤中重复 $N_{\text{迭代}}$ 次，以实现更好的控制对齐。

- 引导递归——类似地，可以通过时间回溯 [16] 从 $\mathbf{z}_{t-1}$ 返回到 $\mathbf{z}_t$，并对 $p(\mathbf{z}_t \mid \mathbf{z}_{t-1})$ 进行采样（即从前向过程进行采样），从而在每个采样步骤中重复整个引导过程 N_{recur} 次。

这定义了 TFG 的搜索空间，包括 ρ、μ、γ和 N_{iter}、N_{recur}。直观地说，ρ 和 μ 会调整引导强度，而 γ 则通过向 $\hat{\boldsymbol{e}}_{0|t}$添加噪声来调整引导的准确性强度。最后，$N_{\text{iter}}$ 和 N_{recur} 设定迭代引导的次数。我们还尝试使用多个控制信号进行引导（例如，同时使用 RMS 能量和音高轮廓）。为处理多个控制信号，TFG 会对每个控制项的损失取平均，并根据控制项设置权重，以平衡不同的损失尺度。请注意，整个框架也兼容无分类器引导（CFG） [25],，我们也在实验中使用了该方法。

2.3. 选择性TFG（我们的方案）

TFG 的设计空间中遗漏了引导的一个关键因素：究竟在哪些扩散步骤应用引导。近期研究表明，标准 CFG [25] 仅在有限区间的扩散步骤中应用，就能改善输出并降低计算负担 [26]。结合在采样过程的不同部分采用不同引导强度这一事实 [16],，这一思路自然可以扩展到 TFG 框架。由于由 $\mathbf{C}(\cdot)$ 定义的任何特定特征，都可能在采样过程的不同阶段产生感知效果，因此将引导限制在相应阶段，可以降低在可能导致输出偏离数据流形的区域应用引导的风险。仅在选定的采样步骤应用 TFG 引导，可以显著降低 TFG 带来的计算开销，同时在控制准确性与音频质量之间提供明确的权衡。例如，增加应用 TFG 的步骤数可以改善控制对齐，但会增加偏离流形的风险。相反，减少应用 TFG 的步骤数可以降低偏离流形的风险，但可能导致控制不够精确。我们将这一思路称为选择性 TFG（图 1）：在 TFG 中加入一组二元标量 $\{\tau_i\}_{i=1}^{T}$，用于标记哪些选定的步骤应用 TFG（T为采样步骤数）。

2.4. Latent-ControlHeads（LatCHs，我们的提案）

尽管人们已经对潜在扩散图像模型的端到端引导进行了探索，但其在潜在扩散音频模型中的发展 [22,21]仍相对有限——这可能是由于一个主要的实际问题：通过音频解码器 $\mathcal{D}$ 进行反向传播极其

昂贵。请注意，我们的变分自编码器以 21.5Hz（64 倍下采样率）将 47.55 秒、44.1kHz 的立体声音频编码为长度为 1024、通道数为 64 的潜变量。因此，每次端到端引导计算都必须通过 $\mathcal{D}$进行反向传播，这在计算上非常密集，不仅会显著增加推理延迟，还会增加显存占用，使其难以（甚至不可能）高效运行（图 1）。

为解决这一问题，我们探索训练潜变量控制头$\mathbf{c}_{\phi}(\mathbf{z}_0)$，使其无需经过解码器 $\mathcal{D}$ 即可预测控制特征，而是直接将潜变量映射为控制信号：

$$\mathbf{C}(\mathcal{D}(\mathbf{z}_0)) \approx \mathbf{c}_{\phi}(\mathbf{z}_0). \tag{4}$$

通过避免端到端的潜变量→音频→控制映射（图 1），直接学习潜变量→到控制信号的映射，只要公式 4 成立，引导计算就能快几个数量级，成本也低得多。

噪声条件训练 ——LatCHs 实际上是潜变量判别模型（即，“强度”LatCH 本质上只是一个针对潜变量的均方根回归器），但我们主要关注它们如何用于引导，而引导是针对逐步去噪的输出定义的。这导致训练与推理之间存在不匹配：训练时，我们直接将干净潜变量映射到目标特征；而推理时，LatCHs 的输入则是带噪潜变量。我们探索了两种解决方案：

