MGE-LDM: Joint Latent Diffusion for Simultaneous Music Generation and Source Extraction

Yunkee chae and Kyogu Lee. Music and Audio Research Group (MARG), Seoul National University
{yunkimo95, kglee}@snu.ac.kr

Audio Samples

TL;DR

We propose a joint latent diffusion model that unifies **music generation, source imputation, and text-guided source extraction** within a single framework

Our Idea / Contributions

- Jointly models **mixture, submixture, and source** tracks in a shared latent space, so all tasks are cast as conditional inpainting.
- Performs **total generation / source imputation / text-guided source extraction** with a more **flexible, joint multi-track design.**
- Uses **track-dependent diffusion timesteps** to better learn conditional scores for partially observed multi-track inputs
- **Class- and dataset-agnostic:** no fixed instrument labels are required, so we can train on any combination of multi-track music datasets.

Motivation

- A recent line of work has focused on simultaneous music generation and separation [1,2]
- However, prior methods typically assume a fixed set of four stem classes: drums, bass, guitar, and piano.
- In addition, some works rely on a waveform additivity assumption, which does not hold well in latent space.

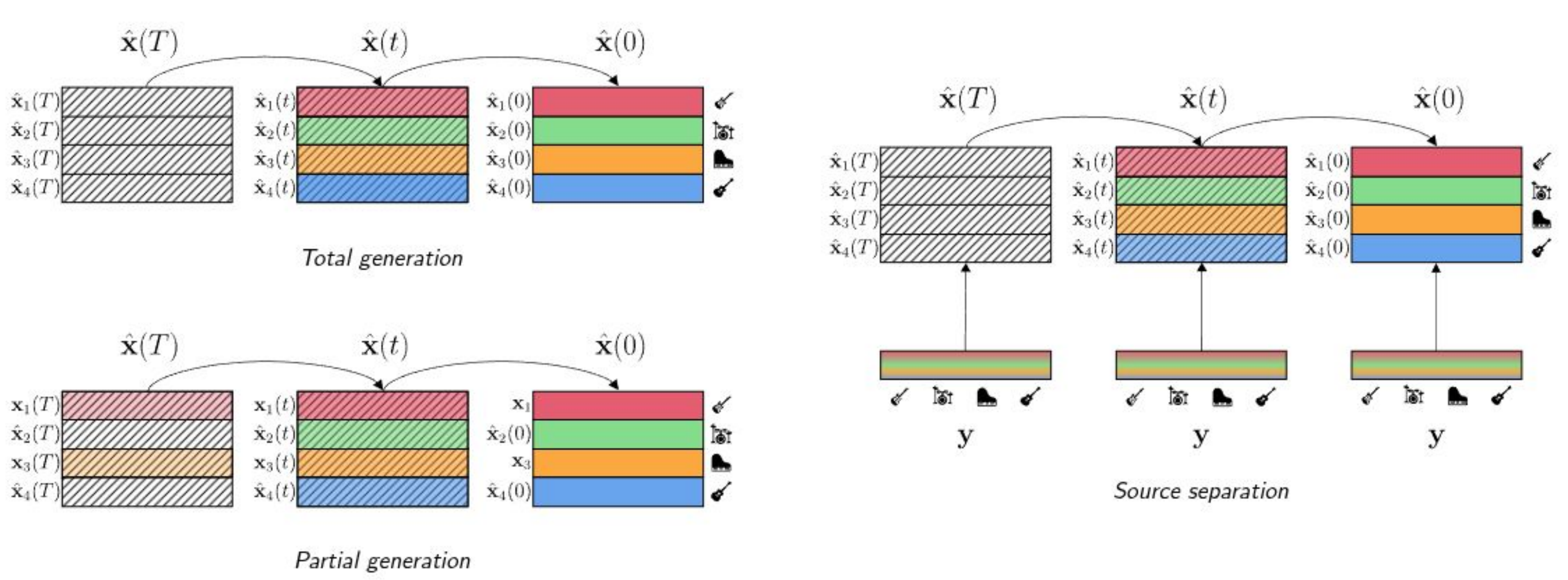


Fig. 1: Multi-track modeling approach of Mariani et al. [1], capable of handling three tasks within a single model. The diffusion model operates directly in waveform domain.

