PAT 498/598 (Fall 2024)

Special Topics:
Generative AI for Music and Audio Creation

Lecture 18: Neural Audio Effects & Auto Mixing

Instructor: Hao-Wen Dong

Final Project

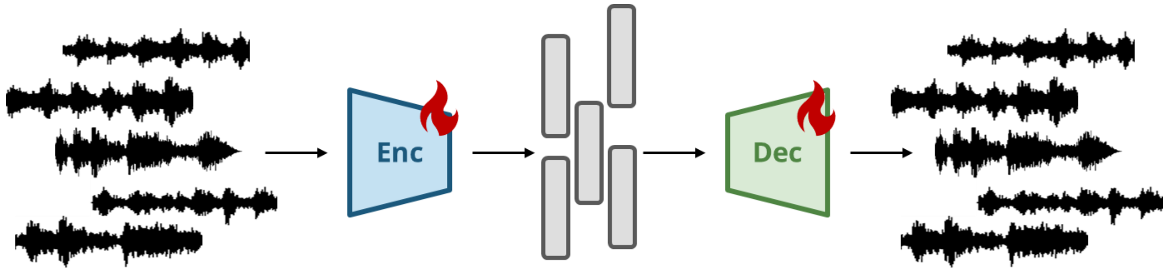
- Milestones (all due at the specified date at **11:59 PM ET**)

Pitch	November 6	Topic & high-level plans
Proposal	November 18	Survey & plans (1 page)
Presentation	December 9	Showcase & report
Final report	December 15	Full report (3-5 pages)

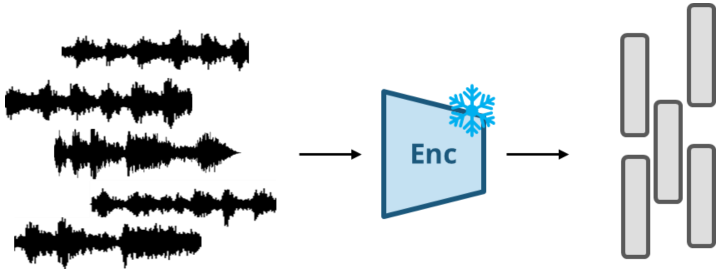
- Instructions will be released on Gradescope
- Late submissions: **NOT accepted**

