Google

Recomposer: Event-Roll-guided Generative Audio Editing

Dan Ellis *Google DeepMind*

Joint work with **Edu Fonseca, Ron Weiss, Kevin Wilson, Pascal Getreuer, Scott Wisdom, Hakan Erdogan, John Hershey, Aren Jansen, Channing Moore, Manoj Plakal**

Mila Conversational AI Reading Group, Jan 29th 2026

https://arxiv.org/abs/2509.05256

The Team

Dan Ellis

Eduardo Fonseca

Ron Weiss

Kevin Wilson

Pascal Getreuer

Scott Wisdom

Hakan Erdogan

John Hershey

Aren Jansen

Channing Moore

Manoj Plakal

Context: **Machine Learning for Audio**

Domain

- Speech
- Music
- **Environmental sounds**

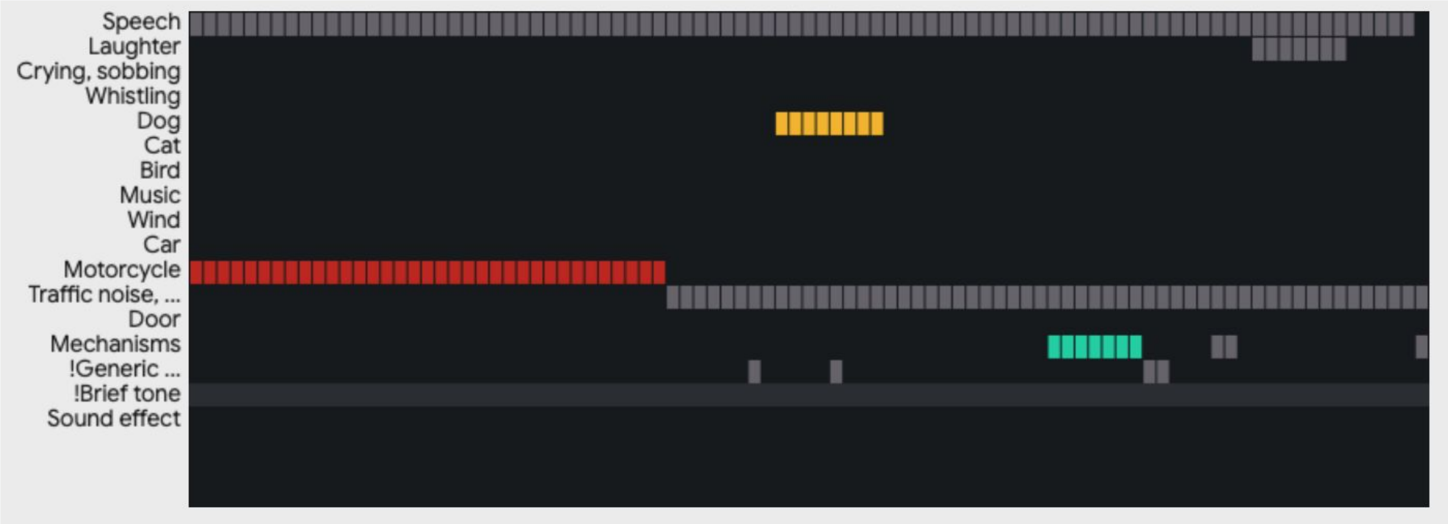
Applications

- Classification / Understanding
- Generation
- **Editing / Modification**

No LLM!

Scenario: **Sound Recomposition**

Contribution: **Event Roll**

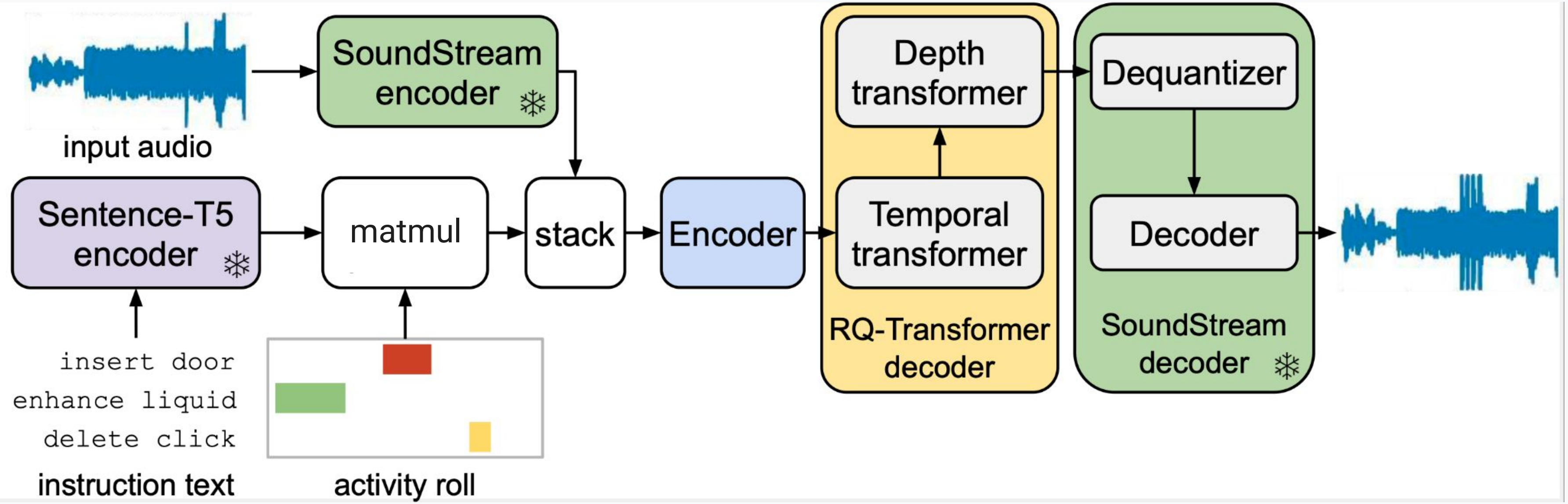


- Represents sound mixture as a collection of **discrete events**
- Each row associated with free-text **label** (Action + Class)
- ***Could*** be generated by an Audio Event Classifier