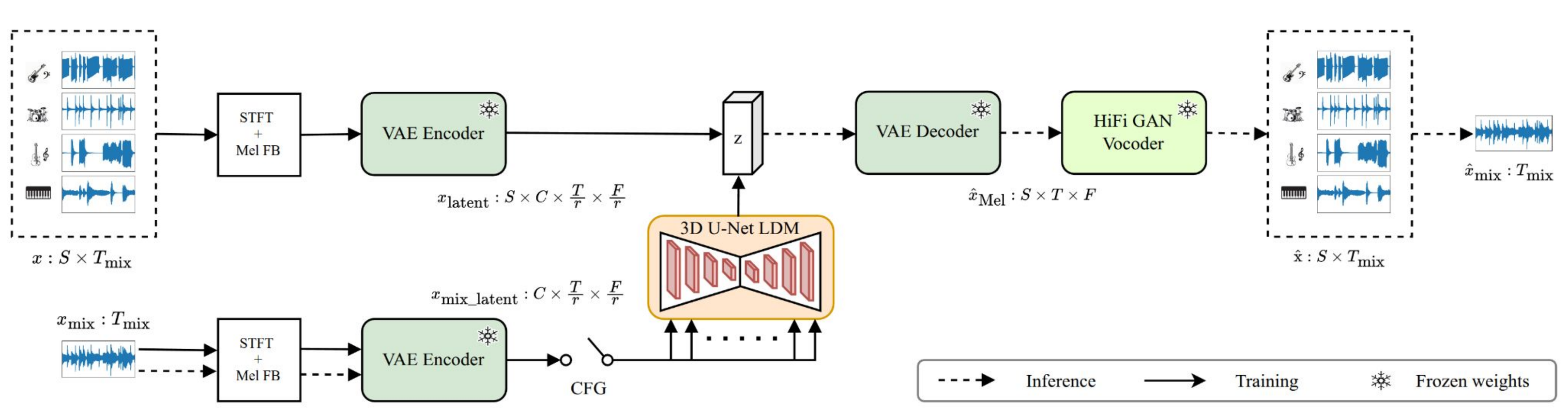
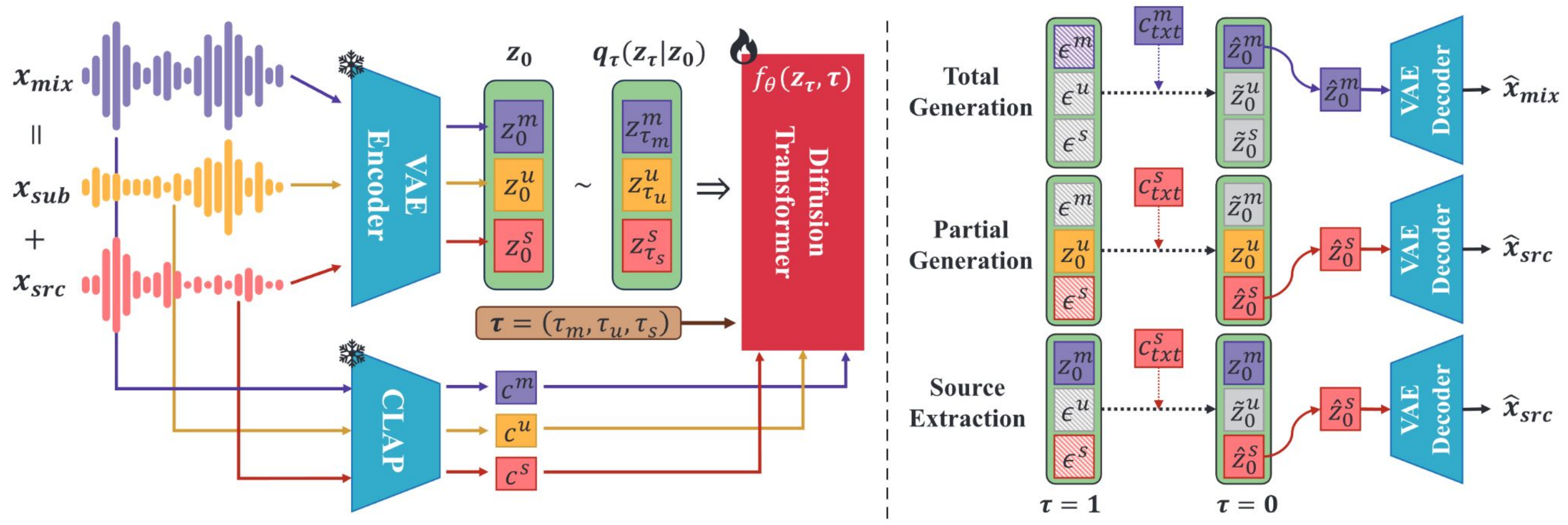