(Recap) Pipeline

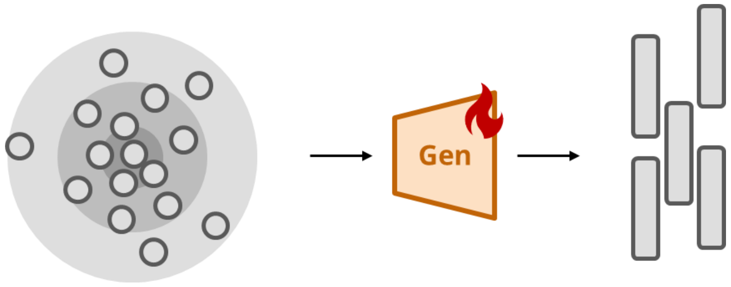
Step 1: Train an Autoencoder



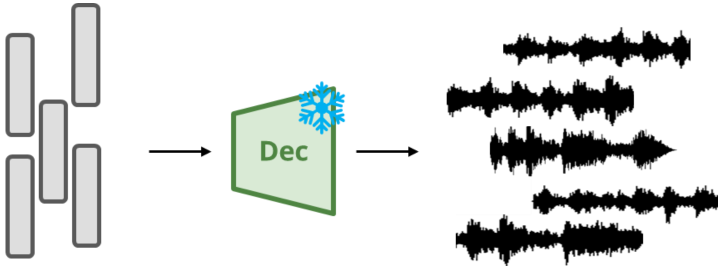
Step 2: Compute the Latent Vectors



Step 3: Train a Latent Generative Model

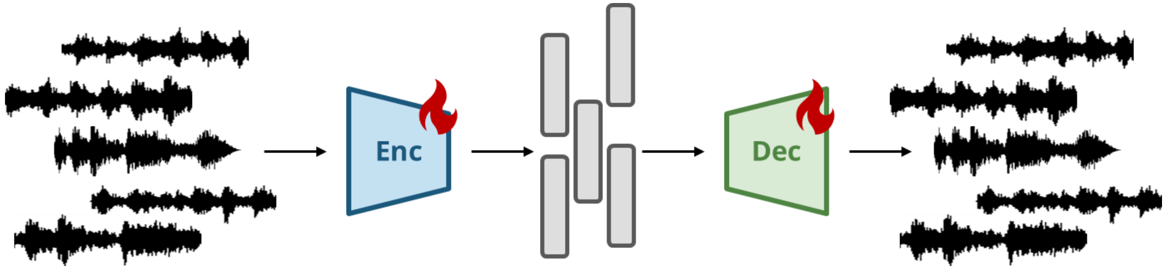


Step 4: Decode the Latent Vectors

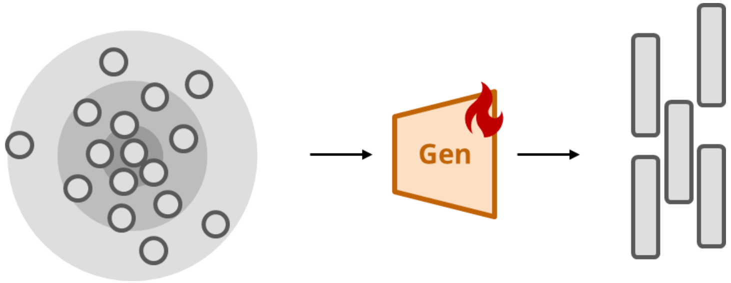


(Recap) Training

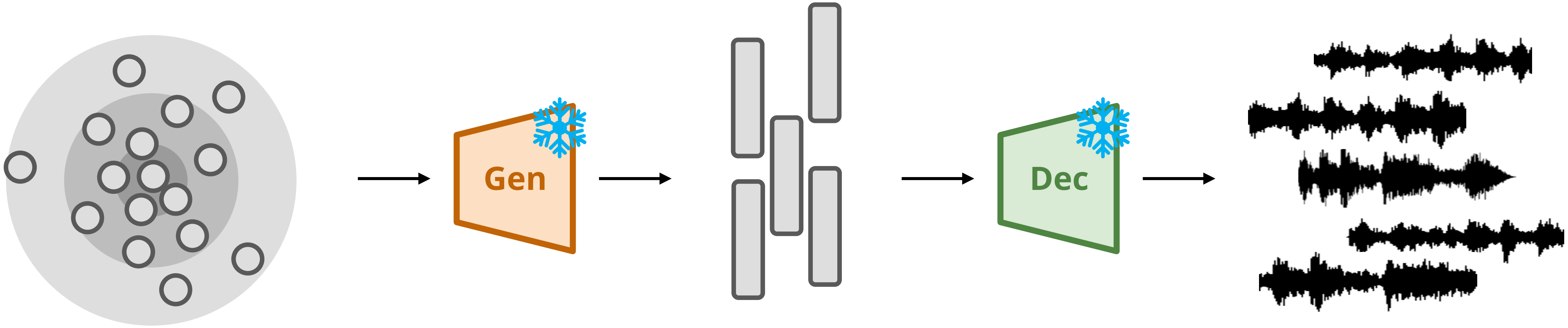
Autoencoder



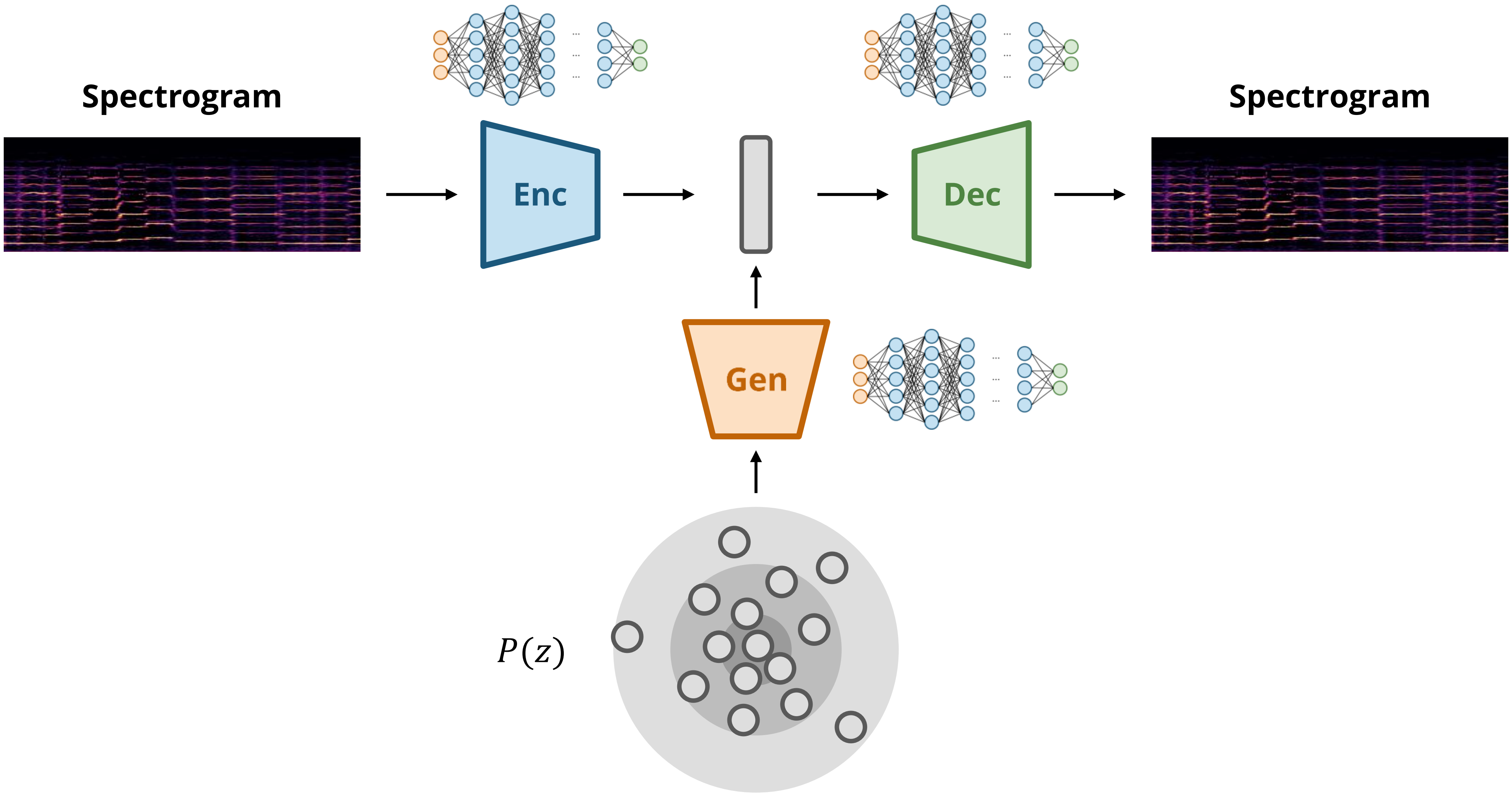
Latent Generative Model



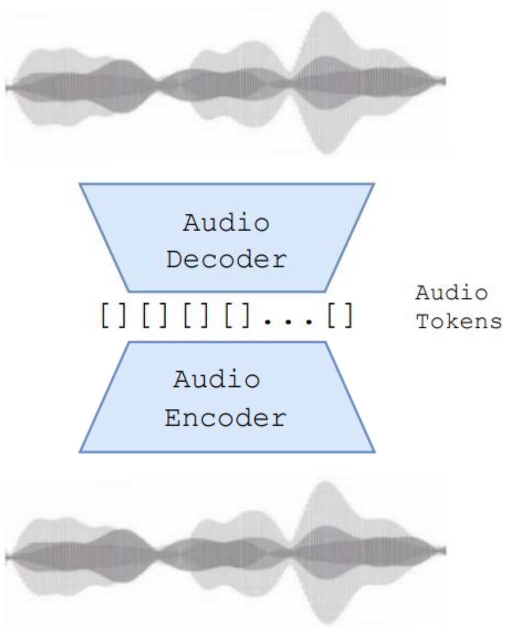
(Recap) Inference



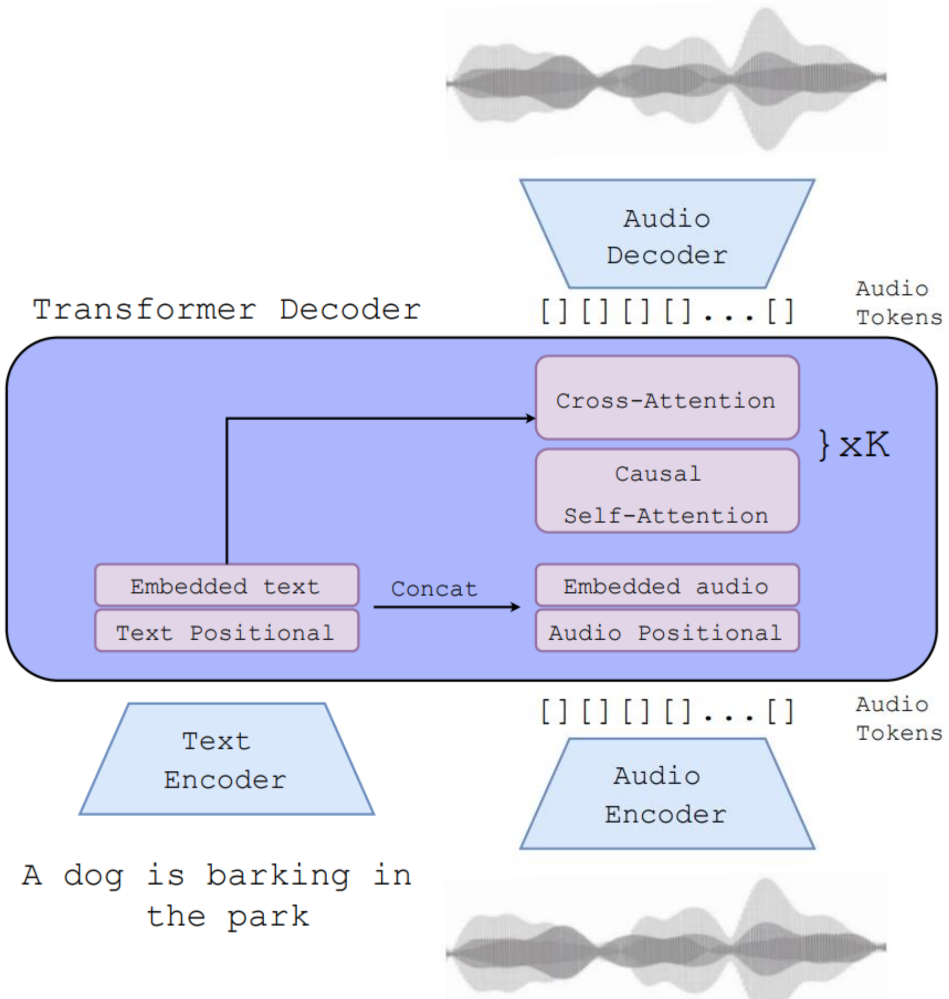
(Recap) Latent-based Audio Synthesis



(Recap) Example: AudioGen (Kreuk et al., 2023)



4k hours
(speech, music, sound effects)

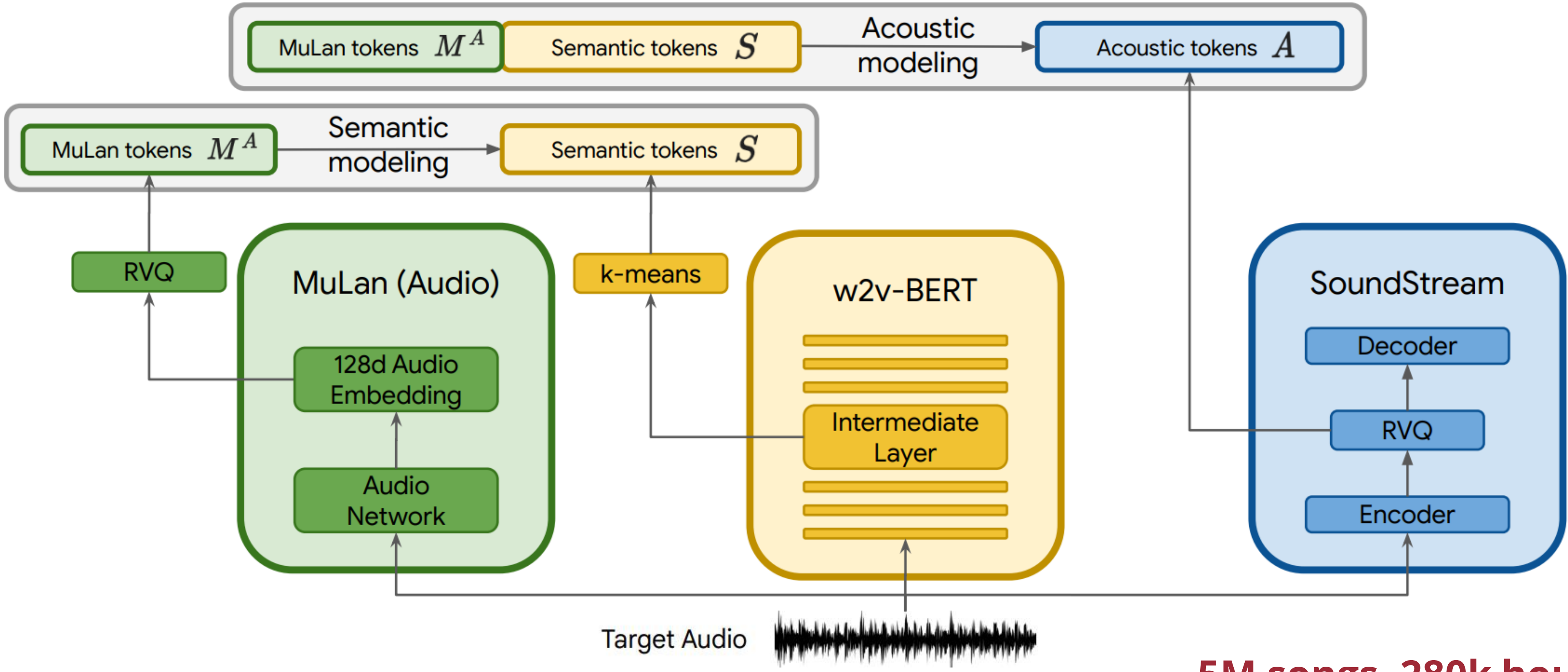


(Source: Kreuk et al., 2022)

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi, "AudioGen: Textually Guided Audio Generation," *ICLR*, 2023.

(Recap) Example: MusicLM (Agostinelli et al., 2023)

Training



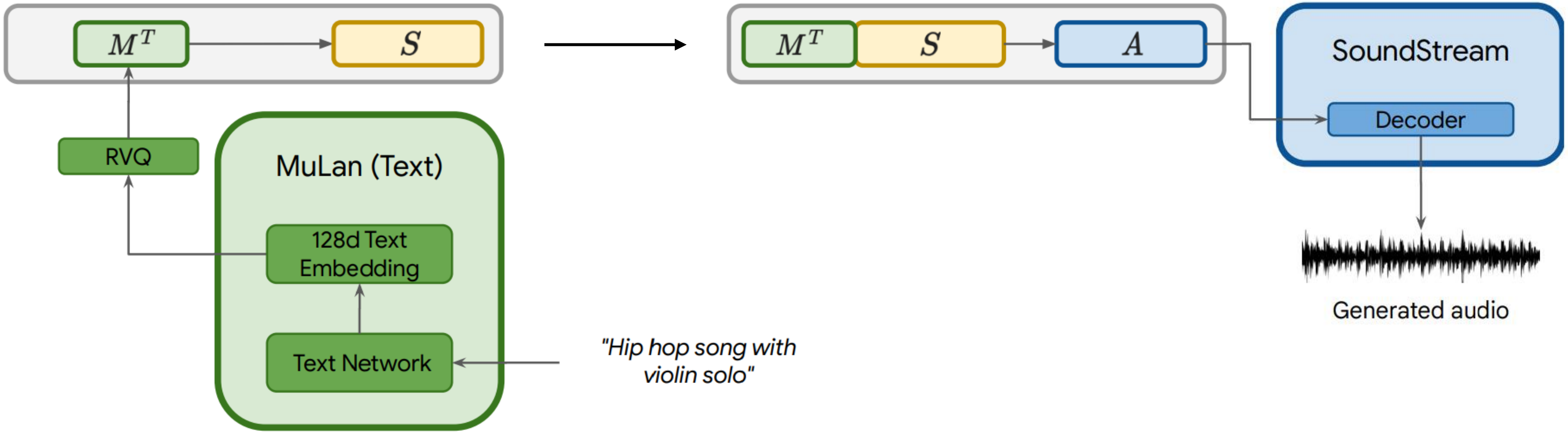
5M songs, 280k hours

(Source: Agostinelli et al., 2022)

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank, "MusicLM: Generating Music From Text," *arXiv preprint arXiv:2301.11325*, 2023.

(Recap) Example: MusicLM (Agostinelli et al., 2023)

Inference

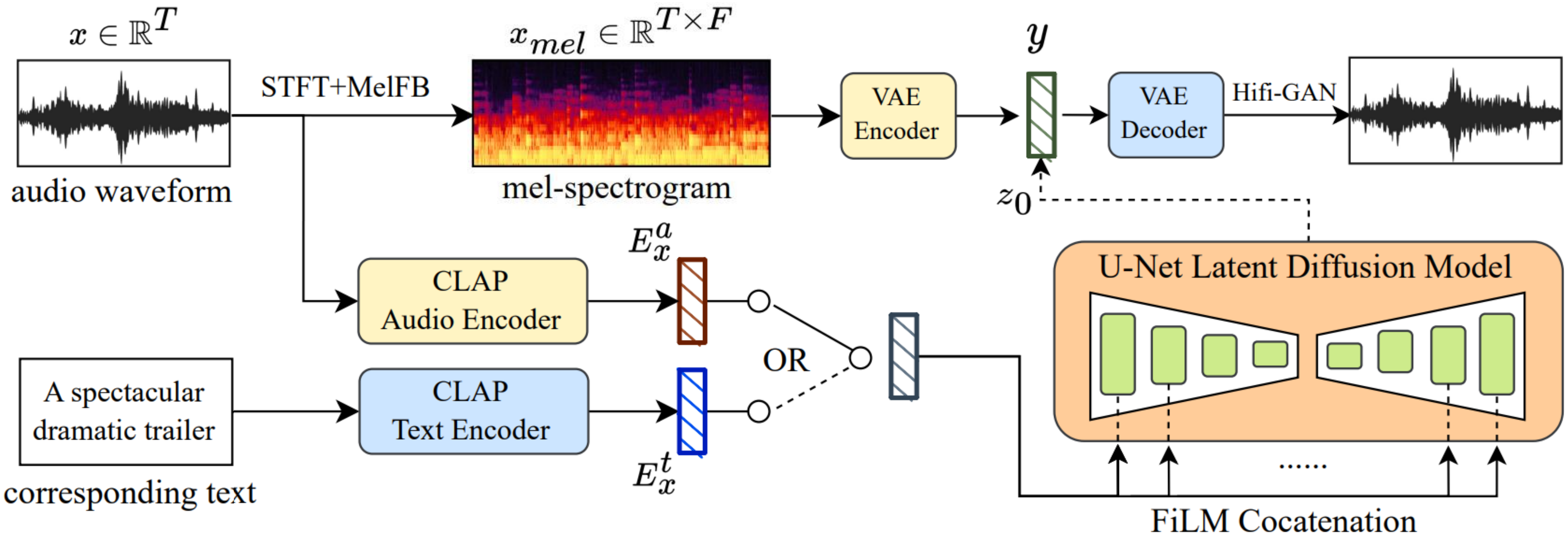


(Source: Agostinelli et al., 2022)

google-research.github.io/seanet/musiclm/examples/

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank, "MusicLM: Generating Music From Text," *arXiv preprint arXiv:2301.11325*, 2023.