Contribution: **Enhancement**

- Enhancement combines **sound separation** and **generation**
  - Identify low-level target event in the input
  - Regenerate missing details to synthesize output
- How to specify “Enhancement”?
  - Constant **gain** applied to input?
  - Constant output target **level** (e.g. 10 dB) regardless of input level
  - (allow user to specify)

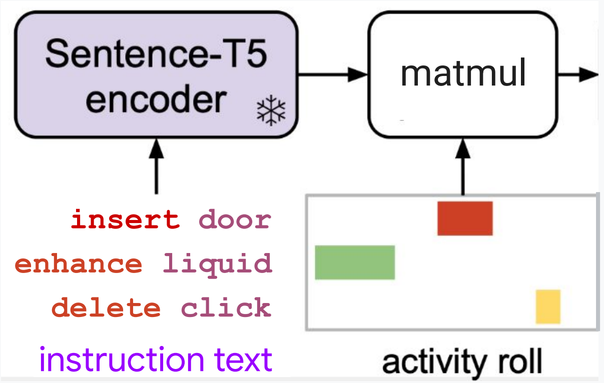
Recomposer Model



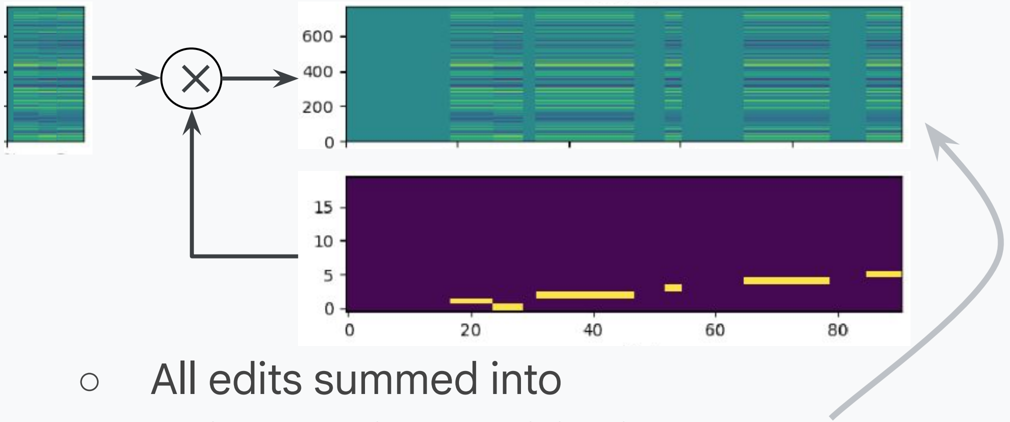
- Input sound **embedding + conditioning** encodes to conditioning **embedding**
- Autoregressive encoder-decoder transformer trained to generate **RVQ tokens**

Autoregressive Image Generation using Residual Quantization, Lee et al, CVPR 2022

Recomposer Model: **Edit conditioning**



- Instruction text is embedded as **general text**
  - **Action** and **Target class** confounded…
  - Permits semantic **adjacency**
- `matmul((embedding, event), (event, time))`
  `→ (embedding, time)`

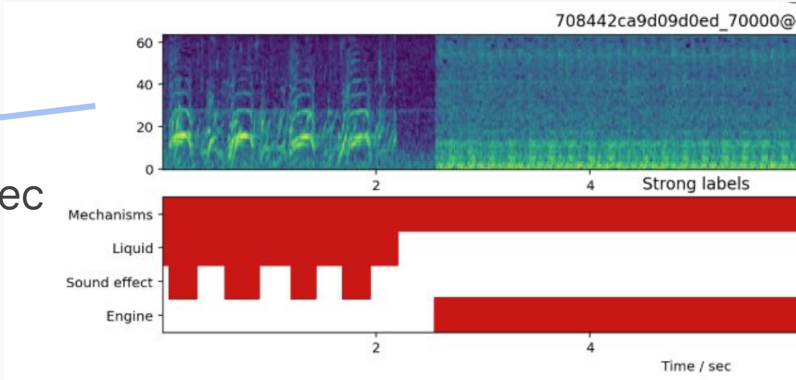


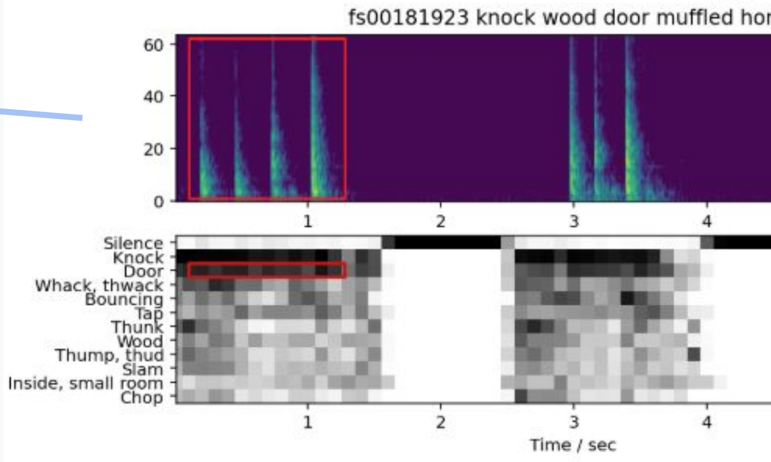
  - All edits summed into
    a **single edit-conditioning vector** per time step

Synthetic Target Data

Training examples as Background + Target

- **Background** from AudioSet (Strong Labeled) (10 sec clips)
    - Selected for Complexity (> 2 classes) and Time coverage (> 9 sec nonsilent)
    - 168k clips
- **Target** events from Freesound
    - Matched by tags + AudioSet classifier
    - Automatic trimming to isolated events of 0.2 .. 2 sec
    - 16k training events from 40 “event-like” AudioSet classes

