Mono-to-stereo Through Parametric Stereo Generation

Joan Serrà, Davide Scaini, Santiago Pascual, Daniel Arteaga, Jordi Pons, Jeroen Breebaart, & Giulio Cengarle

Dolby Laboratories

INTRODUCTION

Upmixing from mono to stereo is still a need

- Historical or originally mono recordings
- "No-width" stereo recordings (e.g., mobile phone)
- Mono-based processing (including deep learning)

The music stereo image, a great generative task!

- Highly creative/artistic (e.g., instrument panning)
- Highly subjective → One-to-many mapping

Existing approaches

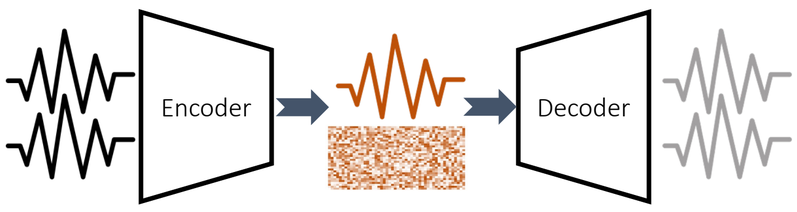
- Based on decorrelation (time delays, all-pass filters, etc.)
  - Limited effect/width
  - Cannot spatially separate individual elements in the mix
- Based on source separation
  - Artifacts + Restrictive (e.g., number and types of sources)
  - Need automatic post-processing (e.g., stereo sources, panning)

CONTRIBUTIONS

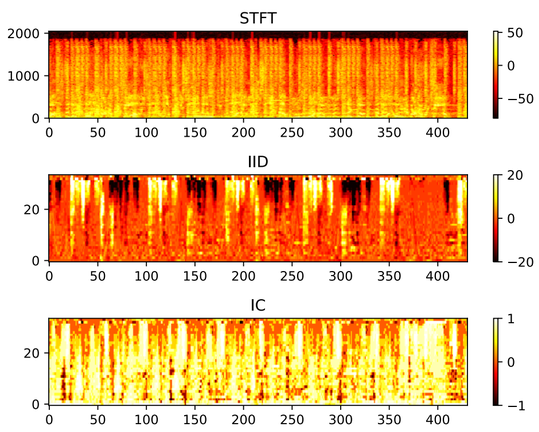
- ➢ Model it as a **generative problem**!
- ➢ We propose to upmix in the **parametric stereo** space
- ➢ We propose to leverage machine learning techniques
  - Classical: **nearest neighbor(s)**
  - Deep learning: **autoregressive** and **masked token modeling** approaches
- ➢ Objective **measures** and subjective **protocol**
- ➢ Discussion

METHOD

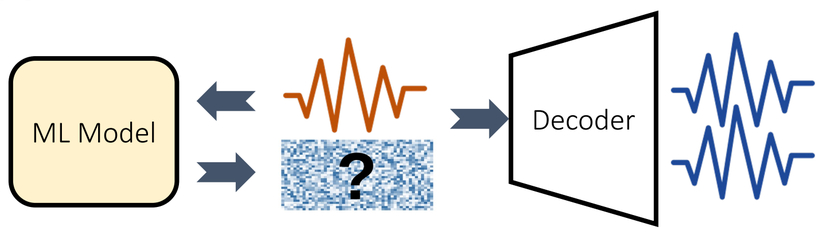
Parametric Stereo (PS) [1]



- Classic coding technique (transmit mono audio + parameters)
- Parameters are frame-based and multi-band
- Based on channel intensity difference (IID) and channel correlation (IC)
- Quantized
- If done right, almost inaudible artifacts



PS Generation



ML Model – Classic

- Nearest neighbor (PS-NN)
  - For each frame (+ context), retrieve most similar from training data, using spectral energies for similarity
  - Take the corresponding PS parameters as prediction
  - Smooth the obtained PS parameter sequence