(Recap) Example: MusicLDM (Chen et al., 2023)

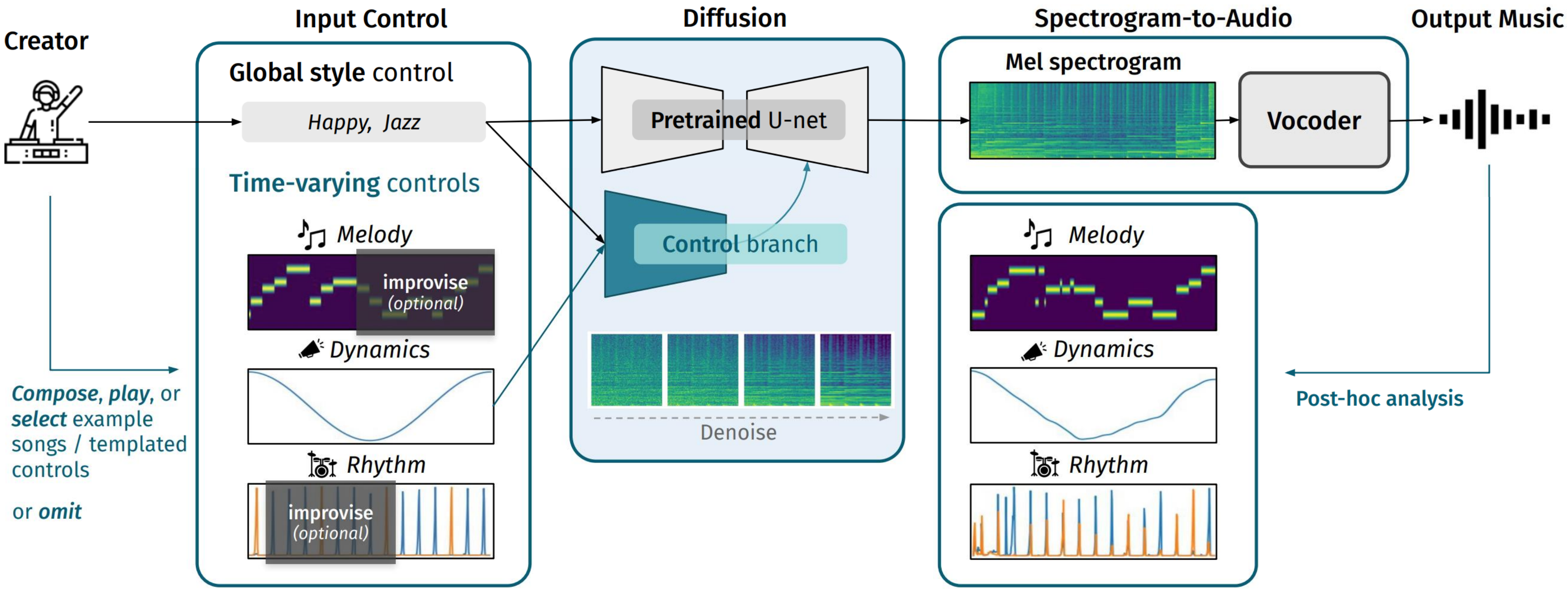


(Source: Ke et al., 2023)

musicldm.github.io

Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "MusicLDM: Enhancing Novelty in Text-to-Music Generation Using Beat-Synchronous Mixup Strategies," *ICASSP*, 2024.

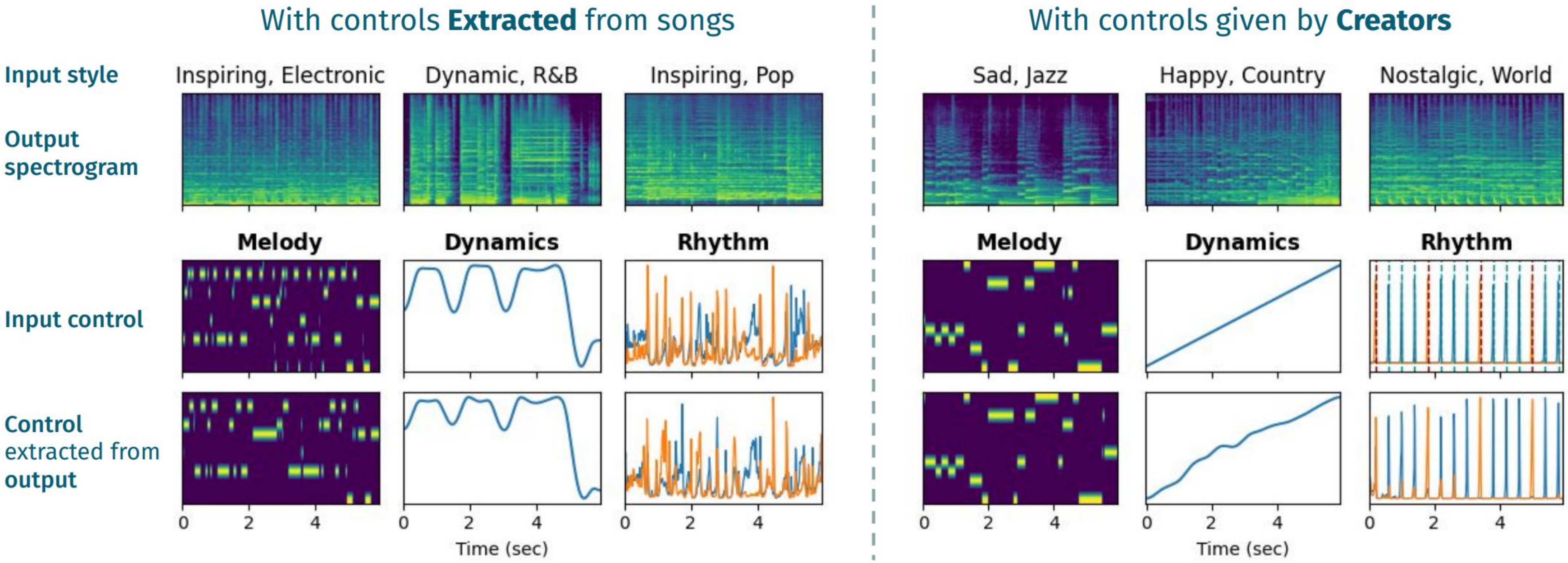
(Recap) Example: Music ControlNet (Wu et al., 2024)



(Source: Wu et al., 2024)

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan, "Music ControlNet: Multiple Time-varying Controls for Music Generation," *TASLP*, 2024.

(Recap) Example: Music ControlNet (Wu et al., 2024)

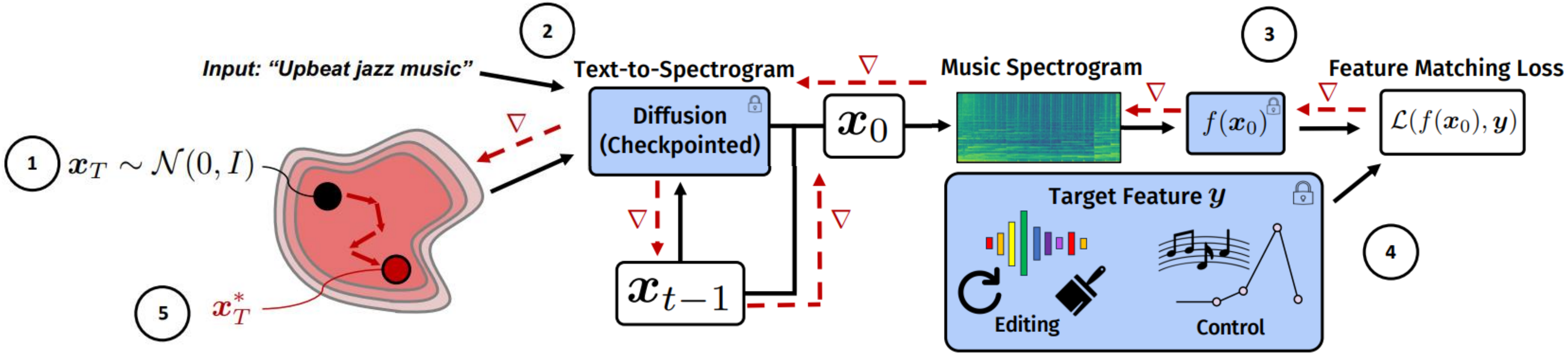


(Source: Wu et al., 2024)

musiccontrolnet.github.io/web

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan, "Music ControlNet: Multiple Time-varying Controls for Music Generation," *TASLP*, 2024.

(Recap) Example: DITTO (Novack et al., 2024)

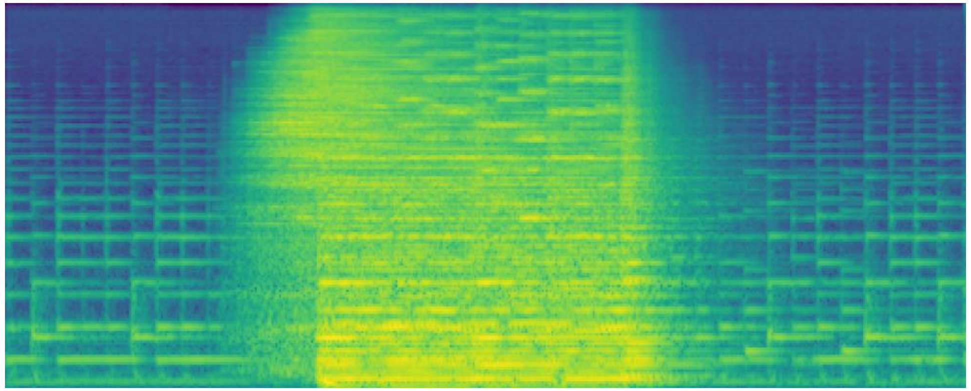


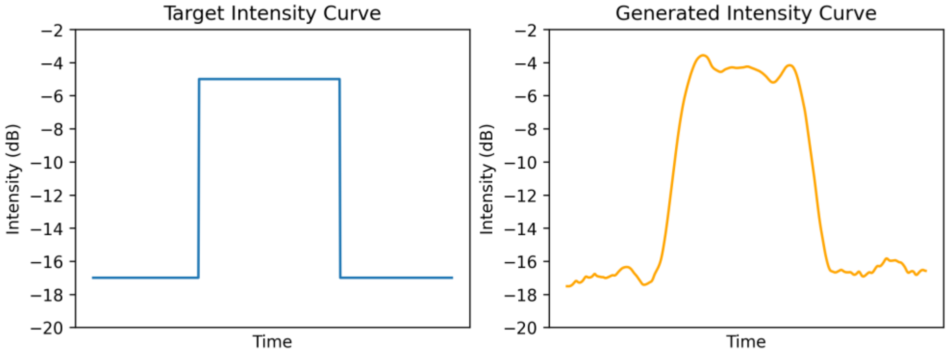
(Source: Novack et al., 2024)

Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan, "DITTO: Diffusion Inference-Time T-Optimization for Music Generation," *ICML*, 2024.

(Recap) Example: DITTO (Novack et al., 2024)

Intensity control

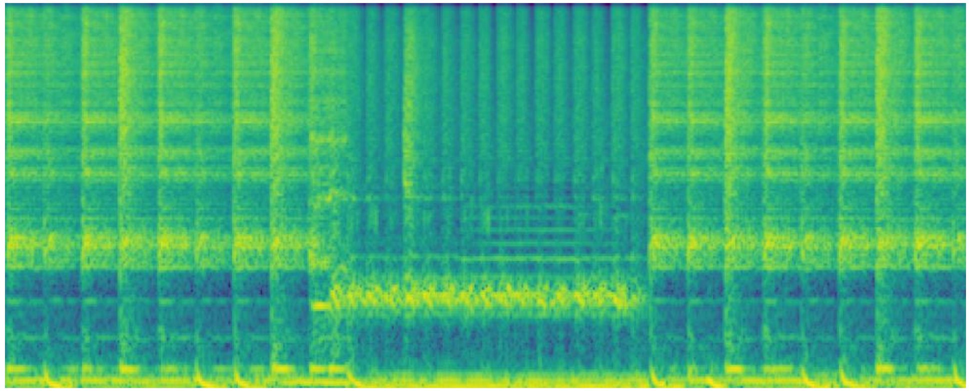


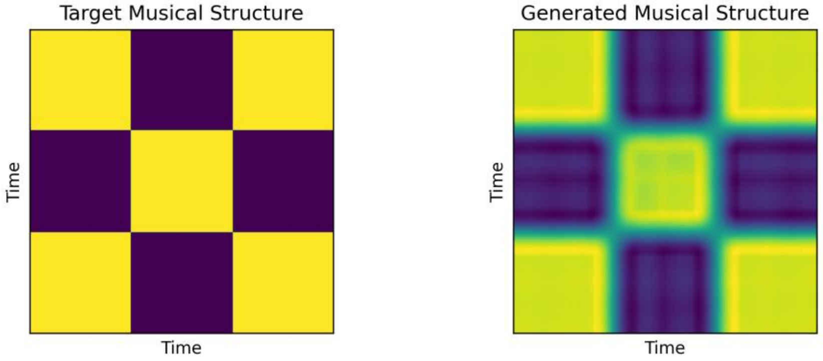


Target Intensity Curve

Generated Intensity Curve

Structure control





Target Musical Structure

Generated Musical Structure

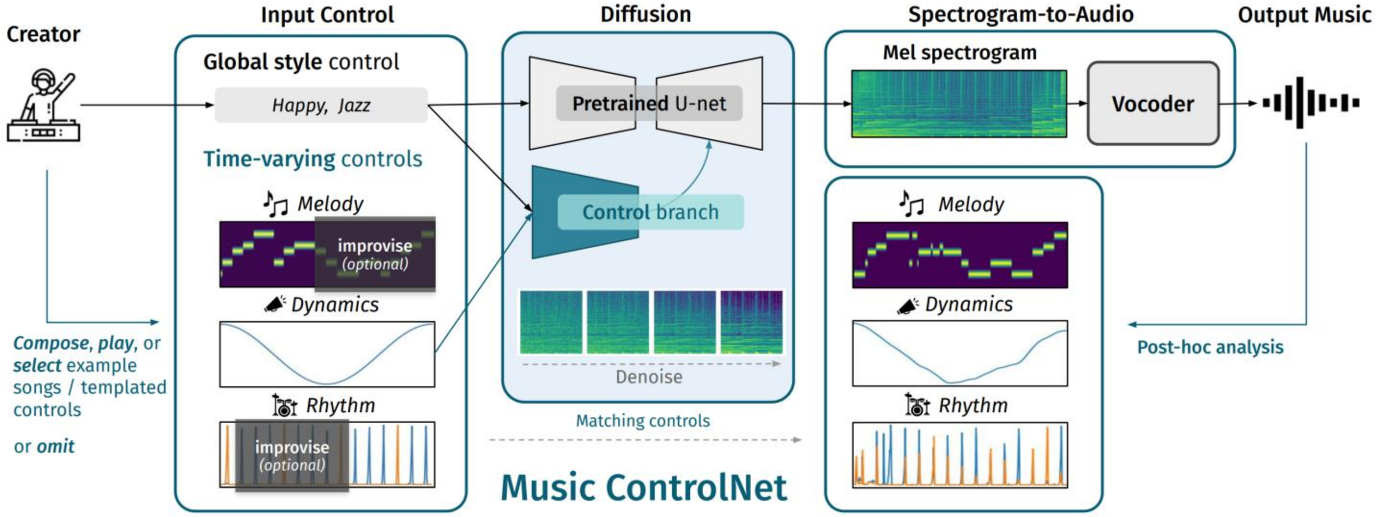
(Source: Novack et al., 2024)

Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan, “DITTO: Diffusion Inference-Time T-Optimization for Music Generation,” *ICML*, 2024.