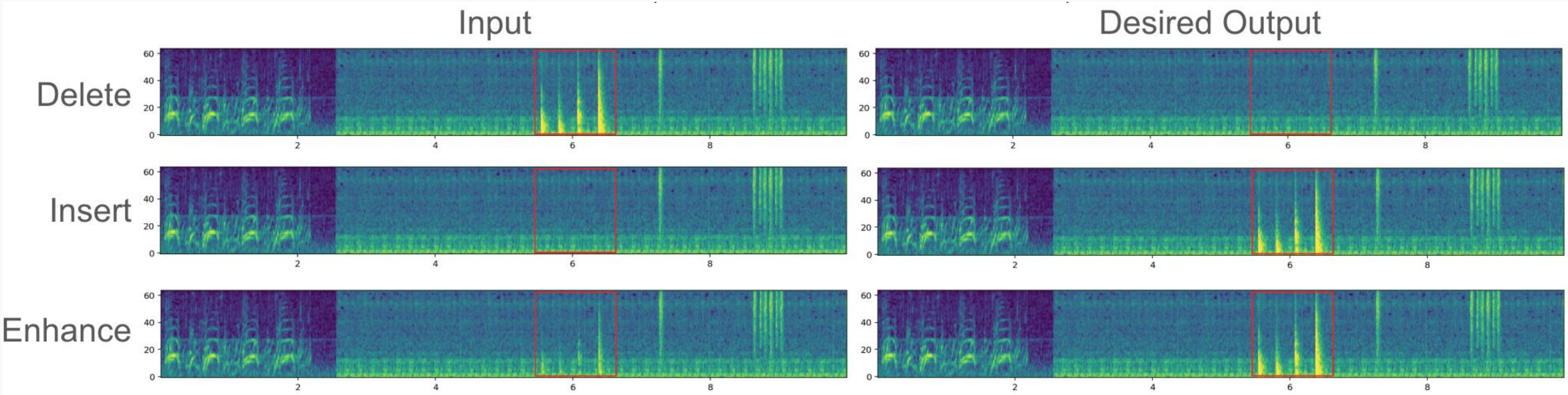




Synthetic Target Data (cont'd)

- Synthetic data {*input, output*} pairs formed by mixing Target + Background
- Fix **Target-to-Background Ratio (TBR)** vs. the overlapping background

Three kinds of Action



Synthetic Target Data details

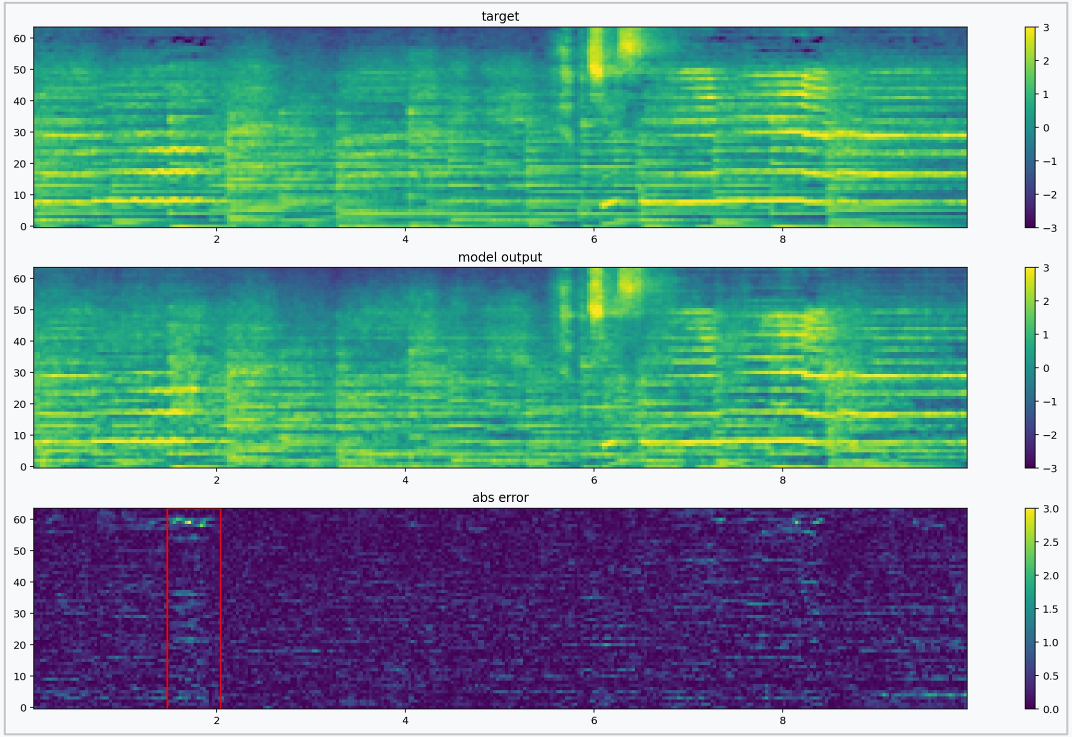
- Training: Generate **on-the-fly** *{input, output}* pairs
    - Two "Targets" per example
        - Each chosen from Enhance, Delete, Insert, or None (EDIN)
        - No time overlap between targets
    - Targets inserted at 10 (± 3) dB TBR
        - Loud enough to measure, still plausible
    - Enhancement inputs at -6 (± 3) dB TBR

Metrics: **Multi-Scale Spectral Distortion** (MSD)