Fig. 2: Work by Karchkhadze et al. [2], which performs the same tasks as [1] using latent diffusion model.

How It Works



(a) Training pipeline (b) Inference pipeline

Joint Latent Diffusion Training

- **Triplet data: mix, submix, source**

$$\mathbf{z}_0 = \left(z_0^{(m)}, z_0^{(u)}, z_0^{(s)}\right) \in \mathbb{R}^{3\times C\times L}$$

- **v-objective diffusion**

$$\mathbf{z}_\tau = \alpha_\tau \mathbf{z}_0 + \beta_\tau \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{z}_0,\boldsymbol{\epsilon},\tau} \|f_\theta(\mathbf{z}_\tau, \tau, \mathbf{c}) - \boldsymbol{v}_\tau\|_2^2, \quad \boldsymbol{v}_\tau = \frac{\partial \mathbf{z}_\tau}{\partial \phi_\tau} = \alpha_\tau \boldsymbol{\epsilon} - \beta_\tau \mathbf{z}_0$$

Inference via Conditional Sampling

- **Total Generation**

$$\boxed{\tilde{z}^{(m)}}, \tilde{z}^{(u)}, \tilde{z}^{(s)} \sim p_\theta(z^{(m)}, z^{(u)}, z^{(s)} | c^{(m)}, \varnothing, \varnothing)$$

- **Partial Generation**

$$\tilde{z}_j^{(m)}, \boxed{\tilde{z}_j^{(s)}} \sim p_\theta(z^{(m)}, z^{(s)} | z_{j-1}^{(u)}, \varnothing, \varnothing, c_j^{(s)})$$

- **Source Extraction**

$$\tilde{z}^{(u)}, \boxed{\tilde{z}^{(s)}} \sim p_\theta(z^{(u)}, z^{(s)} \mid z^{(m)}, \varnothing, \varnothing, c^{(s)})$$

Track-aware Inpainting with Adaptive Timesteps

- Instead of applying the same diffusion timestep to all tracks, we assign **track-specific timesteps**. This approach improves conditional score learning for partially observed inputs.
- Inspired by TD-Paint [3], we compute **per-track timestep vectors** and adjust the corruption process and v-objectives accordingly:

$$\mathbf{z}_{\boldsymbol{\tau}} = \boldsymbol{\alpha}_{\boldsymbol{\tau}} \odot \mathbf{z}_0 + \boldsymbol{\beta}_{\boldsymbol{\tau}} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\boldsymbol{v}_{\boldsymbol{\tau}} = \boldsymbol{\alpha}_{\boldsymbol{\tau}} \odot \boldsymbol{\epsilon} - \boldsymbol{\beta}_{\boldsymbol{\tau}} \odot \mathbf{z}_0$$