(Recap) Music ControlNet vs DITTO

Music ControlNet

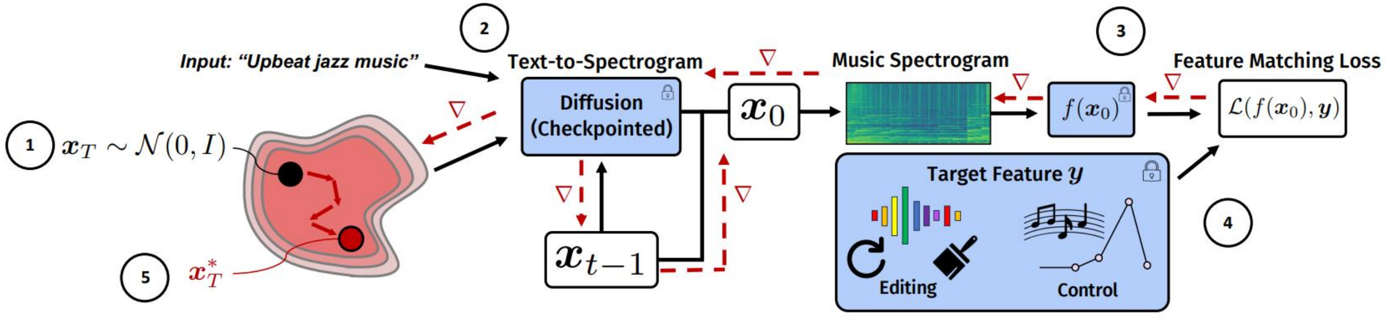
Needs some training!



(Source: Wu et al., 2024)

DITTO

No training needed!

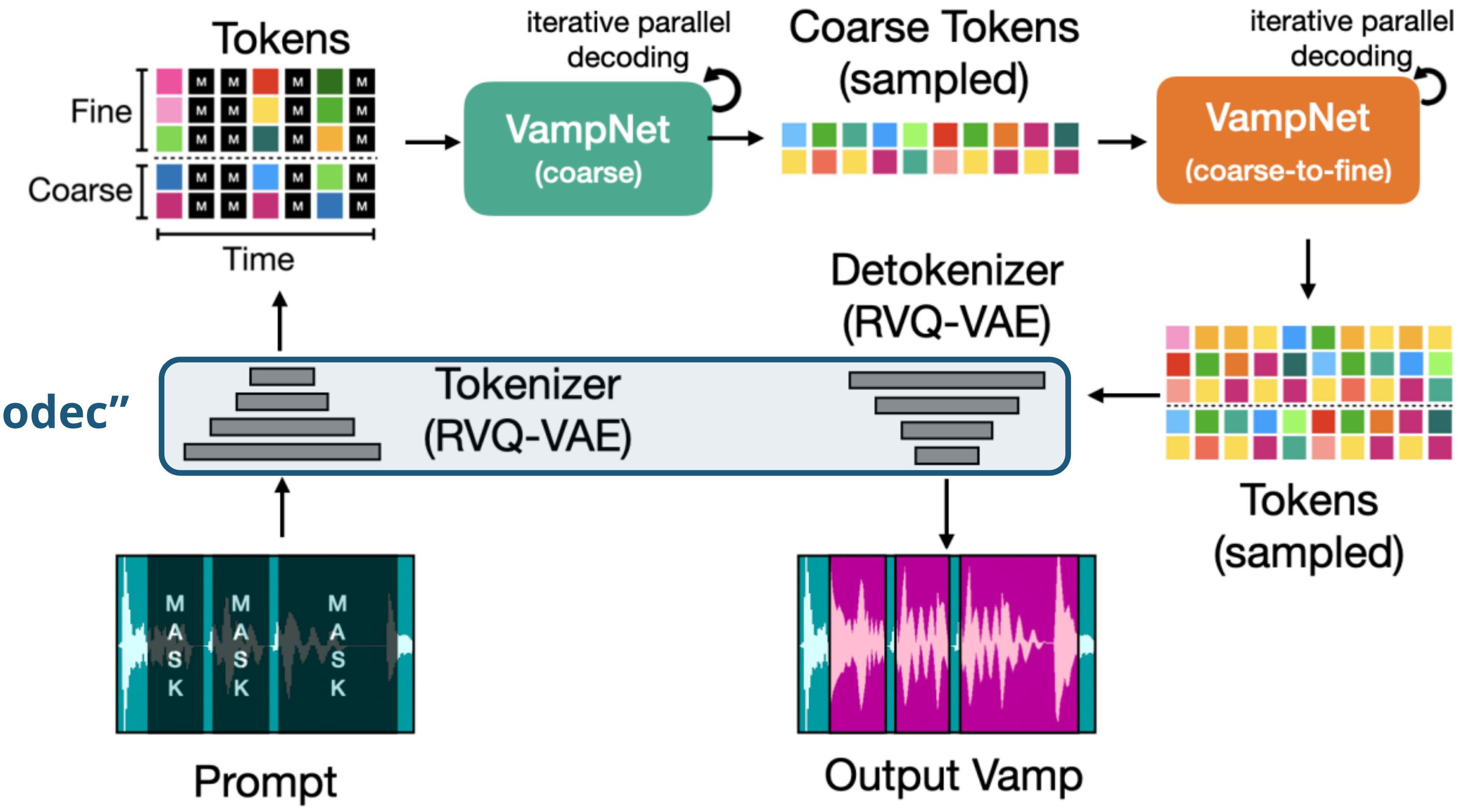


(Source: Novack et al., 2024)

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J. Bryan, "Music ControlNet: Multiple Time-varying Controls for Music Generation," *TASLP*, 2024.
Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J. Bryan, "DITTO: Diffusion Inference-Time T-Optimization for Music Generation," *ICML*, 2024.

(Recap) Example: VampNet (Garcia et al., 2023)

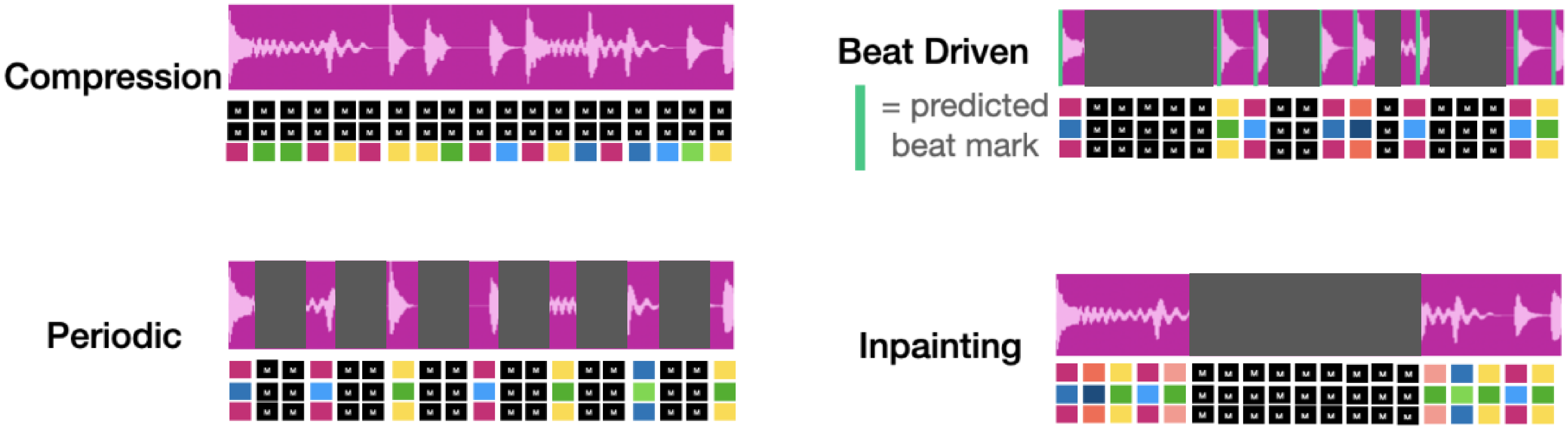
Neural "codec"



(Source: Garcia et al., 2023)

Hugo Flores Garcia, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo, "VampNet: Music Generation via Masked Acoustic Token Modeling," *ISMIR*, 2023.

(Recap) Example: VampNet (Garcia et al., 2023)

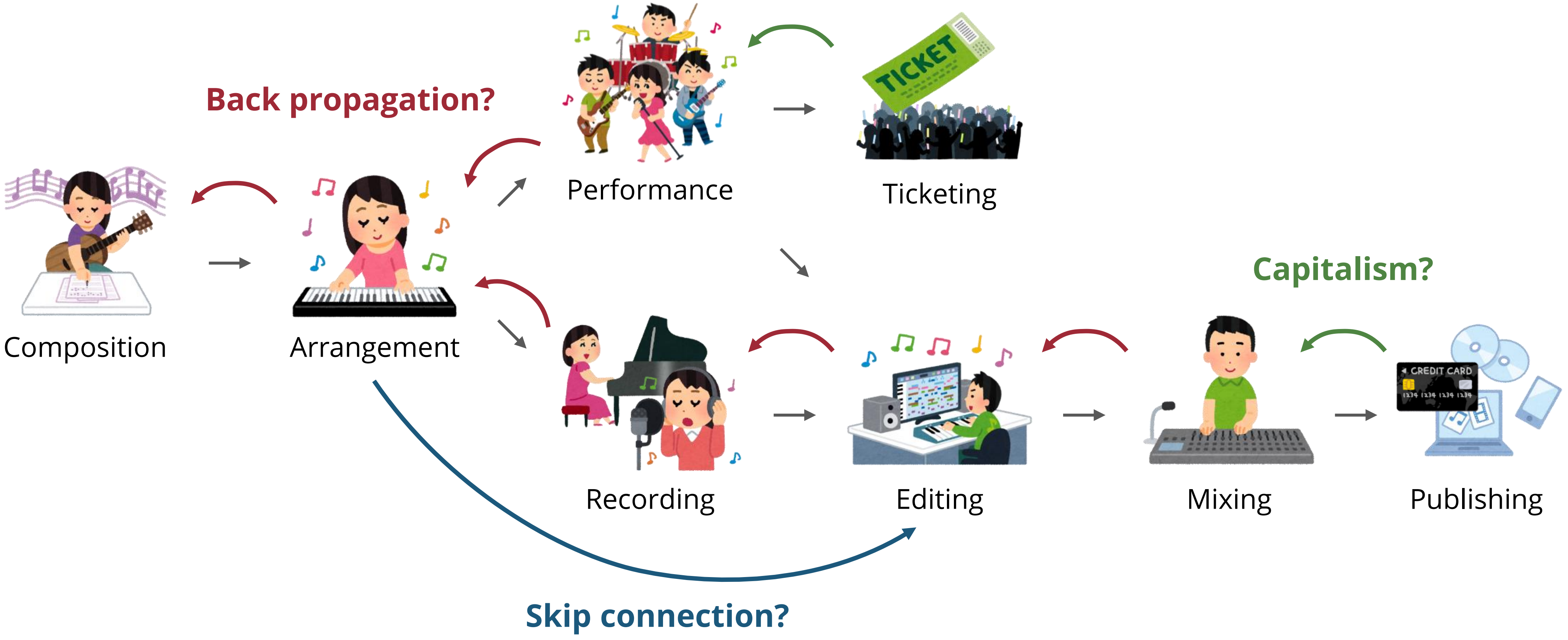


(Source: Garcia et al., 2023)

Hugo Flores Garcia, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo, "VampNet: Music Generation via Masked Acoustic Token Modeling," *ISMIR*, 2023.

The Landscape

A Simplified Music Production Workflow



Back propagation?

Performance

Ticketing

Composition

Arrangement

Capitalism?

