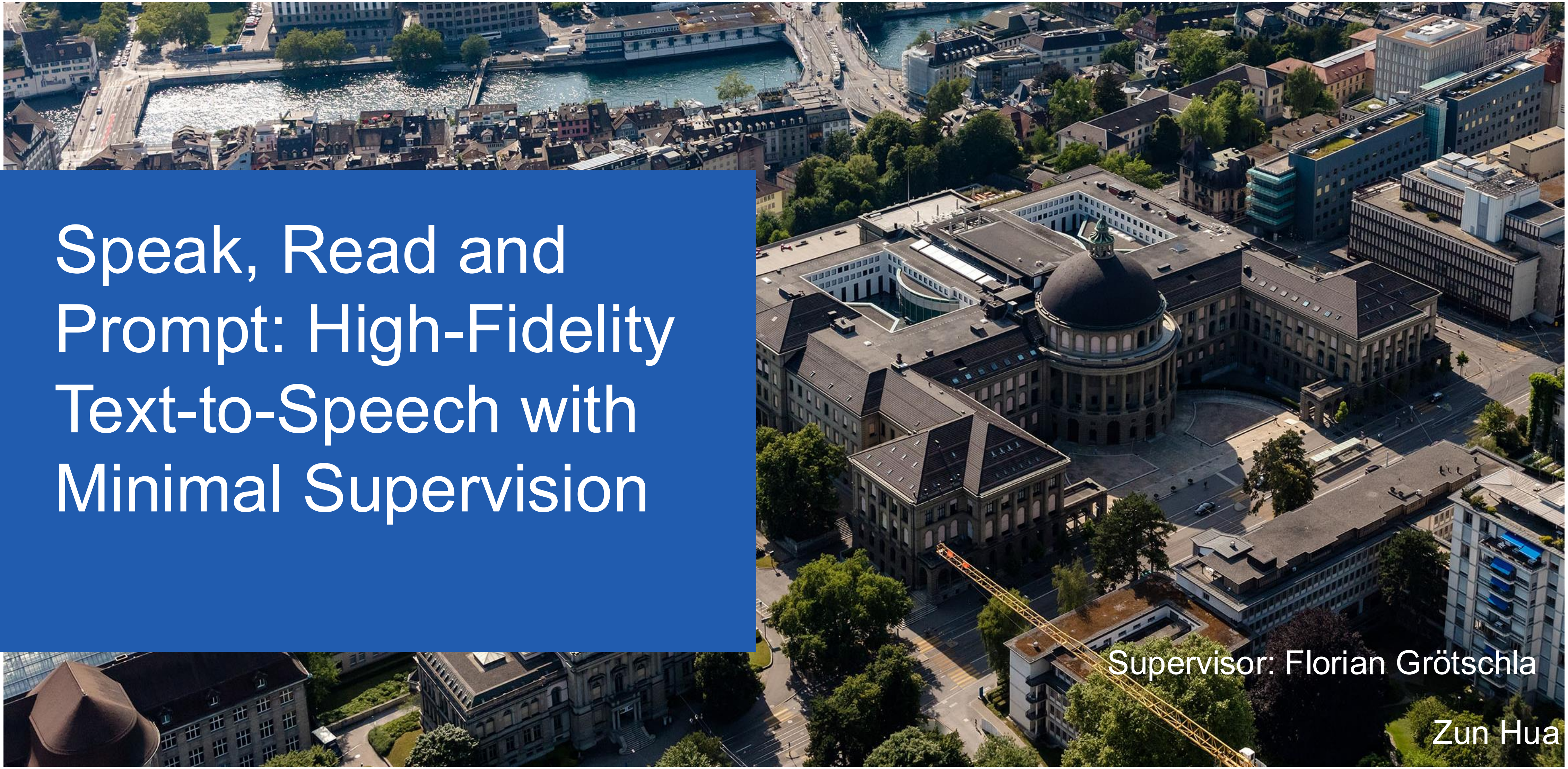
ETH zürich

Seminar in Deep Neural Networks

Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision



Supervisor: Florian Grötschla

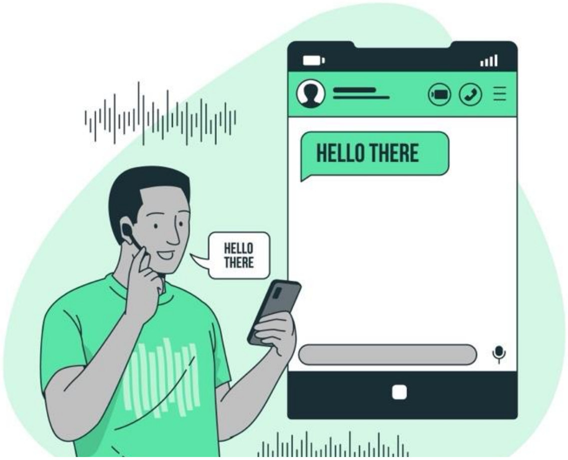
Zun Hua

ETH zürich

Seminar in Deep Neural Networks

Speak, Read and Prompt (SPEAR-TTS)

- High-Fidelity Text-to-Speech with Minimal Supervision



Eugene Kharitonov[1], **Damien Vincent**[2], **Zalán Borsos**[2], **Raphaël