- 前向模拟噪声条件（*LatCH-F*）：我们为 LatCHs 增加一个时间步输入$\mathbf{c}_{\phi}(\mathbf{z}^{0|t},\ t)$，通过 t 的傅里叶投影和序列拼接实现。我们训练噪声条件 LatCH，使其能够根据经前向扩散过程$\mathbf{z}_t$ 扰动的带噪潜变量预测控制信号。

- 反向模拟噪声条件（*LatCH-B*）：与前向噪声条件类似，我们为模型提供时间步输入。然而，我们不使用前向扩散后的（潜变量 +噪声）进行训练，而是从我们的潜在音频扩散模型生成轨迹 $\mathbf{z}_T$，$\mathbf{z}_{t-1}$，.....$\mathbf{z}_0$，并训练 LatCH 将这些配对的中间步骤映射到从生成输出中提取的特征。理论上，通过使用生成的中间步骤进行训练，而非使用前向扩散过程中模拟的步骤，这种方法与推理时所见的噪声分布完全一致。

3. 实验

我们在实验中使用 Stable Audio Open（SAO）[2]，这是一种文本条件潜空间音频扩散模型。实验中，我们选择以下特征提取器 $\mathbf{C}(\cdot)$：

- 强度：使用均方根（RMS）能量来建模响度动态（帧长 8193，跳跃大小 2048）。我们以分贝为单位对强度进行建模，并使用 Savitzky-Golay 滤波器 [12] 进行卷积，以减少高频振荡。

- 节拍：对于节拍对齐，我们使用 All-In-One 结构分析器 [27]。具体而言，我们使用其节拍跟踪模式（该模式预测随时间变化的二值节拍概率）。

- 音高：我们使用 CREPE [28], 提取单音音高，它为每个时间步预测一个音高概率向量。

距离函数 $\delta(\cdot)$ 设为二元交叉熵（BCE），但强度 使用均方误差（MSE）。所有提取器 $\mathbf{C}(\cdot)$ 均处理单声道音频。这样一来，SAO 现在可以根据特定的力度变化、音高和节拍来生成音频。

3.1. 基线

- *Stable Audio Open*（*SAO*） [2]是我们的音频质量基准，因为我们的目标是在不影响原始 SAO 音频质量的前提下引入控制功能。此外，由于 SAO 生成音频时不会遵循任何控制，我们还将其设为较低的随机 基准，用于评估控制对齐。

- 端到端 是一种计算密集型基线，类似于 [22], ，会带来解码的计算负担。它对于研究计算效率与音频质量之间的权衡具有参考意义（在与 LatCH 进行比较时）。

- 读出 [14] 与 LatCH 类似。我们将与它们进行比较，以评估使用扩散模型的中间层而非 VAE 潜变量是否具有任何优势。这正是读出与 LatCHs 之间的主要区别。

3.2. 数据集

我们使用 Free Music Archive 中的知识共享音乐训练 LatCHs，这些音乐也曾用于训练 SAO（13,874 条录音，970 小时）。此外，我们使用歌曲描述数据集 [30] 的非人声子集 [2, 29] 评估所研究的模型。最后，我们的评估需要目标控制信号，这些信号可以来自两个来源：(1) 直接从现有数据中提取控制信号，或 (2) 创建合成的用户自定义控制信号。在定量评估中，我们从歌曲描述数据集中提取控制信号 (1)；在定性评估中，我们则创建合成的用户自定义控制信号 (2)。

3.3. 选择性TFG、DDIM 和 CFG 超参数

鉴于 TFG 超参数调优颇具挑战，我们将列出所选取的数值，并说明选择这些数值时的考量。

- *LatCH*：$\rho = \mu = 0.03$、$\gamma = 0.3$，并且仅在最初 20% 的采样步骤中应用选择性 TFG。如果 ρ、μ 或所选 TFG 步骤的比例过高，就可能偏离流形。较小的 ρ、μ，以及仅在少数采样步骤中应用 TFG，可以确保输出质量不受影响（同时遵循控制信号）。使用多个控制信号时，我们将强度损失乘以 0.0005 进行加权。
- 端到端：$\rho = \mu = 0.03$, $\gamma = 1.5$，并且我们仅在前 20% 的采样步骤中应用选择性 TFG。我们发现，提高 γ至关重要，因为它会为了遵循控制而牺牲音频质量。在这种情况下，通过减弱 ρ, μ来降低引导强度，或减少所选步骤的百分比，都会降低控制跟随效果。因此，我们着重提高 γ，以降低引导的准确性。我们将强度 损失的权重设为 0.001。