Desired output

Model output

Absolute difference



.. sum is ~MSD

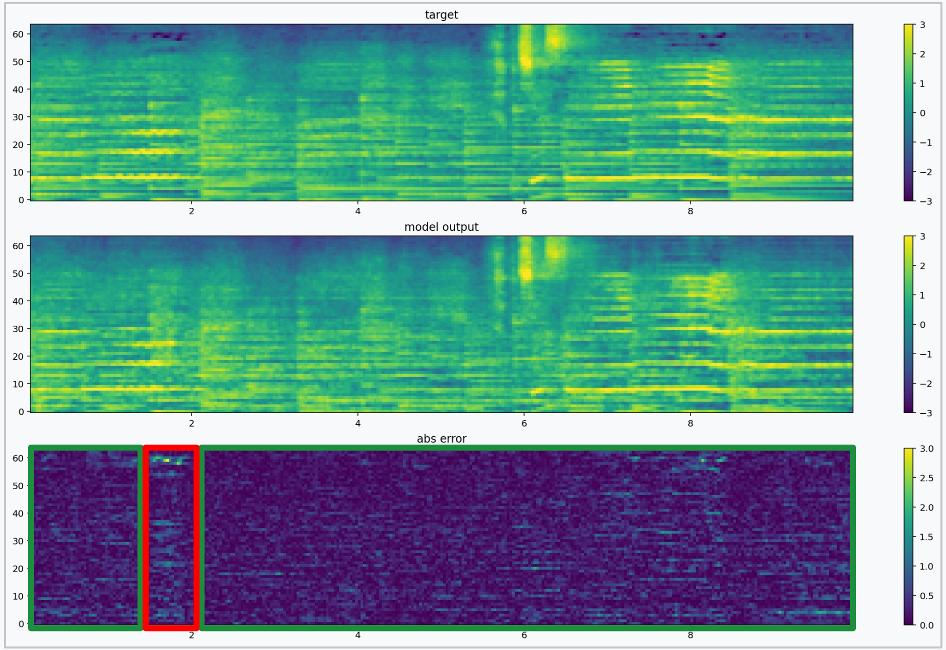
- Constant baseline "**model distortion**" - but differences are not audible

Google

[DDSP: Differentiable Digital Signal Processing](), Engel et al., ICLR 2020.

Metrics: **Modified vs. Unmodified** regions

- Target event occupies ~10% of time
- "Model distortion" over remainder can dominate metrics
- **Framewise** metrics allow decomposition by time range
    - Use ground-truth target region to calculate separate metrics for **modified** & **unmodified** regions



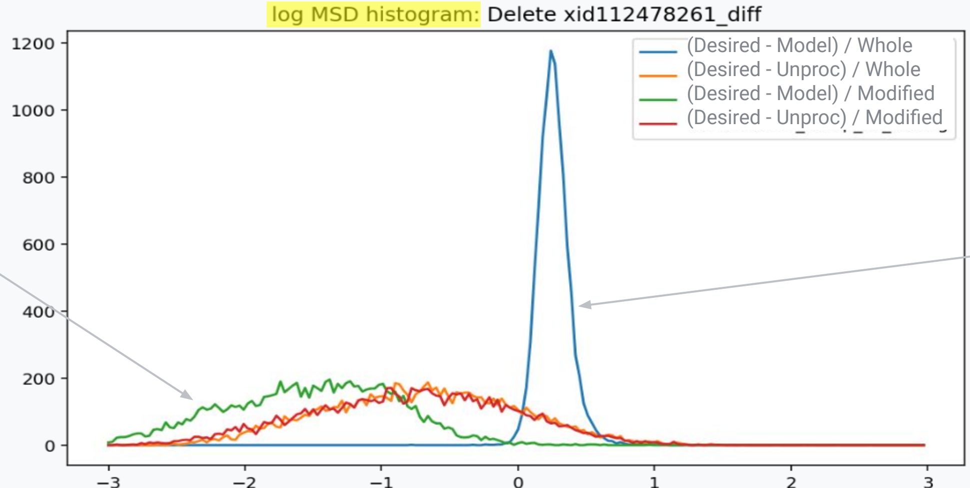
Metrics: **Modified vs. Unmodified** regions

- Look at **histograms** of metrics across eval set
  - Compare `(Desired - Model_output)` and `(Desired - Unprocessed)`
  - Calculate metrics over `Whole_clip` vs. `Modified_region`

- "Model distortion" overwhelms `(Desired - Model_output) / Whole_clip`

  Restricting to `Modified_region` reveals benefit:

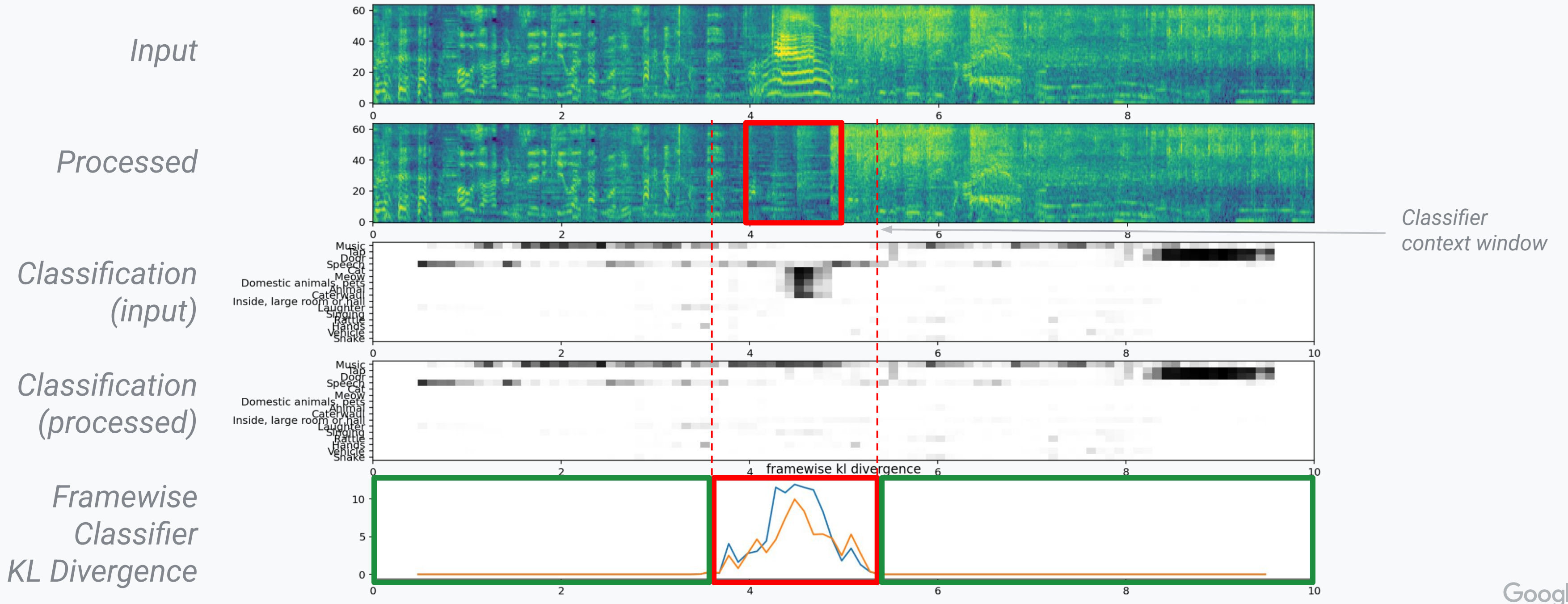
*Modified_region metrics: Model_output **better** than Unprocessed*