Recording

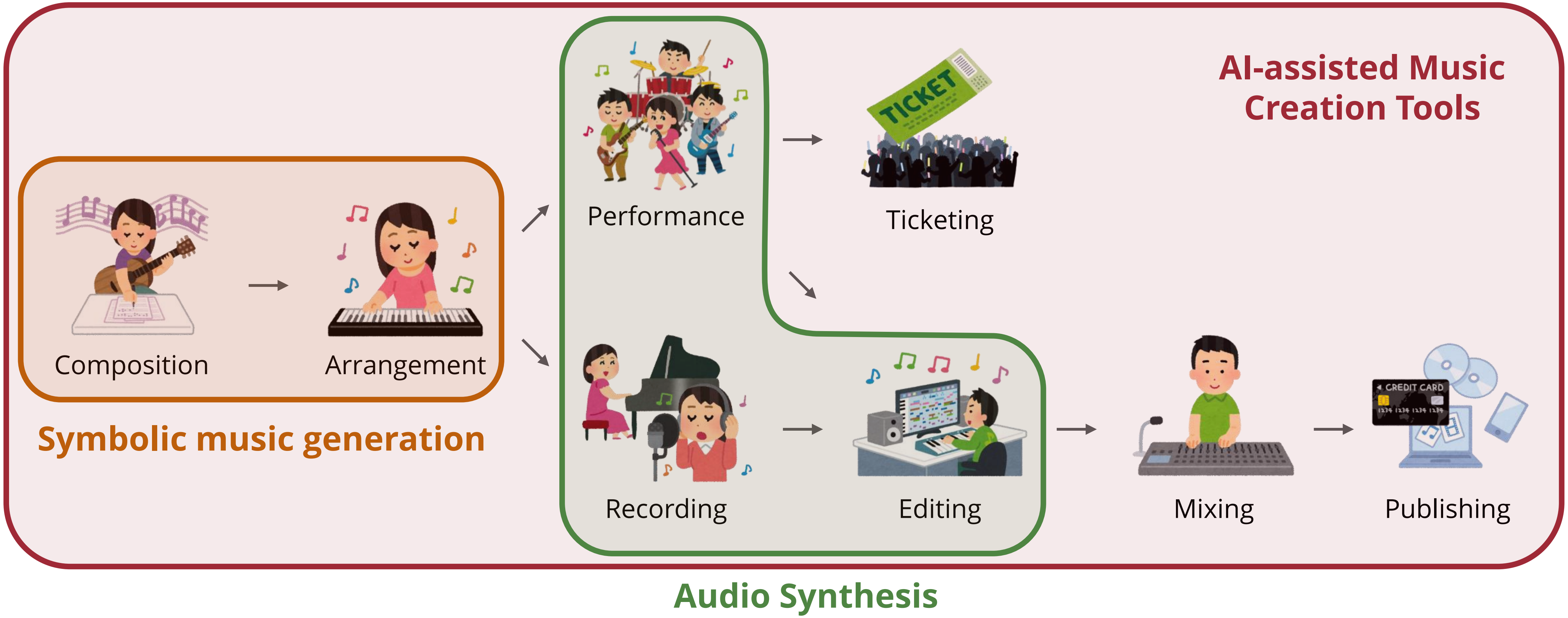
Editing

Mixing

Publishing

Skip connection?

A Simplified Music Production Workflow



AI-assisted Music Creation Tools

Composition → Arrangement

Symbolic music generation

Performance → Ticketing

Recording → Editing → Mixing → Publishing

Audio Synthesis

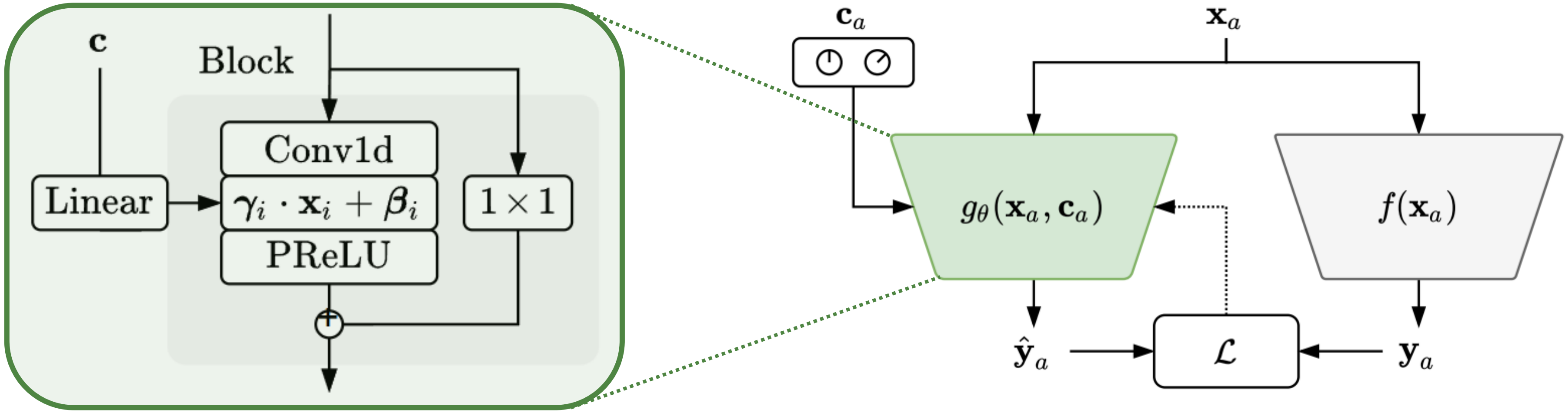
Neural Audio Effects

Example: Neural Audio Effects (Steinmetz et al., 2021)

youtu.be/Zmo8kB-SfF4

Christian J. Steinmetz and Joshua D. Reiss, “Steerable discovery of neural audio effects,” *NeurIPS ML4CD Workshop*, 2021.

Example: Neural Audio Effects (Steinmetz et al., 2021)



(Source: Steinmetz et al., 2021)

csteinmetz1.github.io/steerable-nafx

Christian J. Steinmetz and Joshua D. Reiss, "Steerable discovery of neural audio effects," *NeurIPS ML4CD Workshop*, 2021.

Example: Neural Audio Effects (Steinmetz et al., 2021)

Reverb (vocal)

🔈	**Input (clean)**		
🔈	0	0	Default reverb
🔈	-2	1	Shorter reverb
🔈	-1	5	Longer reverb
🔈	-7	10	Distortion reverb

Reverb (guitar)

🔈	**Input (clean)**		
🔈	-7	10	Large room
🔈	1	1	Small room

csteinmetz1.github.io/steerable-nafx

Christian J. Steinmetz and Joshua D. Reiss, "Steerable discovery of neural audio effects," *NeurIPS ML4CD Workshop*, 2021.

Example: Neural Audio Effects (Steinmetz et al., 2021)

Delay (synth)

🔈	**Input (clean)**		
🔈	0	0	Default reverb
🔈	-3	-3	Shorter reverb
🔈	10	0	Longer reverb

Amplifier (guitar)

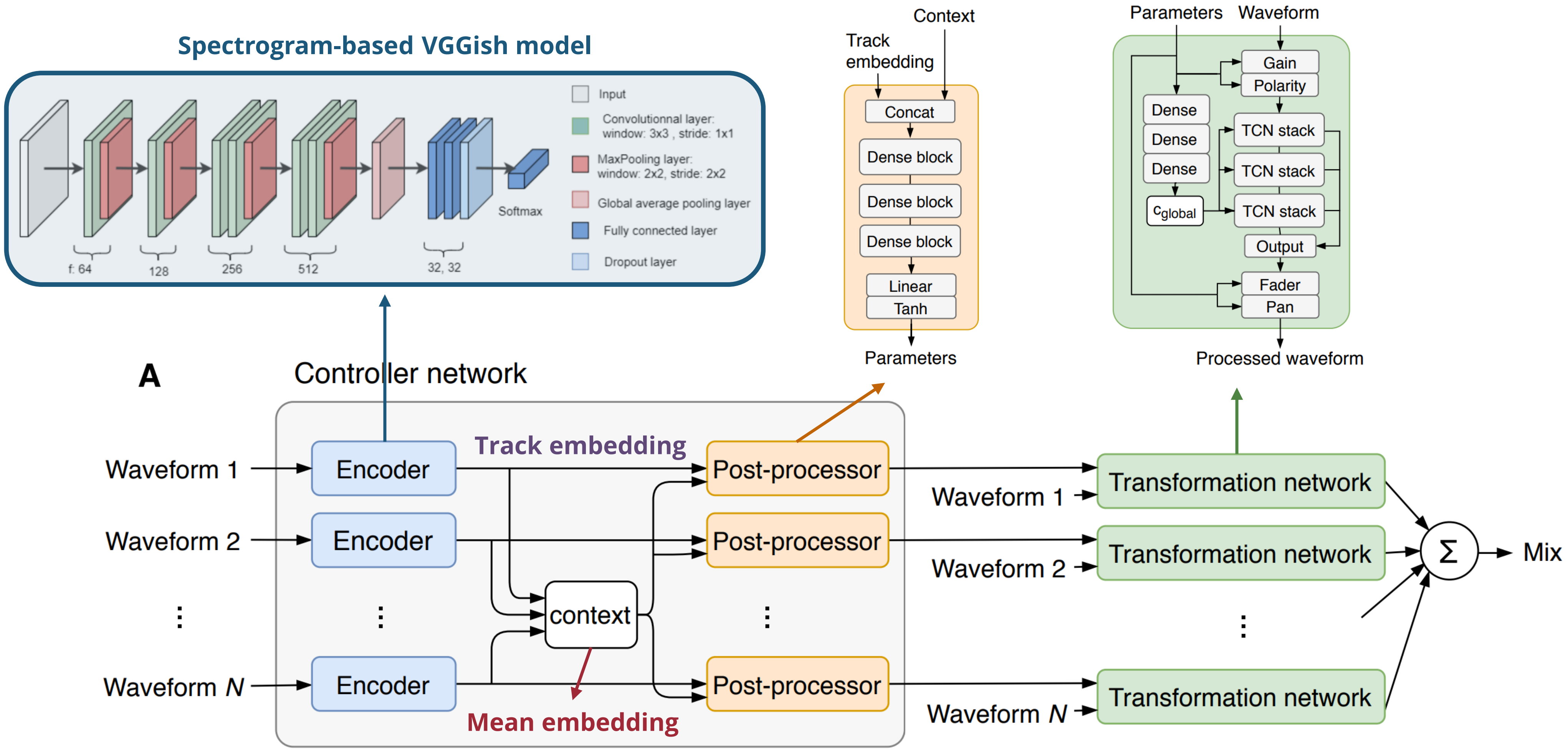
🔈	**Input (clean)**		
🔈	0	0	Amp slapback
🔈	-1	-1	Soft fuzz slap
🔈	10	-10	Tunnel

csteinmetz1.github.io/steerable-nafx

Christian J. Steinmetz and Joshua D. Reiss, "Steerable discovery of neural audio effects," *NeurIPS ML4CD Workshop*, 2021.

Deep Auto-mixing

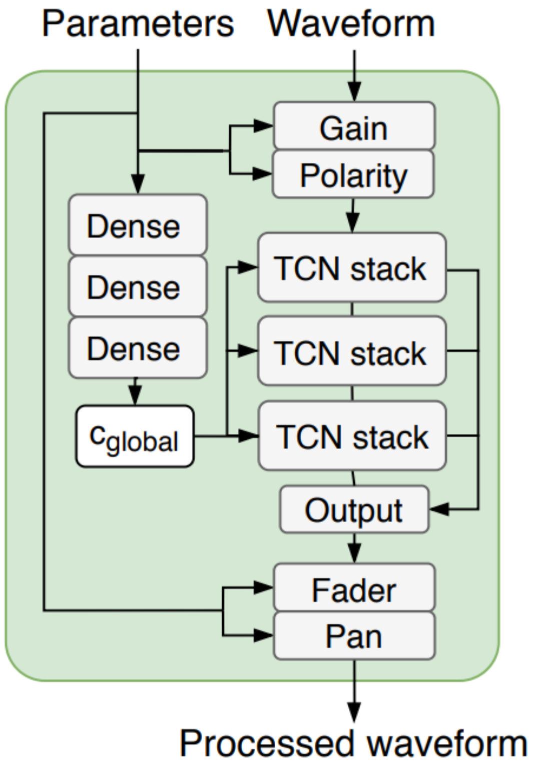
Example: Differentiable Auto-mixing (Steinmetz et al., 2021)



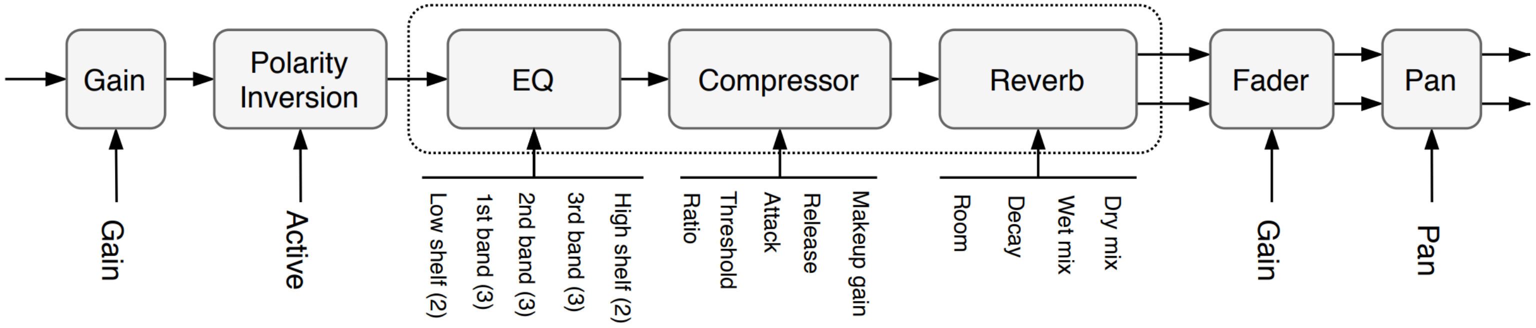
(Source: Steinmetz et al., 2021)

Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," *ICASSP*, 2021.