ML Model – Generative

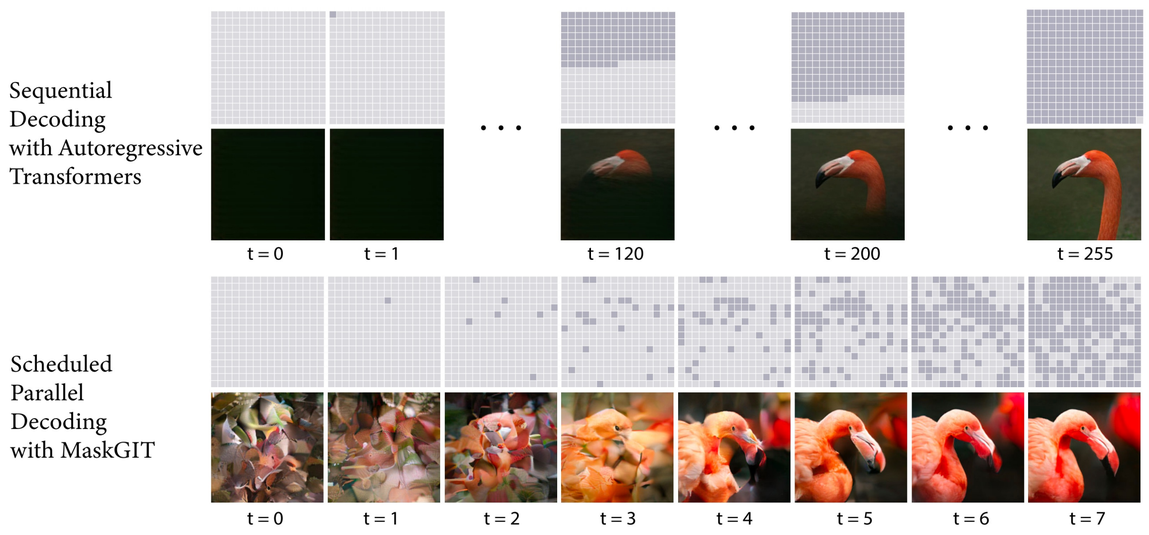
- Autoregressive (PS-AR) [2]
  - Transformer-based (BERT [3]), conditioned on mono spectrogram

$$\mathbf{H} = \phi(\mathbf{S}) + \sum_{i=1}^{B} \xi_i(\mathbf{Q}_{i,:}),$$

  - Classifier-free guidance + Weighted loss

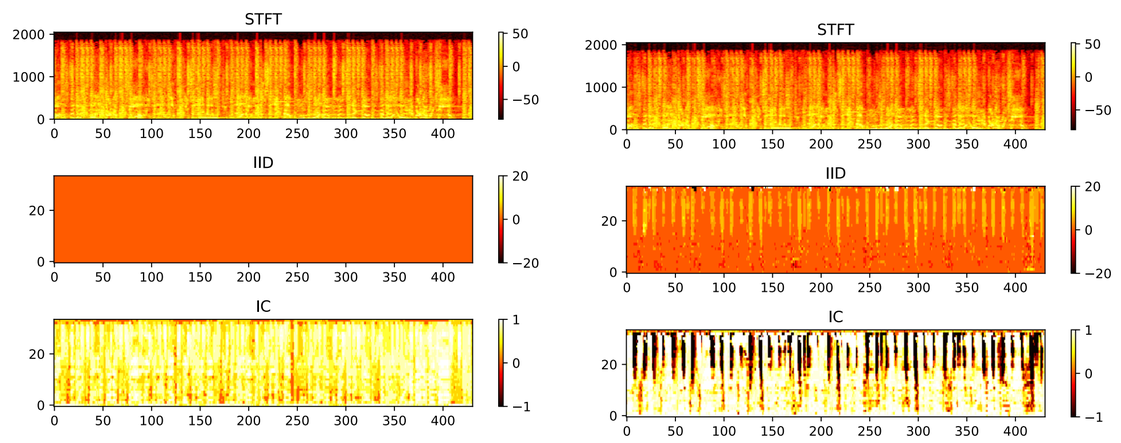
$$\mathbf{U} = (1+\gamma)\mathbf{U}^{\text{cond}} - \gamma\mathbf{U}^{\text{uncond}}. \qquad w = 1 + \lambda\sigma\left(\left[\mathbf{P}^{\text{IID}}\right]_{\pm\epsilon}\right) + \sigma(\mathbf{P}^{\text{IC}}),$$