- 读出：$\rho = 0.1$。读出使用潜在扩散模型的中间层 $\boldsymbol{h}_t$，即，$\tilde{\boldsymbol{e}}_t = \mathbf{r}(\boldsymbol{h}_t(\boldsymbol{z}_t))$。因此，均值引导项 $\nabla_{\mathbf{z}_{0|t}}\delta(\tilde{\boldsymbol{e}}_{0|t},\ \boldsymbol{e})$为零，因为$\mathbf{r}(\boldsymbol{h}_t(\boldsymbol{z}_t))$是 $\boldsymbol{z}_t$（而非$\boldsymbol{z}_{0|t}$）的函数。因此，读出不适用于均值引导、均值引导迭代，也不适用于将高斯噪声卷积到$\mathbf{z}^{0|t}$ 中；它们仅可用于方差引导。我们将强度损失的权重设为 0.005。

我们在整个过程中设置为 $N_{\text{iter}} = 4$ 和 $N_{\text{recur}} = 1$，与 [21] 中相同。通过实证我们发现，使用 SAO 训练时采用的方差保持形式，比其公开代码中的方差爆炸形式更加稳定，因此我们采用均匀噪声调度、100 步随机DDIM采样器和 7 的 CFG尺度。

我们未报告超参数消融结果，因为大幅调整超参数并不起作用，而小幅变化的影响可以忽略不计。

表1。 定性和定量结果。 ↓越低越好，或 ↑越高越好。

		质量指标			控制对齐指标 (↓)			计算成本 (↓)		定性结果 (平均意见分 ↑)		
控制项	方法	FD_{openl3} (↓)	KL_{passt} (↓)	CLAP (↑)	强度 MSE	音高 BCE	节拍 BCE	运行时间开启 图形处理器 (个)	VRAM 使用量（GB）	音频 质量	提示词 遵循度	控制 对齐
–	SAO [2]	96.51	0.55	0.41	32.91	0.070	0.351	11.3	5.51	–	–	–
节拍	LatCH-B	89.43	0.55	36.77	–	–	0.138	17.6	5.59	4.5	4.6	4.0
节拍	LatCH-F	101.24	0.7	33.49	–	–	0.161	17.6	5.59	4.1	4.1	2.9
节拍	读出	97.79	0.61	36.96	–	–	0.209	15.7	5.59	4.4	4.7	2.4
节拍	端到端	87.49	0.52	37.47	–	–	0.200	150.1	30.42	4.5	4.6	3.0
节拍+强度	LatCH-B	87.23	0.54	36.60	4.79		0.141	19.5	5.61	4.1	4.1	4.3 / 4.5
节拍+强度	LatCH-F	92.9	0.66	36.27	15.58	–	0.254	21.4	5.73	3.1	3.9	4.3 / 3.4
节拍+强度	读出	103.53	0.59	37.26	2.29	–	0.226	16.1	5.62	3.2	3.9	2.4 / 3.5
节拍+强度	端到端	86.50	0.54	37.99	5.67	–	0.200	240.0	32.24	4.4	4.8	4.6 / 4.3
音高	LatCH-B	106.96	0.51	36.05	–	0.028	–	17.7	5.65	3.3	3.9	4.3
音高	LatCH-F	105.57	0.59	35.82	–	0.041	–	17.7	5.65	3.6	3.7	3.3
音高	读出	107.45	0.54	37.08	–	0.038	–	15.7	5.66	3.5	3.5	1.3
音高	端到端	163.24	0.75	32.75	–	0.030	–	173.1	35.61	3.7	4.5	3.1
音高+强度	LatCH-B	125.7	0.49	35.46	3.35	0.030	–	19.5	5.69	3.1	4.0	4.7 / 3.8
音高+强度	LatCH-F	77.86	0.58	37.36	15.27	0.057	–	21.3	5.80	3.7	3.9	1.7 / 1.5
音高+强度	读出	104.45	0.53	38.34	1.59	0.047	–	16.1	5.69	3.5	3.8	3.1 / 3.5
音高+强度	端到端	147.64	0.65	33.95	1.76	0.033	–	261.1	37.23	3.8	4.1	3.5 / 4.3
强度	LatCH-B	77	0.54	39.23	2.52	–	–	17.5	5.56	4.7	4.7	4.5
强度	LatCH-F	69.9	0.55	38.62	15.35	–	–	17.5	5.56	4.1	3.9	2.2
强度	读出	89.81	0.56	38.55	1.38	–	–	15.6	5.57	4.7	4.5	4.7
强度	端到端	80.76	0.53	39.62	2.14	–	–	103.0	26.31	4.6	4.5	4.4