Example: Differentiable Auto-mixing (Steinmetz et al., 2021)



(Source: Steinmetz et al., 2021)



(Source: Steinmetz et al., 2021)

A differentiable (and thus trainable) mixing console!

github.com/csteinmetz1/pymixconsole

Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," *ICASSP*, 2021.

Example: Differentiable Auto-mixing (Steinmetz et al., 2021)

Transformation Network

Input	**Target**	**Output**
🔈	🔈	🔈
🔈	🔈	🔈

csteinmetz1.github.io/dmc-icassp2021

Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” *ICASSP*, 2021.

Example: Differentiable Auto-mixing (Steinmetz et al., 2021)

Drum mixing

(Same mixing style)

DMC	Mono	Random	Target
🔈	🔈	🔈	🔈
🔈	🔈	🔈	🔈

Multitrack mixing

(Diverse mixing style)

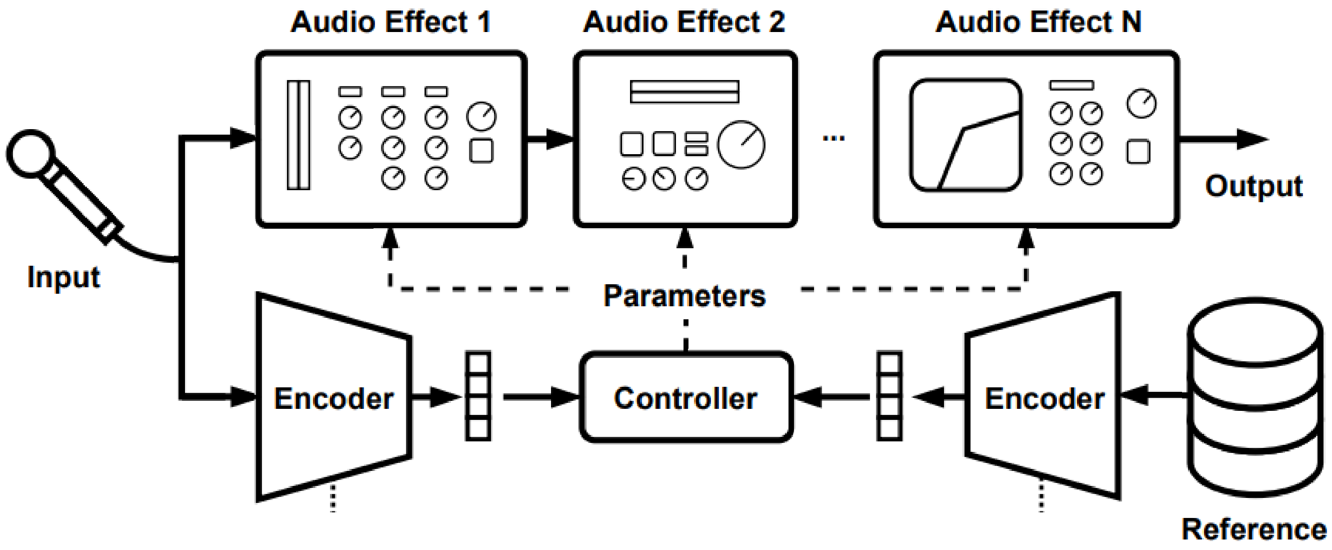
DMC	Mono	Random	Target
🔈	🔈	🔈	🔈
🔈	🔈	🔈	🔈

csteinmetz1.github.io/dmc-icassp2021

Christian J. Steinmetz, Jordi Pons, Santiago Pascual, and Joan Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” *ICASSP*, 2021.

Effects & Mixing Style Transfer

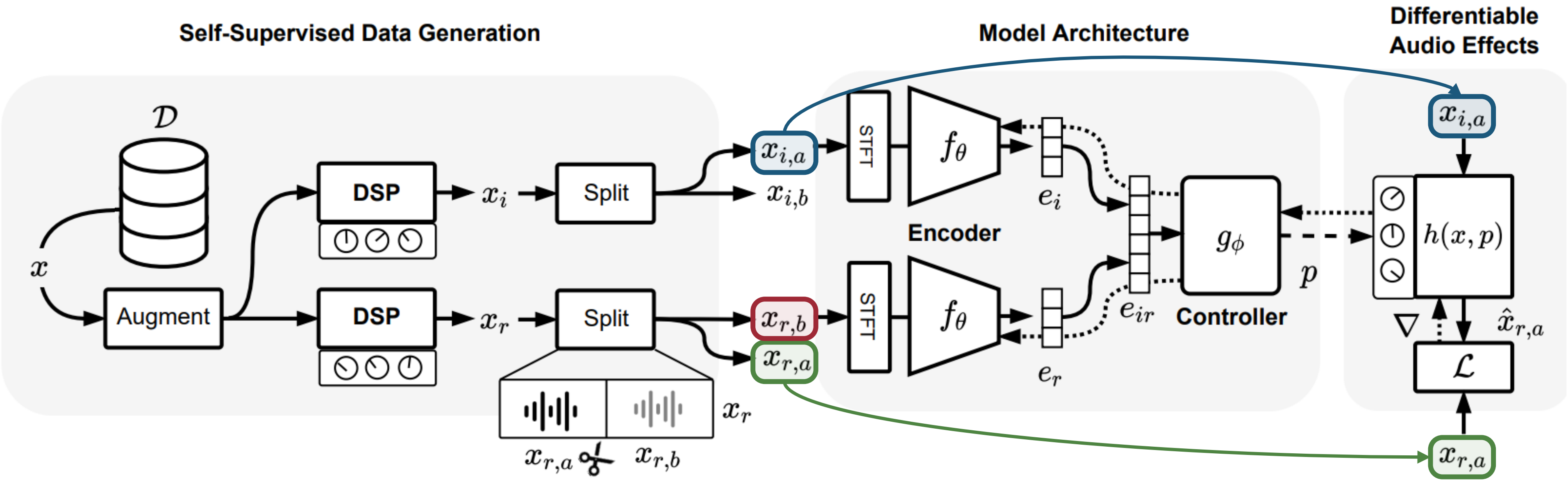
Example: DeepAFx-ST (Steinmetz et al., 2022)



(Source: Steinmetz et al., 2022)

Christian J. Steinmetz, Nicholas J. Bryan, and Joshua D. Reiss, "Style Transfer of Audio Effects with Differentiable Signal Processing," *JAES*, 2022.

Example: DeepAFx-ST (Steinmetz et al., 2022)

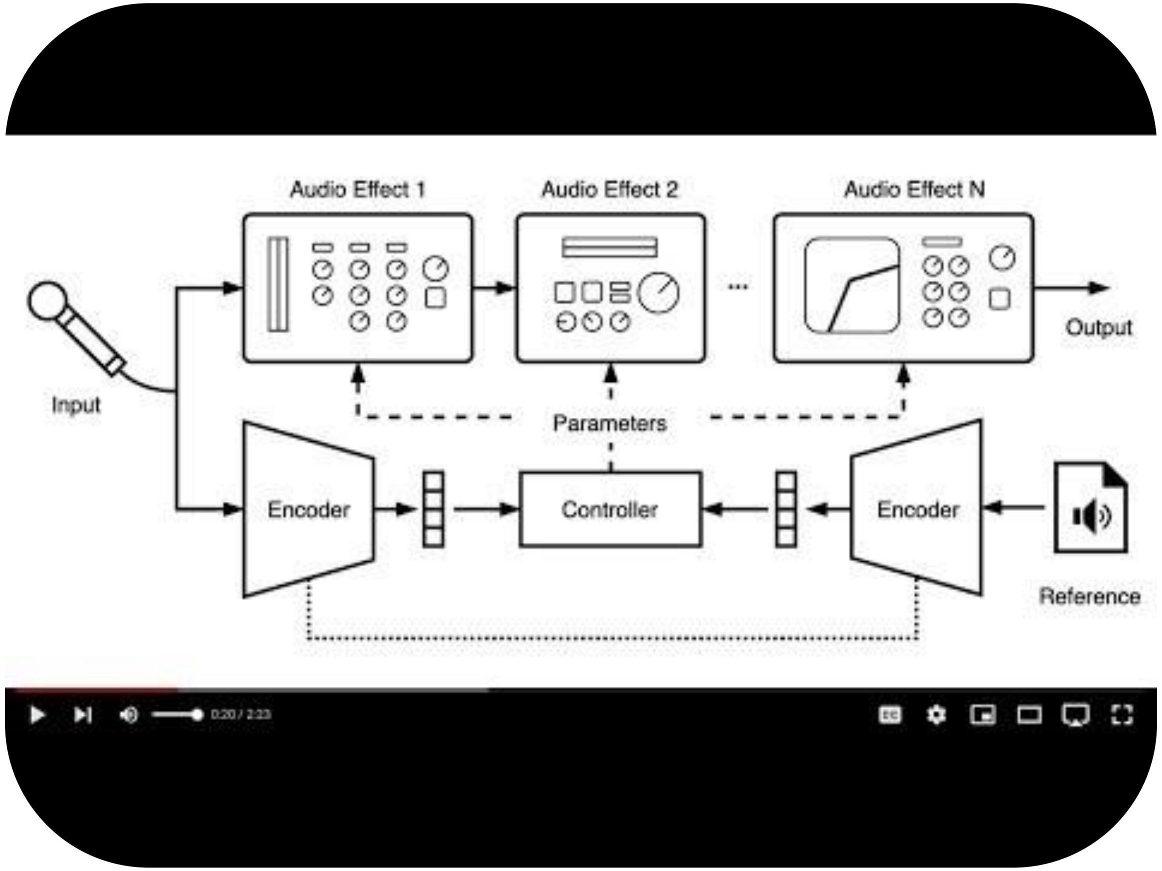


(Source: Steinmetz et al., 2022)

csteinmetz1.github.io/DeepAFx-ST

Christian J. Steinmetz, Nicholas J. Bryan, and Joshua D. Reiss, “Style Transfer of Audio Effects with Differentiable Signal Processing,” *JAES*, 2022.