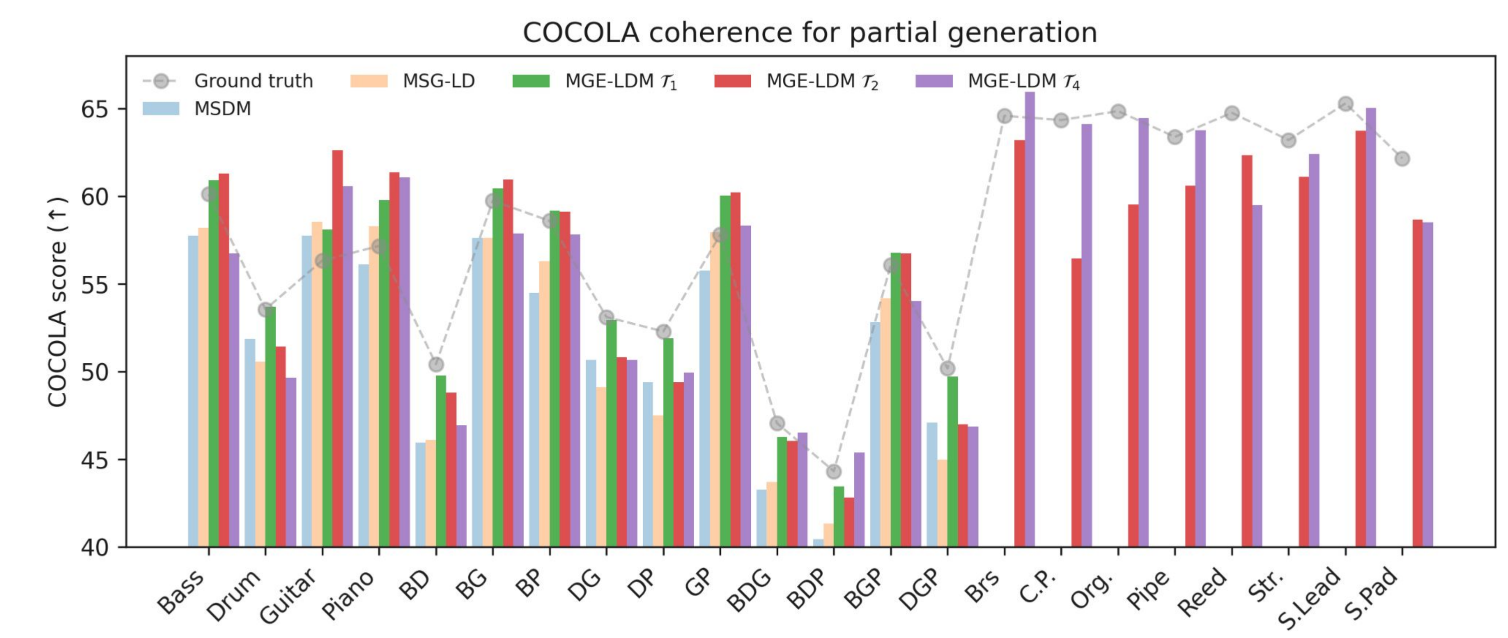
$$\boldsymbol{\tau} \in \{(\tau,\tau,\tau), (0,\tau,\tau), (\tau,0,\tau), (\tau,\tau,0)\}$$