Marinier**[1], **Sertan Girgin**[1], **Olivier Pietquin**[1], **Matt Sharifi**[2], **Marco Tagliasacchi**[2], **Neil Zeghidour**[1]

[1]Google, France [2]Google, Switzerland

{kharitonov,damienv,neilz}@google.com

Organisationseinheit verbal
optional auf 2 Zeilen

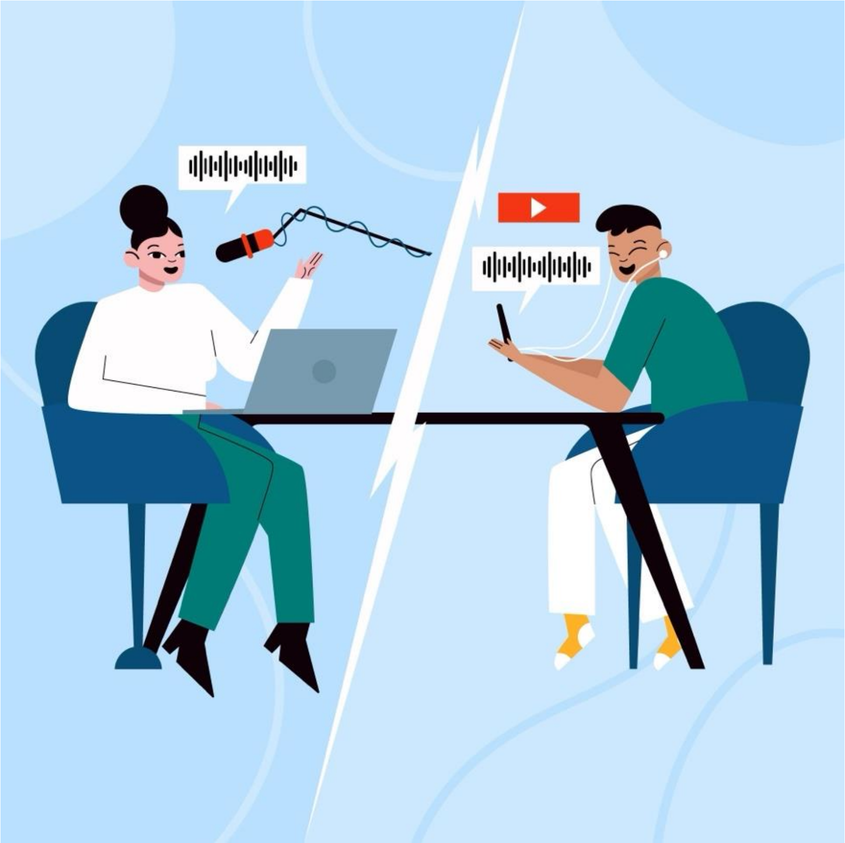
High-Fidelity Text-to-Speech with Minimal Supervision

Can we build great TTS with unlabeled data?

Motivation: Why Do We Need Data-Efficient TTS?