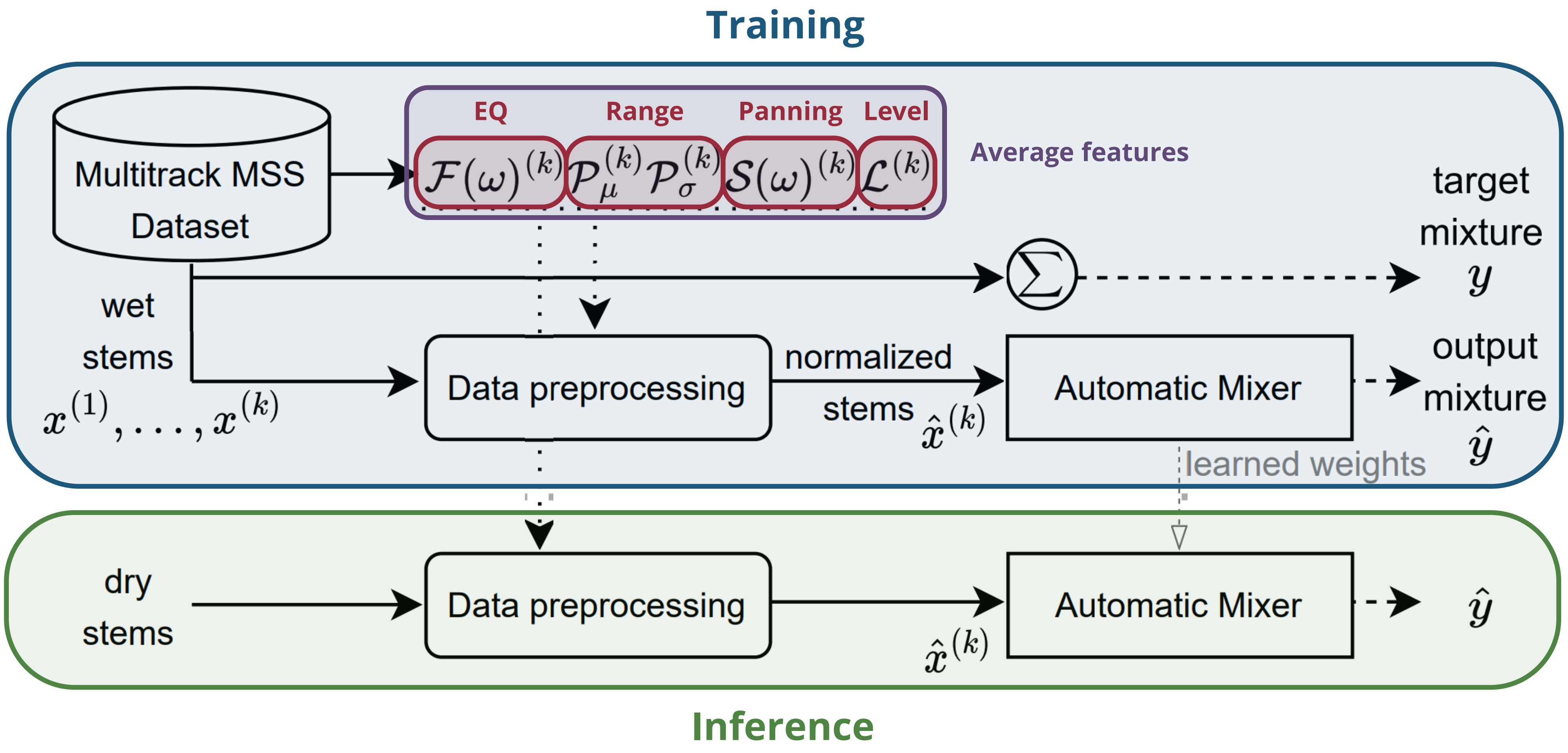
Example: DeepAFx-ST (Steinmetz et al., 2022)



youtu.be/IZp455wiMk4?t=100

Christian J. Steinmetz, Nicholas J. Bryan, and Joshua D. Reiss, “Style Transfer of Audio Effects with Differentiable Signal Processing,” *JAES*, 2022.

Example: FX Normalization (Martínez-Ramírez et al., 2022)



(Source: Martínez-Ramírez et al., 2022)

Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Giorgio Fabbro, Stefan Uhlich, Chihiro Nagashima, and Yuki Mitsufuji, "Automatic music mixing with deep learning and out-of-domain data," *ISMIR*, 2022.

Example: FX Normalization (Martínez-Ramírez et al., 2022)

	Dry	Normalized
Vocals	🔊	🔊
Drums	🔊	🔊
Bass	🔊	🔊
Other	🔊	🔊
Mix	🔊	🔊

Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Giorgio Fabbro, Stefan Uhlich, Chihiro Nagashima, and Yuki Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” *ISMIR*, 2022.

Example: FX Normalization (Martínez-Ramírez et al., 2022)

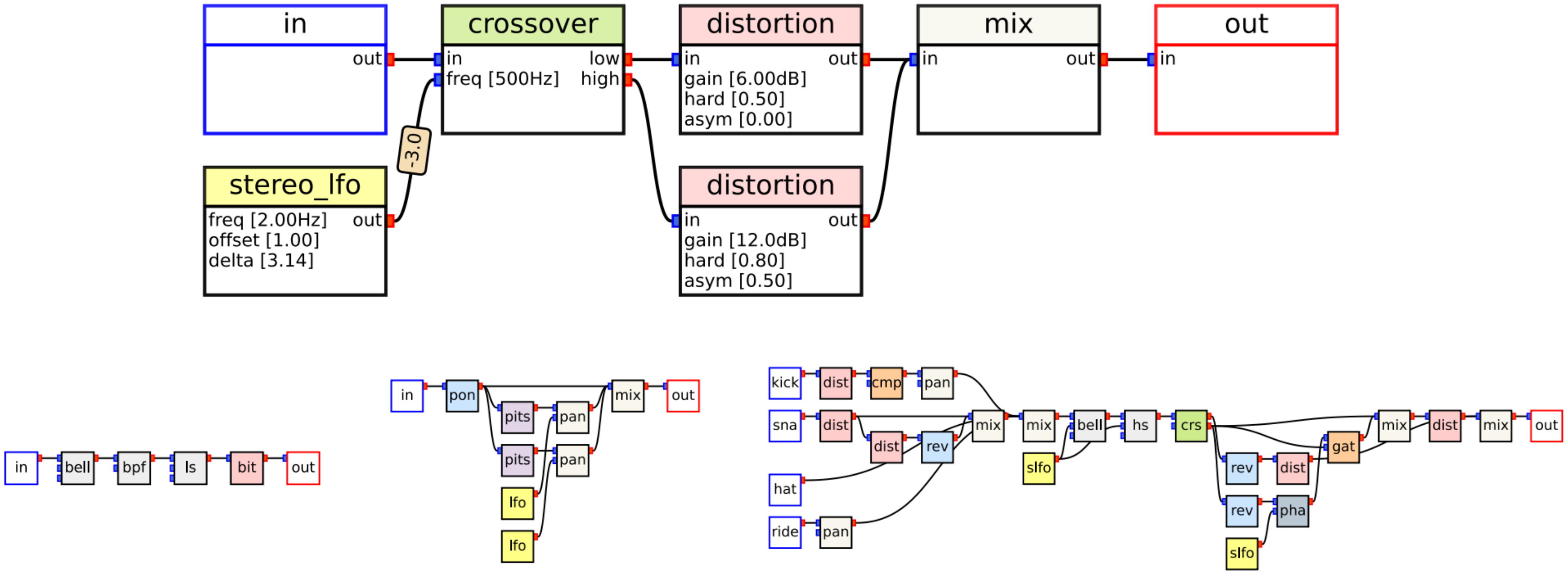
	Large private dataset		Small public dataset
Human	Loss B	Loss A	Loss B
🔊	🔊	🔊	🔊
🔊	🔊	🔊	🔊

marco-martinez-sony.github.io/FxNorm-automix/AUDIO_SAMPLES
github.com/sony/fxnorm-automix

Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Giorgio Fabbro, Stefan Uhlich, Chihiro Nagashima, and Yuki Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” *ISMIR*, 2022.

Beyond Fixed Processing Graph

Example: Audio Processing Graph (Lee et al., 2022)



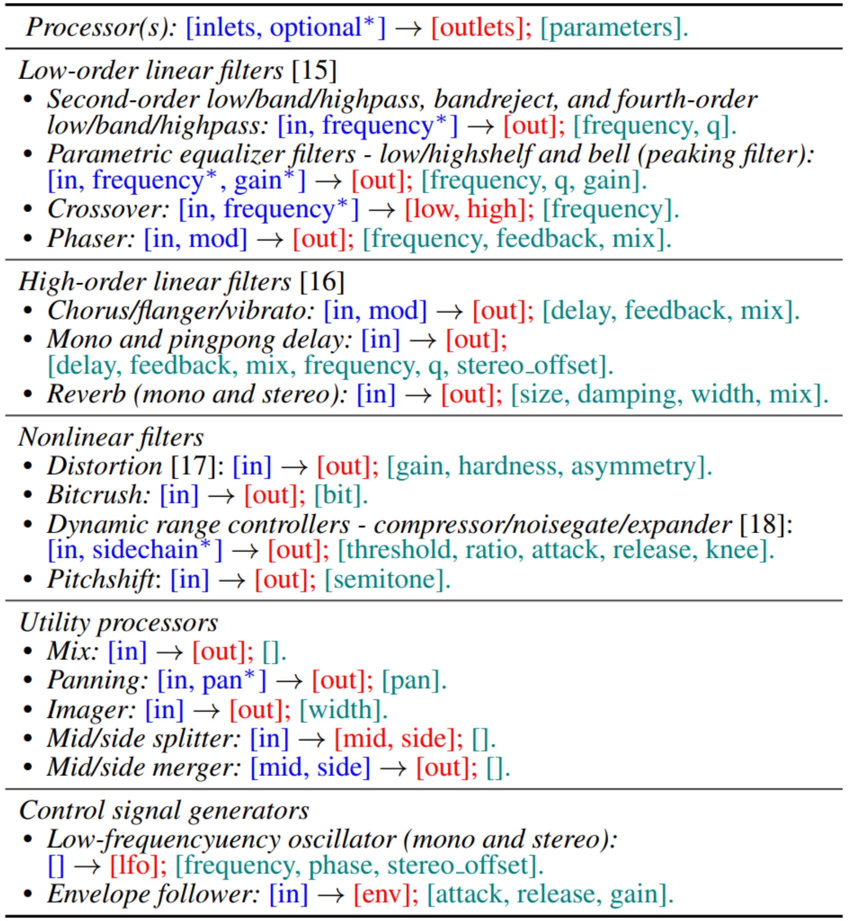
Can we predict the audio processing graph used in a reference recording?

(Source: Lee et al., 2023)

Sungho Lee, Jaehyun Park, Seungryeol Paik, and Kyogu Lee, "Blind Estimation of Audio Processing Graph," *ICASSP*, 2023.

Example: Audio Processing Graph (Lee et al., 2022)

Supported processors



Processor(s): [inlets, optional*] → [outlets]; [parameters].

Low-order linear filters [15]

- *Second-order low/band/highpass, bandreject, and fourth-order low/band/highpass:* [in, frequency*] → [out]; [frequency, q].
- *Parametric equalizer filters - low/highshelf and bell (peaking filter):* [in, frequency*, gain*] → [out]; [frequency, q, gain].
- *Crossover:* [in, frequency*] → [low, high]; [frequency].
- *Phaser:* [in, mod] → [out]; [frequency, feedback, mix].

High-order linear filters [16]

- *Chorus/flanger/vibrato:* [in, mod] → [out]; [delay, feedback, mix].
- *Mono and pingpong delay:* [in] → [out]; [delay, feedback, mix, frequency, q, stereo_offset].
- *Reverb (mono and stereo):* [in] → [out]; [size, damping, width, mix].

Nonlinear filters

- *Distortion* [17]: [in] → [out]; [gain, hardness, asymmetry].
- *Bitcrush:* [in] → [out]; [bit].
- *Dynamic range controllers - compressor/noisegate/expander* [18]: [in, sidechain*] → [out]; [threshold, ratio, attack, release, knee].
- *Pitchshift*: [in] → [out]; [semitone].

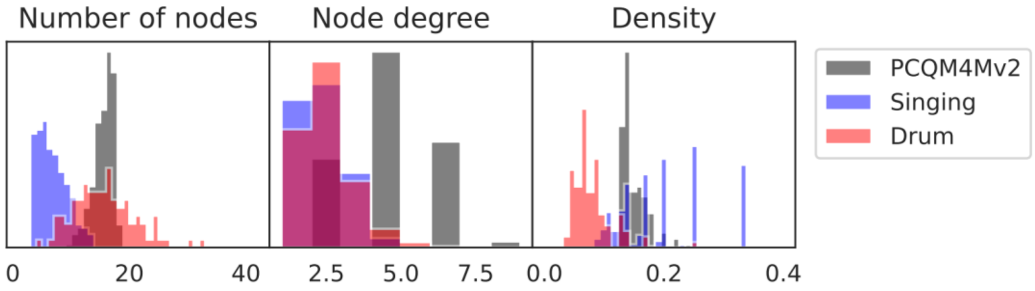
Utility processors

- *Mix:* [in] → [out]; [].
- *Panning:* [in, pan*] → [out]; [pan].
- *Imager:* [in] → [out]; [width].
- *Mid/side splitter:* [in] → [mid, side]; [].
- *Mid/side merger:* [mid, side] → [out]; [].

Control signal generators

- *Low-frequencyuency oscillator (mono and stereo):* [] → [lfo]; [frequency, phase, stereo_offset].
- *Envelope follower:* [in] → [env]; [attack, release, gain].

Data statistics

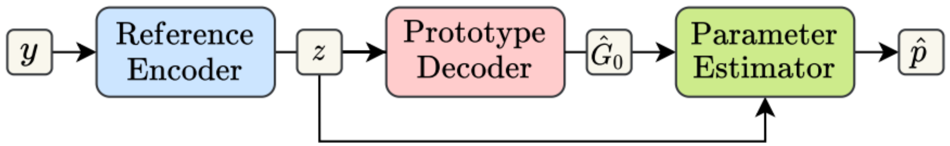


(Source: Lee et al., 2023)

Sungho Lee, Jaehyun Park, Seungryeol Paik, and Kyogu Lee, "Blind Estimation of Audio Processing Graph," *ICASSP*, 2023.