*Whole_clip metrics: Model_output **worse** than Unprocessed*

Metrics: **Classifier KL Divergence** (KLD)

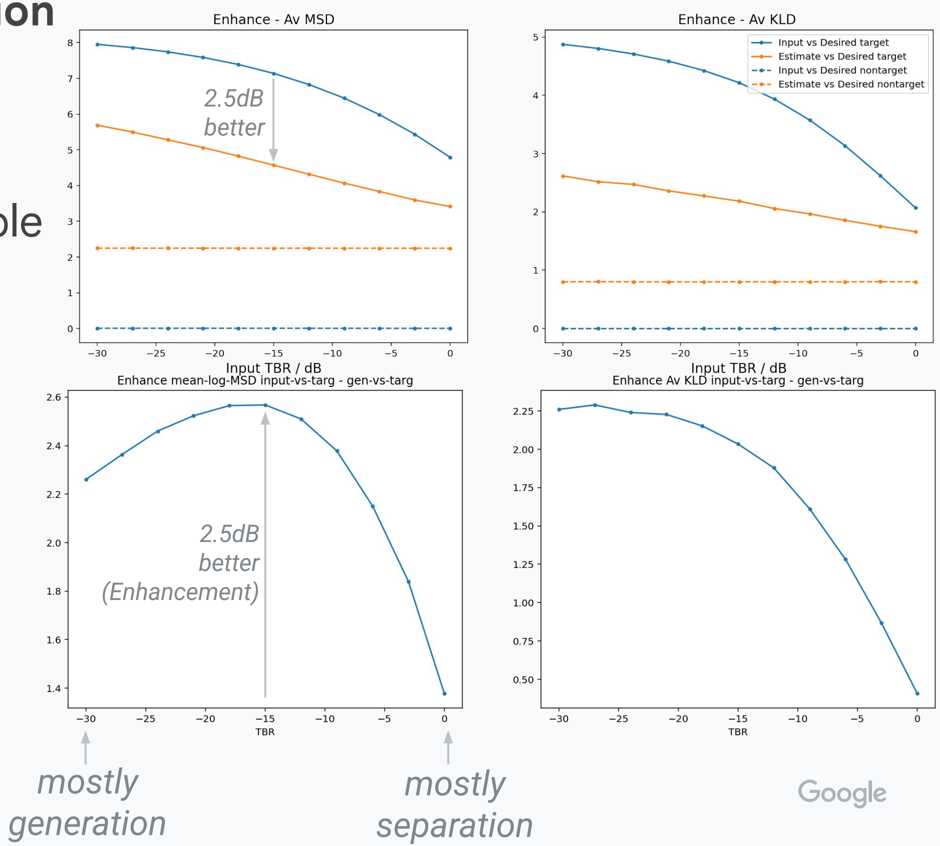
- KLD compares **framewise classifier** outputs - needed to evaluate **Insert**
  - **YAMNet** AudioSet classifier evaluated every 100 ms
  - Normalize each frame across class, calculate **KL divergence**, average along time



[AudioGen: Textually Guided Audio Generation](), Kreuk et al., ICLR 2023.

Results: **Varying Input Level for Enhancement**

- Focus on generation-to-separation **transition** in Enhancement
- Training:
  - One Enhancement target per example
  - Inputs ranged over -30 to 0 dB TBR
  - Output always at +15 dB (minimize uncertainty)
- Evaluation:
  - Metrics improve with TBR
  - Improvement-over-unprocessed peaks at ~ -15 dB TBR (for MSD)
  - (KLD has no peak)