Results

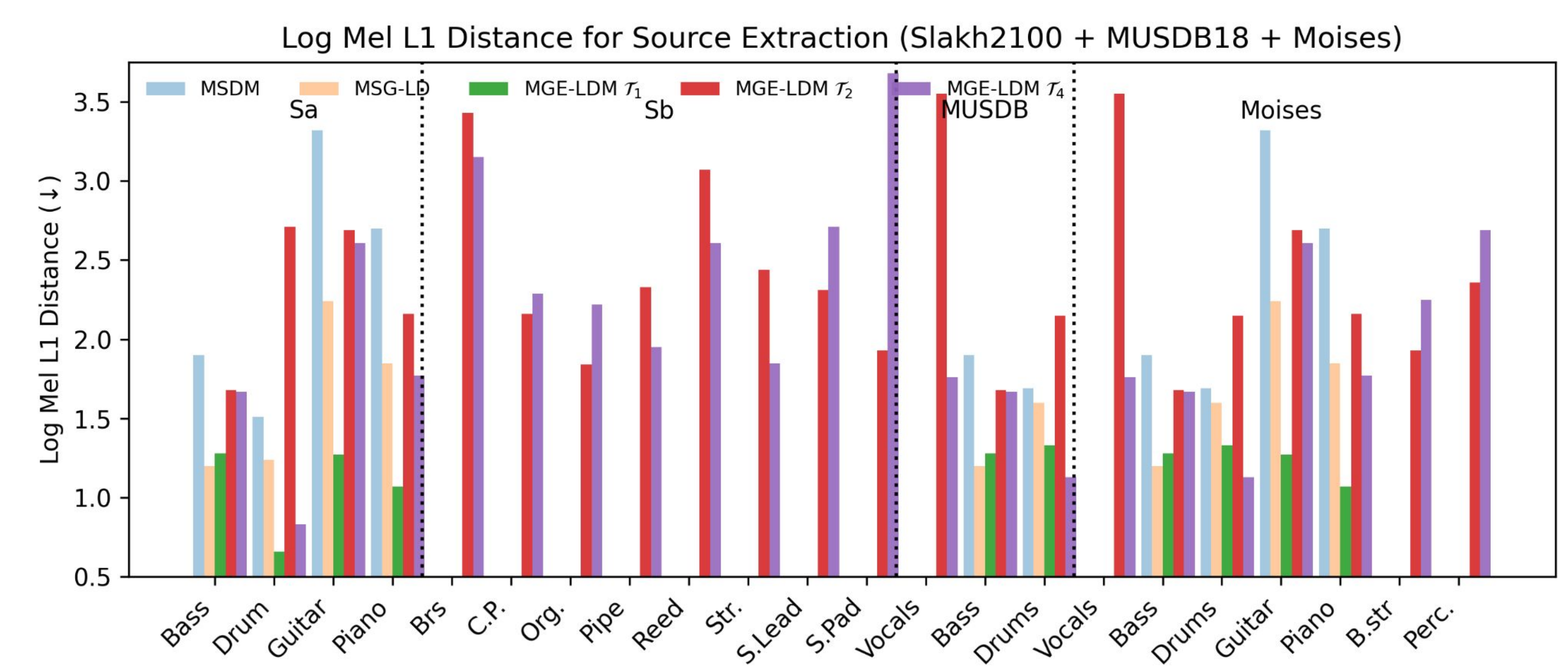
Total Generation

Model		Train Set $\mathbf{S}_A$	Train Set $\mathbf{S}_B$	Train Set $\mathbf{M}_u$	Train Set $\mathbf{M}_o$	Test Set $\mathbf{S}_A$	Test Set $\mathbf{S}_{\text{Full}}$	Test Set $\mathbf{M}_u$	Test Set $\mathbf{M}_o$
MSDM		✓	×	×	×	4.21	6.04	7.92	7.41
MSG-LD		✓	×	×	×	1.38	<u>1.55</u>	<u>4.61</u>	<u>4.26</u>
MGE (ours)	$\mathcal{T}_1$	✓	×	×	×	<u>**0.47**</u> (3.57)	1.79	6.34	5.90
	$\mathcal{T}_2$	✓	✓	×	×	3.14 (2.24)	**0.63**	5.46	4.73
	$\mathcal{T}_3$	×	×	✓	✓	8.80 (3.96)	6.56	2.87	1.59
	$\mathcal{T}_4$	✓	✓	✓	✓	6.83 (5.05)	4.22	**2.78**	**1.47**

Partial Generation



Source Extraction



Adaptive Timestep Ablation (Source Extraction)