3.4. 训练 LatCHs

假设$\mathbf{C}$ 和 $\mathcal{E}$ 的输出具有相同的时间分辨率，LatCHs 是双向 Transformer，基于带有 RoPE [31],的 VAE 潜变量运行，随后通过投影层映射到特征的输出维度。因此，我们将提取的特征下采样或上采样到与潜变量相同的时间分辨率，因为控制信号无需以远高于 21.5Hz [2]的速率变化。LatCHs 包含 $\approx$ 7M 个参数（占基础生成模型参数的 $<$ 1% [2]），在单个 H100 GPU 上训练耗时 ≈ 4 小时，相较于条件训练提供了一种资源需求更低的控制方法。

一些特征提取器，例如，CREPE [28] 或 All-in-One [27], , 是神经网络模型。我们训练 LatCHs 来匹配它们的 logits，而不是使用硬标签目标。此外，提取器可能具有稀疏性，例如CREPE，它有 160 个音高分箱，每次只有 1 个分箱处于激活状态。在这种情况下，我们采用稀疏性感知损失：给定一个用于标记未激活特征的阈值，将阈值以下和以上的损失取平均，以提高稀疏特征的重要性。因此，我们采用以下训练方案：(1) 强度 *LatCHs* 使用均方误差对均方根进行训练；(2) 音高 *LatCHs* 使用稀疏加权二元交叉熵（阈值=0.2）对 CREPE logits 进行训练；(3) 节拍 *LatCHs* 使用二元交叉熵对 All-in-One logits 进行训练。

3.5. 定量评估

我们的评估基于公认的 [1] 质量指标，包括 FD_{openl3} [32]、KL_{passt} [33]和 CLAP [34], , 后者用于衡量文本-音频相似度。我们还根据距离函数$\delta(\cdot)$评估生成的音频与控制信号的匹配程度：强度使用 均方误差 ，音高和节拍使用 二元交叉熵 。最后，我们还评估了在 H100 GPU 上的运行时间和显存使用量。

3.6. 定性评估

我们通过听力测试对我们的方法进行定性评估。参与者（$n = 15$）被要求对示例的音频质量、提示遵循度以及控制对齐进行评分。我们在从1（差）到5（优秀）的5分制上报告平均意见分（平均意见分）。

4. 结果与讨论

接下来，我们讨论定性和定量结果（见表1）：

- LatCH-B 在音频质量、提示遵循度、控制对齐和效率方面表现最佳。
- 端到端方法也表现良好，但计算成本较高。
- 在我们的研究中，读出往往表现较差，可能是因为它缺少 TFG 中与“均值引导”相关的术语（第 3.4 节）。
- 对于节拍、强度、端到端方法和 LatCH-B，平均意见分的结果总体良好，质量指标与 SAO 相当。
- 多个引导控制也有效（例如，节拍+强度）。

我们发现，对于渐变或低频控制信号，所研究的方法更为可靠，例如强度 或节拍。变化更大的控制信号（例如音高，涉及快速的音符变化）会带来挑战，并且在各项质量指标上的表现都较差（见表1）。另请注意，强度 和节拍 由一维输出构成，而音高 输出 160 个音级。我们还尝试了 PANNs（527 个标签）和 Chroma（12 个音高）特征提取器，但结果不一。基于这些观察，我们推测，与稀疏的高维输出相比，一维输出可能更适合用于引导。