- Masked token modeling (PS-MTM)
  - Same setup as PS-AR
  - Sampling based on MaskGIT [4]



RESULTS

Preliminary: Regression vs. Generative



Subjective Evaluation: Expert Listeners

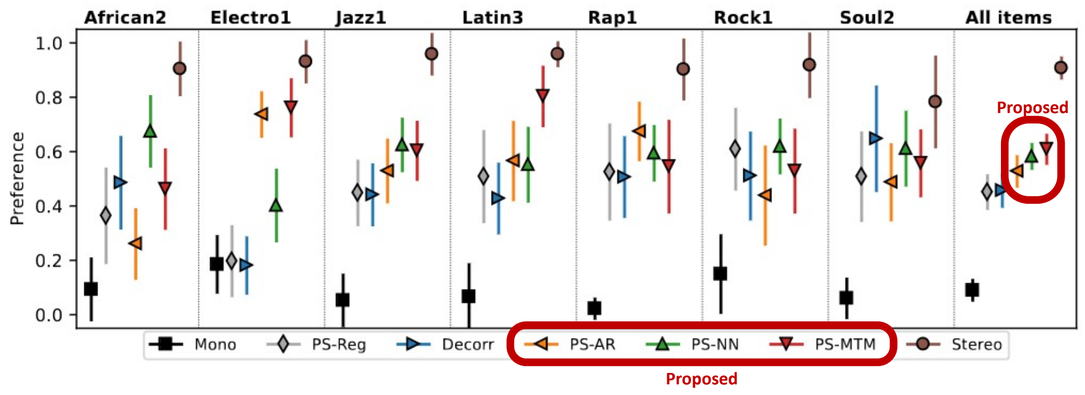


Figure 1. Preference results for the items included in the subjective test (Sec. 4.3). Markers indicate average values and vertical bars indicate the 95% confidence interval associated to them.

Objective Evaluation: Metrics, Significance, Runtime...

Approach	$E_{\text{min}}\downarrow$	$D_{\text{F}}\downarrow$	Preference ↑
Mono	0.104	20.89	0.090 ± 0.042
PS-Reg	0.069	8.11	0.451 ± 0.066
Decorr	0.093	8.32	0.457 ± 0.064
PS-AR	0.074	0.62	0.527 ± 0.060
PS-NN	0.089	3.08	0.582 ± 0.057
PS-MTM	0.068	0.59	0.608 ± 0.050
Stereo	0.000	0.03	0.908 ± 0.042

Table 1. Results for the objective (E_{min}, D_{F}) and subjective (Preference $\pm$ 95% confidence interval) evaluations.

	PS-Reg	Decorr	PS-AR	PS-NN	PS-MTM	Stereo
Mono	✓	✓	✓	✓	✓	✓
PS-Reg		✗	✗	✓	✓	✓
Decorr			✗	✓	✓	✓
PS-AR				✗	✗	✓
PS-NN					✗	✓
PS-MTM						✓

Table 2. Pairwise statistical significance for the case of all test items (12 subjects times 7 excerpts, see Sec. 4.3). The obtained p-value threshold is 0.0053.

Approach	Learnable parameters	RTF ↓ CPU	RTF ↓ GPU
Decorr	0	0.25	n/a
PS-Reg	30.1 M	0.32	0.21
PS-NN	34.0 M†	0.82	n/a
PS-MTM	34.5 M	5.81	0.33
PS-AR	34.5 M	255.87	8.38

Table 3. Number of learnable parameters and average real-time factor (RTF). Superscript † indicates an estimation of 0.5 M key-value pairs with $B = 34$ bands (Sec. 3.1). RTFs are measured on a Xeon(R) 2.20 GHz CPU and on a GeForce GTX 1080-Ti GPU.

References:
[1] Breebart et al. (2005), *Parametric coding of stereo audio*, EURASIP Journal on Advances on Signal Processing.
[2] Chen et al. (2018), *PixelSNAIL: an improved autoregressive generative model*, ICML.
[3] Radford et al. (2018), Improving language understanding by generative pre-training, Technical Report.
[4] Chang et al. (2022), *MaskGIT: masked generative image transformer*, CVPR.