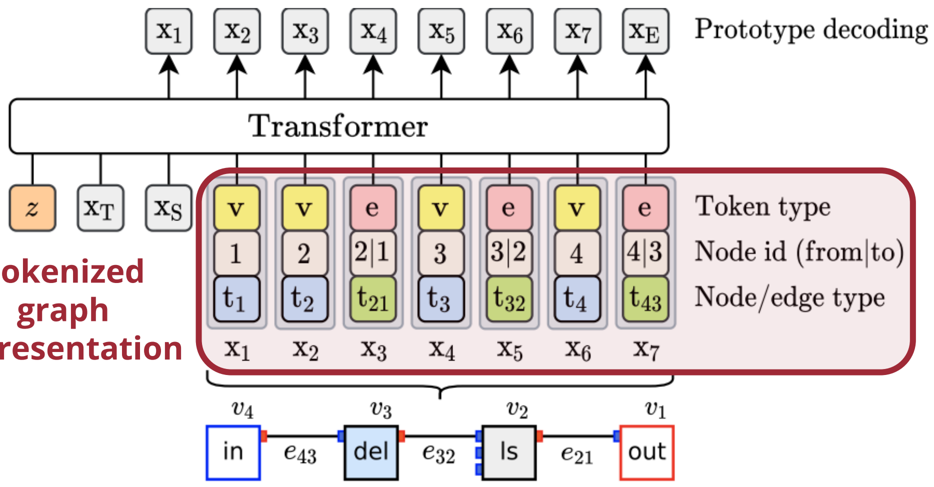
Example: Audio Processing Graph (Lee et al., 2022)

Blind estimation framework



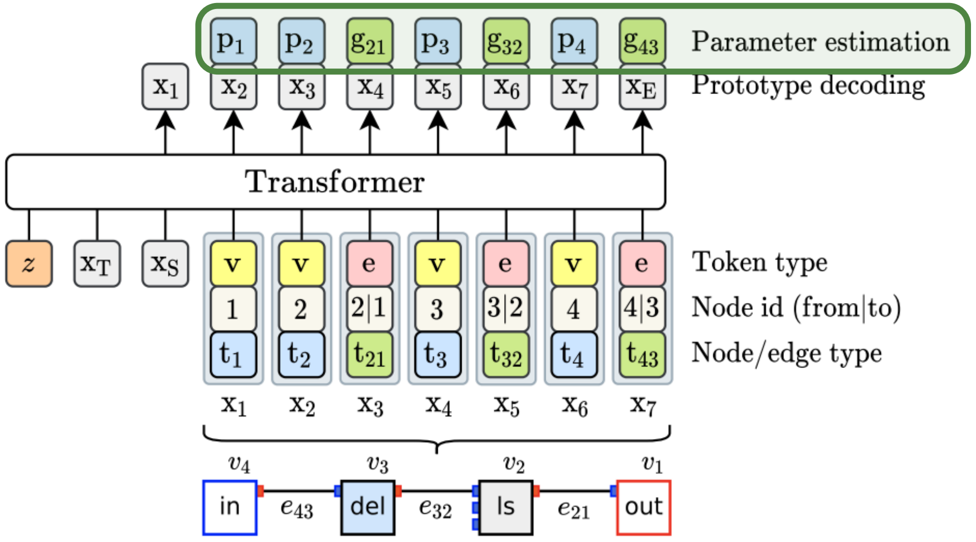
Sources are not given!

Prototype decoder


x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_E
Prototype decoding
Transformer
z x_T x_S
v v e v e v e
Token type
1 2 2|1 3 3|2 4 4|3
Node id (from|to)
t_1 t_2 t_{21} t_3 t_{32} t_4 t_{43}
Node/edge type
x_1 x_2 x_3 x_4 x_5 x_6 x_7
v_4 v_3 v_2 v_1
in e_{43} del e_{32} ls e_{21} out

Tokenized graph representation

Parameter estimator


p_1 p_2 g_{21} p_3 g_{32} p_4 g_{43}
Parameter estimation
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_E
Prototype decoding
Transformer
z x_T x_S
v v e v e v e
Token type
1 2 2|1 3 3|2 4 4|3
Node id (from|to)
t_1 t_2 t_{21} t_3 t_{32} t_4 t_{43}
Node/edge type
x_1 x_2 x_3 x_4 x_5 x_6 x_7
v_4 v_3 v_2 v_1
in e_{43} del e_{32} ls e_{21} out

(Source: Lee et al., 2023)

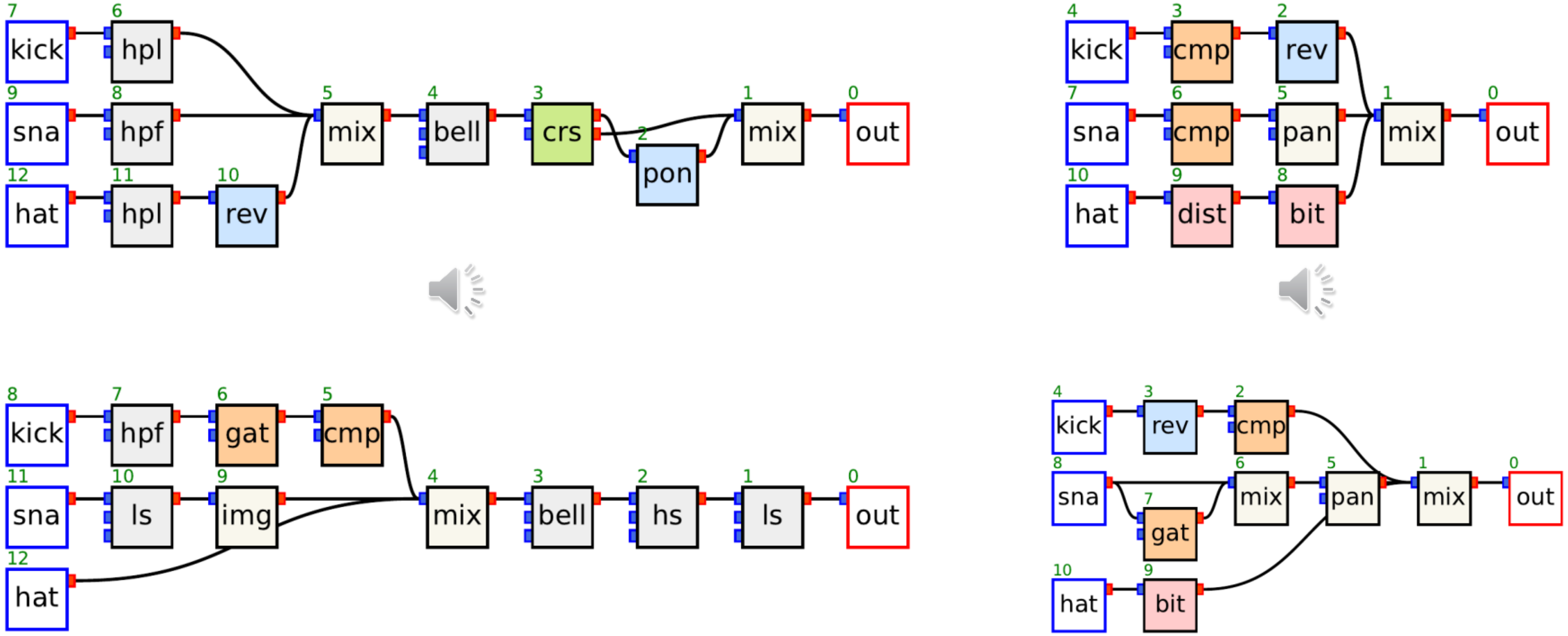
Sungho Lee, Jaehyun Park, Seungryeol Paik, and Kyogu Lee, "Blind Estimation of Audio Processing Graph," *ICASSP*, 2023.

Example: Audio Processing Graph (Lee et al., 2022)

Dry

Reference

Estimation



(Source: Lee et al., 2023)

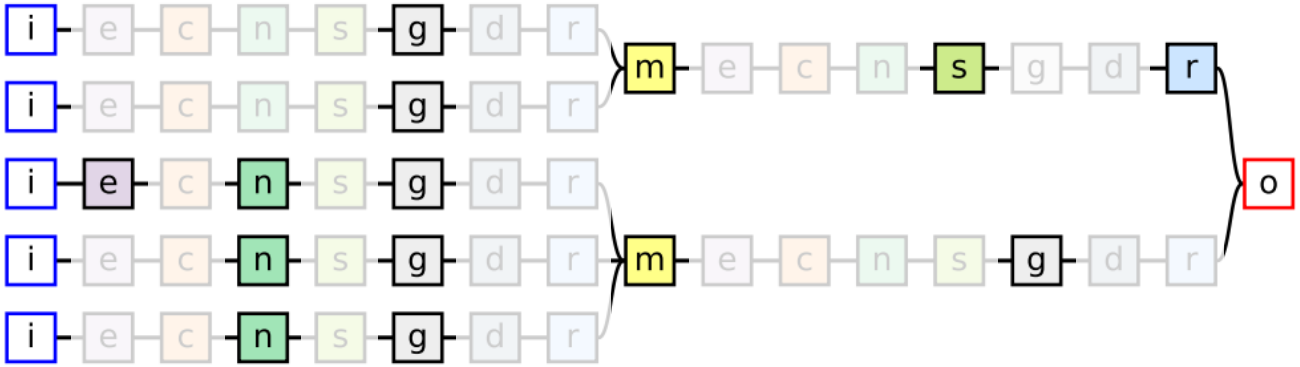
sh-lee97.github.io/apg

Sungho Lee, Jaehyun Park, Seungryeol Paik, and Kyogu Lee, “Blind Estimation of Audio Processing Graph,” *ICASSP*, 2023.

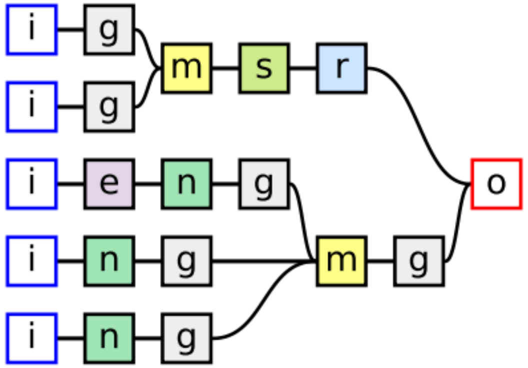
Example: Music Mixing Graph (Lee et al., 2024)

Can we predict the music mixing graph given the sources and reference mixture?

Full mixing console (before pruning)



Pruned graph



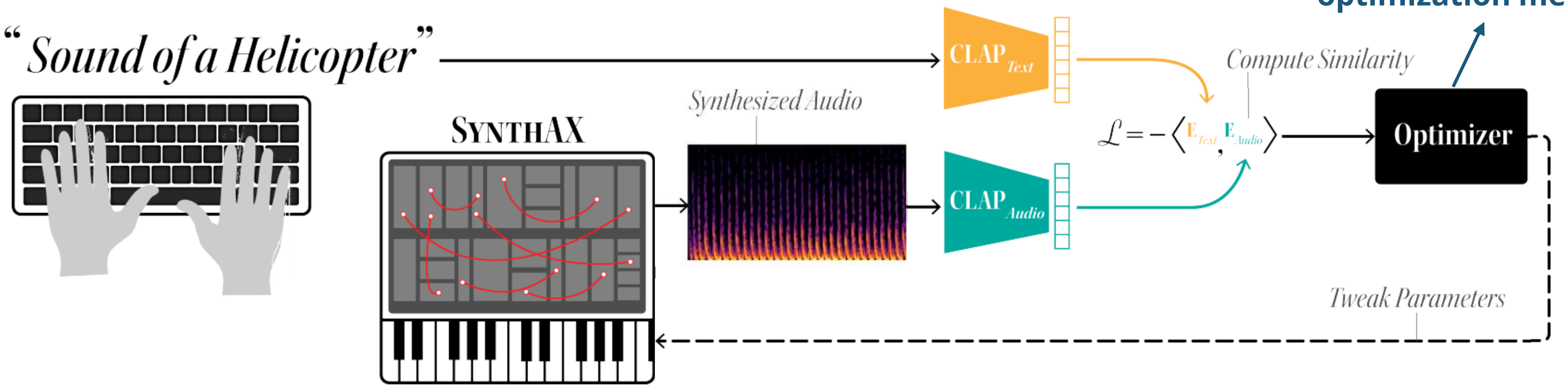
(Source: Lee et al., 2024)

sh-lee97.github.io/grafx-prune

Sungho Lee, Marco A. Martínez-Ramírez, Wei-Hsiang Liao, Stefan Uhlich, Giorgio Fabbro, Kyogu Lee, and Yuki Mitsufuji, "Searching For Music Mixing Graphs: A Pruning Approach," *DAFx*, 2024.

Example: CTAG (Cherep et al., 2024)

Non-gradient-based optimization methods



Does not need to be differentiable!

(Source: Cherep et al., 2024)

ctag.media.mit.edu

Manuel Cherep, Nikhil Singh, and Jessica Shand, "Creative Text-to-Audio Generation via Synthesizer Programming," *ICML*, 2024.