5. 结论

我们提出了一种面向潜在音频扩散模型的低资源推理时控制框架，将选择性TFG与轻量、可训练的潜在控制头（LatCHs）相结合。我们的方法能够在无需重新训练基础生成模型、也不承担端到端引导高计算成本的情况下实现控制。由于LatCHs直接在潜在空间中运行，因此无需像端到端引导那样通过解码器进行反向传播。

此外，我们引入了选择性TFG，它在少数选定的扩散步骤上策略性地施加引导，从而在保持生成质量的同时提高效率。我们的方法有效平衡了控制精度、音频保真度与运行性能，实现了长时程可引导音频合成（最长可达47.55秒）。我们的方法还可以同时使用多种控制信号和特征。

6. 参考文献

[1] Zach Evans、CJ Carr、Josiah Taylor、Scott H. Hawley 和 Jordi Pons，《快速时序条件潜在音频扩散》，*ICML*，2024年。[2] Zach Evans、Julian D Parker、CJ Carr、Zack Zukowski、Josiah Taylor 和 Jordi Pons，《Stable Audio Open》，*arXiv*，2024年。[3] Shih-Lun Wu、Chris Donahue、Shinji Watanabe 和 Nicholas J. Bryan，《Music ControlNet：用于音乐生成的多种时变控制信号》，*IEEE/ACM音频、语音和语言处理汇刊*，2024年。[4] Hugo Flores Garc´ıa、Oriol Nieto、Justin Salamon、Bryan Pardo 和 Prem Seetharaman，《Sketch2Sound：通过时变信号和声音模仿实现可控音频生成》，载于*ICASSP*，2025年。[5] Or Tal、Alon Ziv、Itai Gat、Felix Kreuk 和 Yossi Adi，《面向时间可控文本生成音乐的音频与符号联合条件》，*arXiv*，2024年。[6] Simon Rouard、Yossi Adi、Jade Copet、Axel Roebel 和 Alexandre D´efossez，《通过离散瓶颈特征进行音乐生成的音频条件》，*arXiv*，2024年。[7] Santiago Pascual、Chunghsin Yeh、Ioannis Tsiamas 和 Joan Serr`a，《具有增强同步性的掩码生成式视频到音频变换器》，载于*ECCV*，2024年。[8] Fang-Duo Tsai、Shih-Lun Wu、Haven Kim、Bo-Yu Chen、Hao-Chung Cheng 和 Yi-Hsuan Yang，《音频提示适配器：通过轻量级微调释放文本生成音乐的音乐编辑能力》，*arXiv*，2024年。[9] Julian D Parker、Janne Spijkervet、Katerina Kosta、Furkan Yesiler、Boris Kuznetsov、Ju-Chiang Wang、Matt Avent、Jitong Chen 和 Duc Le，《StemGen：聆听音乐的生成模型》，载于*ICASSP*，2024年。[10]Javier Nistal、Marco Pasini、Cyran Aouameur、Maarten Grachten 和 Stefan Lattner，《Diff-a-Riff：通过潜扩散模型共同创作音乐伴奏》，*arXiv*，2024年。[11] Amir Hertz、Ron Mokady、Jay Tenenbaum、Kfir Aberman、Yael Pritch 和 Daniel Cohen-Or，《使用交叉注意力控制进行 Prompt-to-Prompt 图像编辑》，*arXiv*，2022年。[12] Zachary Novack、Julian McAuley、Taylor Berg-Kirkpatrick 和 Nicholas J. Bryan，《DITTO：面向音乐生成的扩散推理时 T 优化》，载于*ICML*，2024年。[13] Zachary Novack、Julian McAuley、Taylor Berg-Kirkpatrick 和 Nicholas Bryan，《Ditto-2：面向音乐生成的蒸馏扩散推理时 T 优化》，*arXiv*，2024年。[14] Grace Luo、Trevor Darrell、Oliver Wang、Dan B Goldman 和 Aleksander Holynski，《读出引导：从扩散特征中学习控制》，*CVPR*，2024年。[15] Prafulla Dhariwal 和 Alexander Nichol，《扩散模型在图像合成方面胜过生成对抗网络》，*NeurIPS*，2021年。