Results: **Overall Performance**

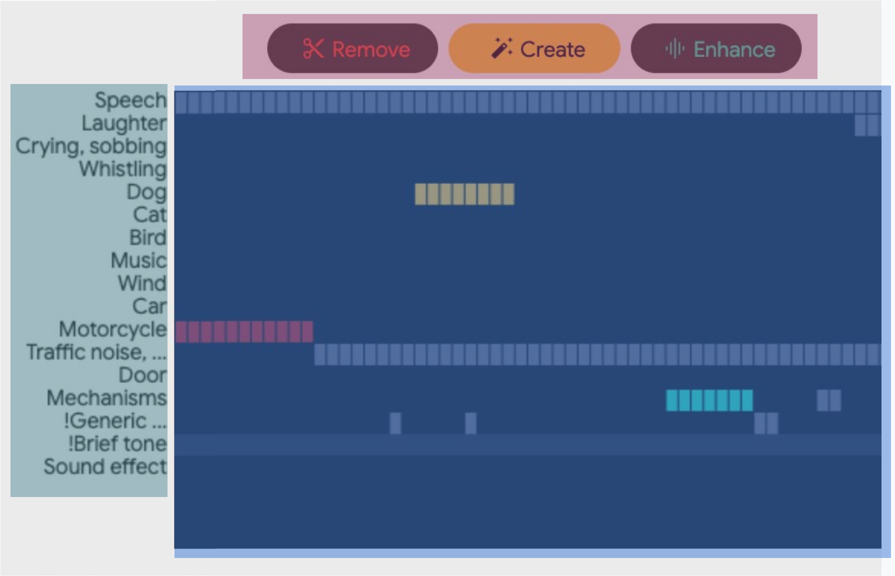
Region	Signal	Delete MSD	Delete KLD	Insert MSD	Insert KLD	Enhance MSD	Enhance KLD
Target	input	4.8	1.6	4.8	2.8	3.4	1.6
	estimate	2.5	0.5	5.1	1.9	2.6	0.9
Nontarget	input	0.0	0.0	0.0	0.0	0.0	0.0
	estimate	1.3	0.3	1.3	0.3	1.3	0.3

(lower is better)

- Nontarget input is perfect - distances of 0
- Nontarget estimate is “**model distortion**” - limitation of copying
- Target estimate minus input gives **improvement** from processing
    - For MSD, **Delete** does well, **Insert** is made worse (specific output is unknown)
    - For KLD, **Insert** reveals benefit (because target class is specified)
    - **Enhance** results are for ~ -6 dB TBR inputs, limited headroom

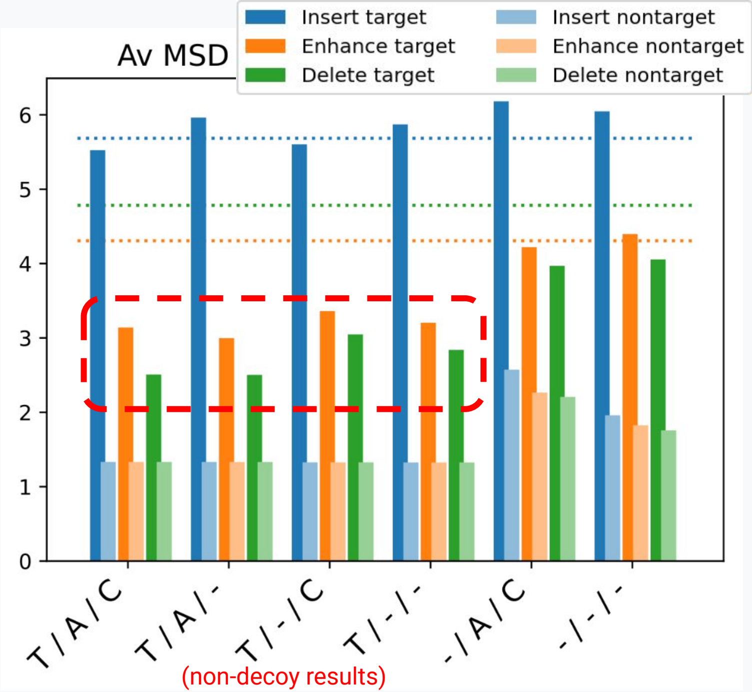
Conditioning Ablation

- Results show improvement - we're done?
  - Or is there more insight?
- Conditioning has 3 components
  - Timing (Event Roll)
  - Action (Delete/Insert/Enhance)
  - Class (Description of target)
- Train 6 models with partial conditioning
- Evaluate with **Decoy** examples
  - Minimize cues in input



Conditioning Ablation

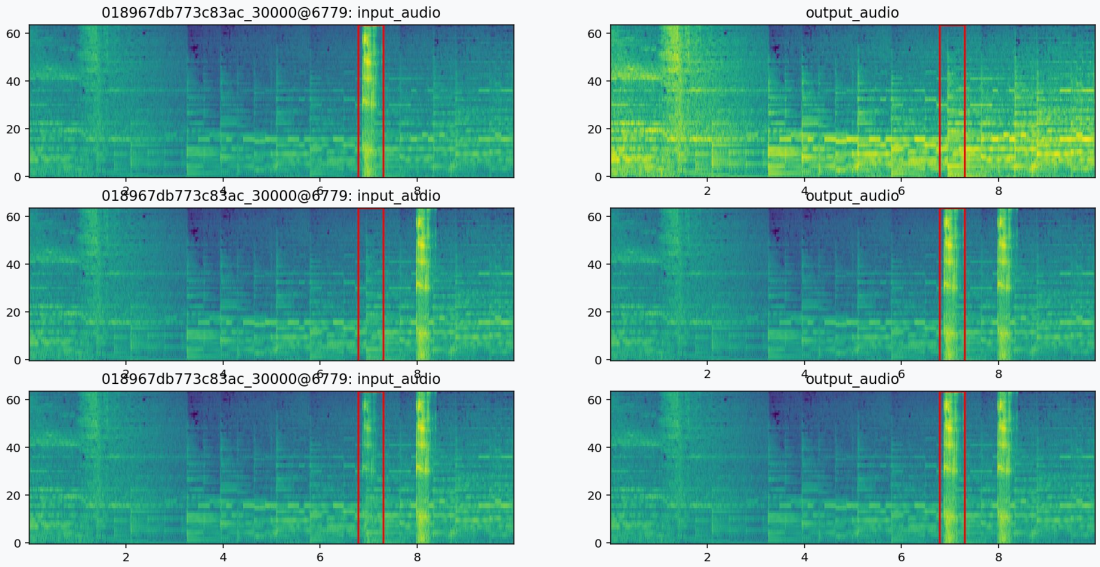
- Initial results: **Action** (and **Class**) have little impact
  - Model can infer them from input?