[16] Jiwen Yu、Yinhuai Wang、Chen Zhao、Bernard Ghanem 和 Jian Zhang，《FreeDoM：免训练能量引导的条件扩散模型》，国际计算机视觉大会，2023年。[17] Hyungjin Chung、Jeongsol Kim、Michael T. McCann、Marc Louis Klasky 和 J. C. Ye，《用于一般含噪逆问题的扩散后验采样》，载于国际学习表征会议，2023年。[18] Yutong He、Naoki Murata、Chieh-Hsin Lai、Yuhta Takida、Toshimitsu Uesaka、Dongjun Kim、Wei-Hsiang Liao、Yuki Mitsufuji、J Zico Kolter、Ruslan Salakhutdinov 和 Stefano Ermon，《流形保持引导扩散》，载于国际学习表征会议，2024年。[19] Jiaming Song、Qinsheng Zhang、Hongxu Yin、Morteza Mar- dani、Ming-Yu Liu、Jan Kautz、Yongxin Chen 和 Arash Vah- dat，《用于即插即用可控生成的损失引导扩散模型》，载于*ICML*，2023年。[20] Arpit Bansal、Hong-Min Chu、Avi Schwarzschild、Soumyadip Sengupta、Micah Goldblum、Jonas Geiping 和 Tom Gold- stein，《扩散模型的通用引导》，载于*CVPR*，2023年。[21] Haotian Ye、Haowei Lin、Jiaqi Han、Minkai Xu、Sheng Liu、Yitao Liang、Jianzhu Ma、James Zou 和 Stefano Ermon，《TFG：扩散模型的统一免训练引导》，载于*NeurIPS*，2024年。[22]Mark Levy、Bruno Di Giorgi、Floris Weers、Angelos Katharopoulos 和 Tom Nickson，《利用扩散模型和引导梯度实现可控音乐制作》，载于*NeurIPS* 扩散模型研讨会，2023年。[23] Brian DO Anderson，《反向时间扩散方程模型》，《随机过程及其应用》*Stochastic Processes and their Applications*，第12卷第3期，第313–326页，1982年。[24] Jiaming Song、Chenlin Meng 和 Stefano Ermon，《去噪扩散隐式模型》，载于国际学习表征会议，2020年。[25] Jonathan Ho 和 Tim Salimans，《无分类器扩散引导》，*arXiv*，2022年。[26] Tuomas Kynk¨a¨anniemi、Miika Aittala、Tero Karras、Samuli Laine、Timo Aila 和 Jaakko Lehtinen，《在有限区间内应用引导可提升扩散模型中的样本质量和分布质量》，*arXiv*，2024年。[27]Taejun Kim 和 Juhan Nam，《利用分离音频上的邻域注意力进行一体化韵律与功能结构分析》，载于音频与声学信号处理研讨会，2023年。[28] Jong Wook Kim、Justin Salamon、Peter Li 和 Juan Pablo Bello，《CREPE：用于音高估计的卷积表征》，载于*ICASSP*，2018年。[29] Zach Evans、Julian Parker、CJ Carr、Zack Zukowski、Josiah Taylor 和 Jordi Pons，《利用潜在扩散进行长篇音乐生成》，*arXiv*，2024年。[30] Ilaria Manco、Benno Weck、Seungheon Doh、Minz Won、Yix- iao Zhang、Dmitry Bodganov、Yusong Wu、Ke Chen、Philip Tovstogan、Emmanouil Benetos 等，《歌曲描述数据集：用于音乐与语言评估的音频描述语料库》，*arXiv*，2023年。[31] Jianlin Su、Murtadha Ahmed、Yu Lu、Shengfeng Pan、Wen Bo 和 Yunfeng Liu，《Roformer：采用旋转位置嵌入的增强型变换器》，《神经计算》*Neurocomputing*，2024年。[32] Aurora Linh Cramer、Ho-Hsiang Wu、Justin Salamon 和 Juan Pablo Bello，《看、听，并学习更多：深度音频嵌入的设计选择》，载于*ICASSP*，2019年。

[33] Khaled Koutini、Jan Schl¨uter、Hamid Eghbal-Zadeh 和 Ger- hard Widmer，《利用 Patchout 高效训练音频变换器》，*arXiv*，2021。[34] Yusong Wu、Ke Chen、Tianyu Zhang、Yuchen Hui、Taylor Berg-Kirkpatrick 和 Shlomo Dubnov，《通过特征融合和关键词到字幕增强进行大规模对比式语言-音频预训练》，载于*ICASSP*，2023年。