Decoy Data

Each sample input has **one event** (or decoy) at 10 dB TBR

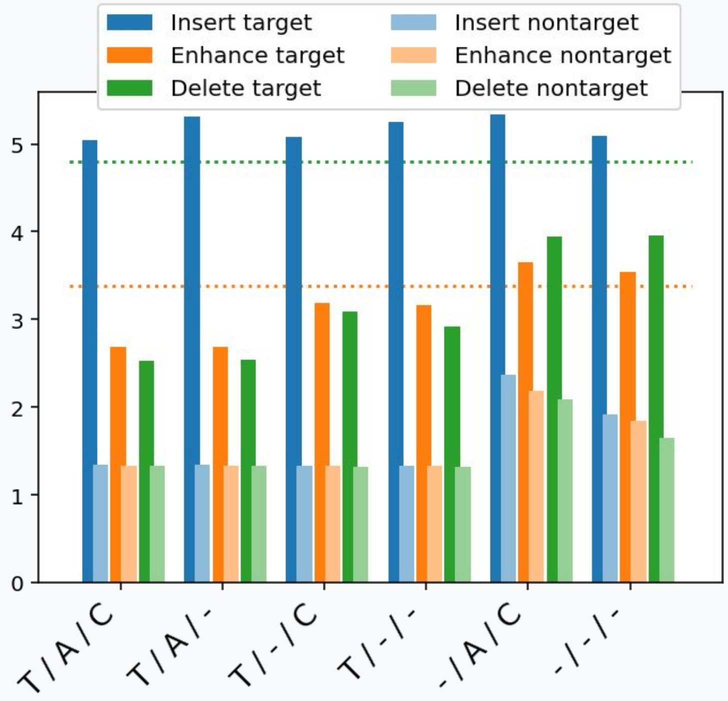
- **Delete**: Output has no event
- **Insert**: Output has a second event (event in input is **decoy**)
- **Enhance**: Input has decoy plus -6dB TBR target event



⇒ Model cannot guess action from input

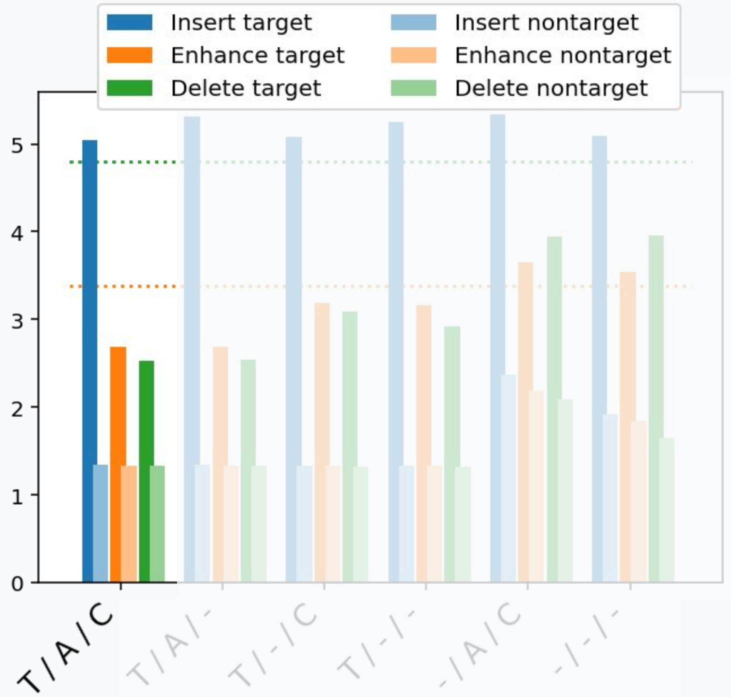
- Model was trained with 0 .. 2 edits per sample

Results: **MSD**



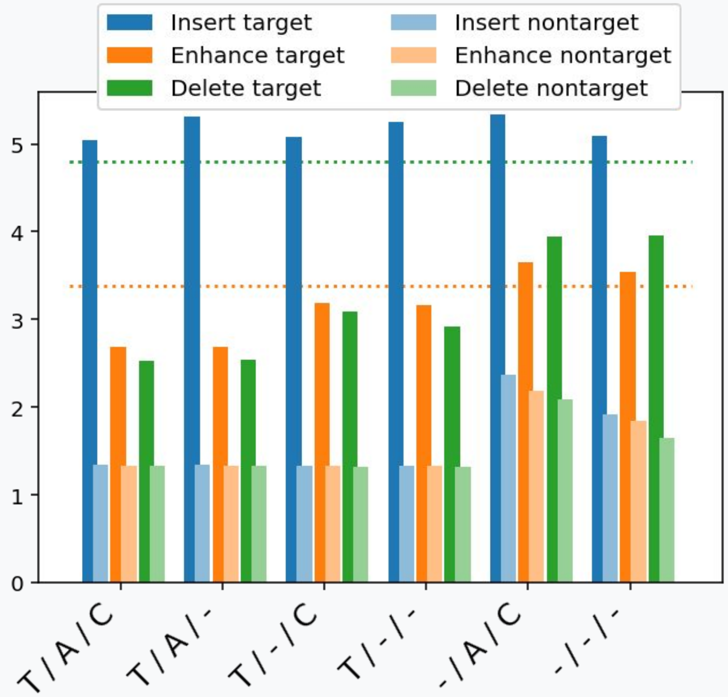
***T**iming*
***A**ction*
***C**lass*

Results: **MSD - Full-Conditioning**



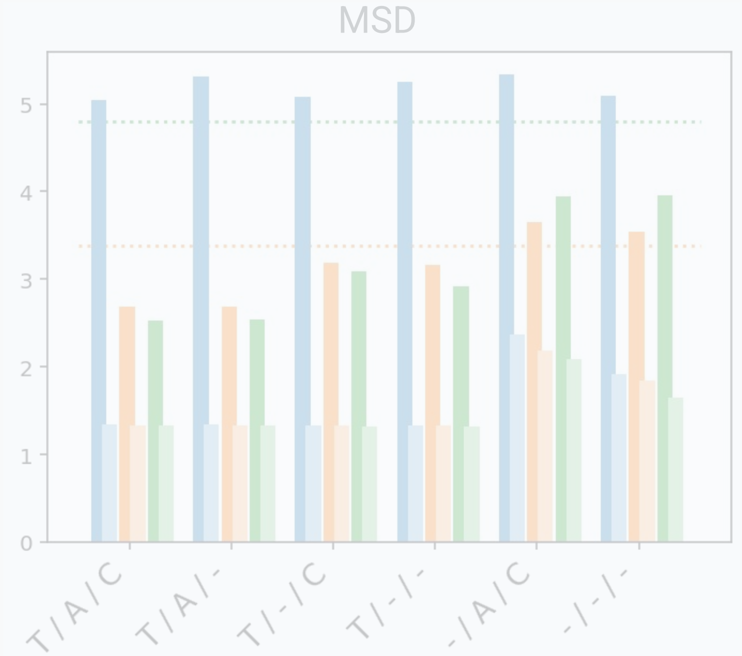
- Unprocessed baseline (dotted): **Insert** & **Delete** same (4.8), **Enhance** better (3.3)
- Unmodified region (nontarget, pale bars): “Model distortion” floor (1.3)
- Modified region (target, dark bars): **Delete** ~ **Enhance** (2.5), **Insert** much worse (5.0)

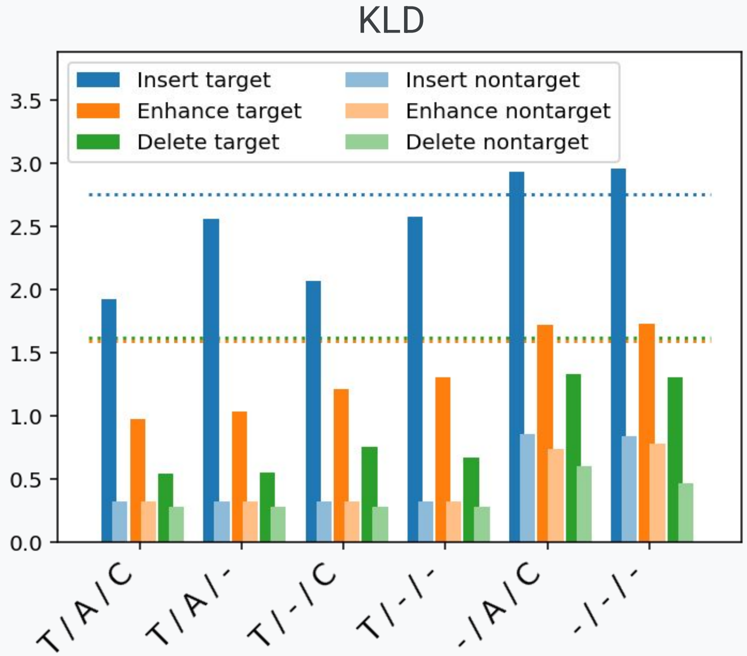
Results: **MSD - Conditioning Ablation**



- Remove Class (T / A / -) ⟶ **Insert** worse, **Delete** & **Enhance** unchanged?
- Remove Action (T / - / C) ⟶ **Delete** & **Enhance** worse (confused?)
- Remove Timing (- / A / C) ⟶ **Delete**, **Enhance** (and nontarget) worse

Results: **KLD**



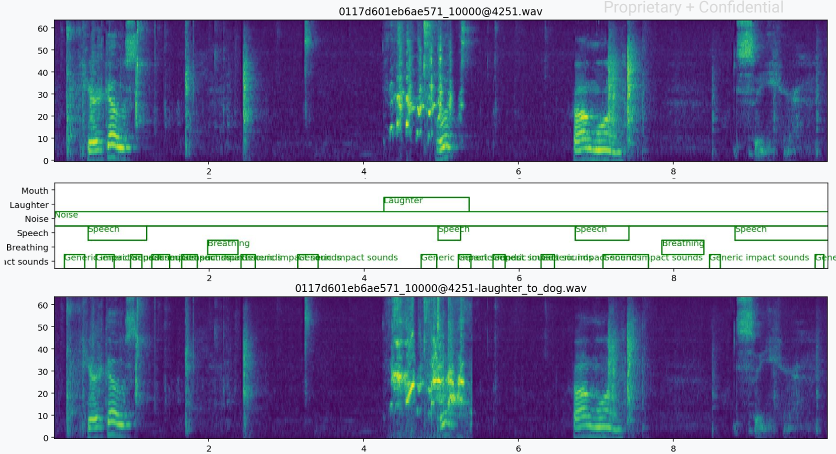


- Unprocessed baseline (dotted): **Delete** ≈ **Enhance**, but **Insert** much worse
- Processed: **Delete** better than **Enhance**
- **Insert** benefits from Class

Future Work

- Richer **control** of generation
  - More structured attributes (loudness, pitch, reverberation)
  - Richer text-to-audio generation
    - training data?
- Broader conditioning e.g. **Video**
  - Audio-Video joint generation
- Sound **Transformation** ...

Sound Transformation via **Cross-Class Enhancement**



- Input audio contains a given source_class
- Run an enhancement-only Recomposer model to `enhance <destination_class>` at times when the **source_class** occurs.
  - The model was never trained to do this

Clip ID	*0239dc6ce0480dc9_30000@4598*	*0253eeff2c4b4f68_230000@5512*	*0117d601eb6ae571_10000@4251*	*33ad41049a6fbf1f_30000@3207*
Source class	Human locomotion	Cough	Laughter	Ring
Destination class	Digestive	Dog	Dog	Bird
Input audio				
Output audio				

Conclusions

- Autoregressive Encoder-Decoder models can **edit scene details**
  - **Event roll** as a precise way to specify timing
- **Ablations** reveal complex interactions
  - Each part of the conditioning has a different effect
- Future work:
  - Richer **control** of generated sound events
  - Additional conditioning, e.g. **video**