- Traditional TTS requires large labeled datasets
- Hard to adapt to new speakers & languages

- Audio-only data is abundant and untapped



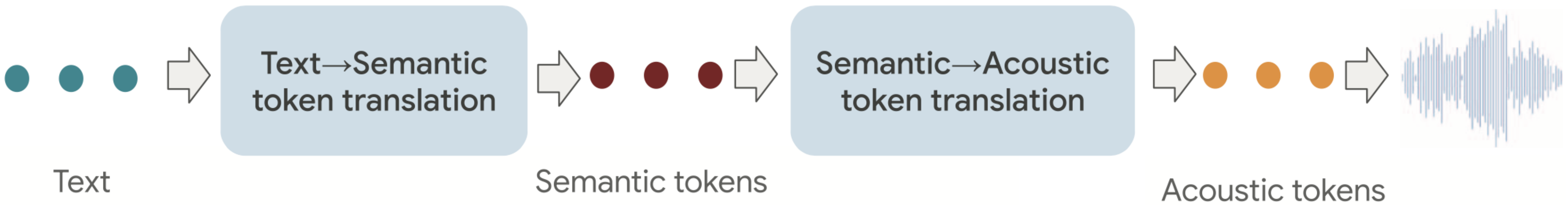
How Others Approach TTS

1. Tacotron (2017)
2. FastSpeech / FastSpeech2 (2019-2020)
3. VALL-E (2023)
4. AudioLM (2022)

SPEAR-TTS

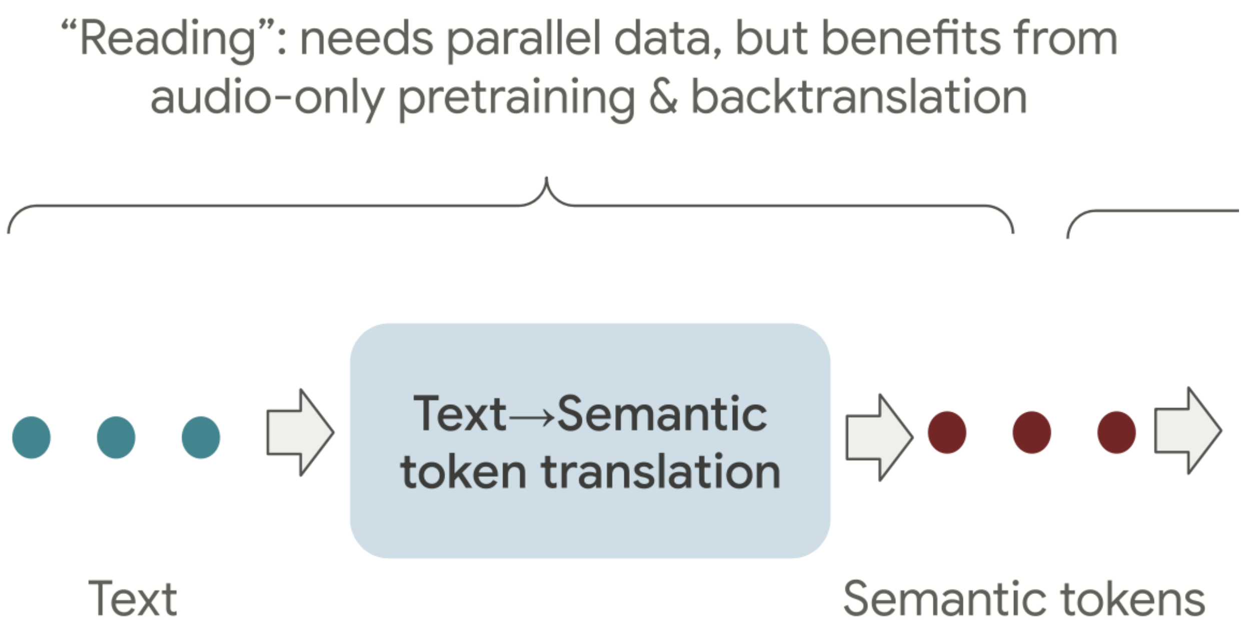
Two-Stage Architecture: Read & Speak

Two-Stage Architecture: Read & Speak



Two-Stage Architecture: Read & Speak

- Stage 1: Text → Semantic Tokens



Stage 1: What happens with tokenization?

Training Time:

•**(text, audio)** pairs.

•audio passed through **pretrained tokenizer** (HuBERT + K-Means).

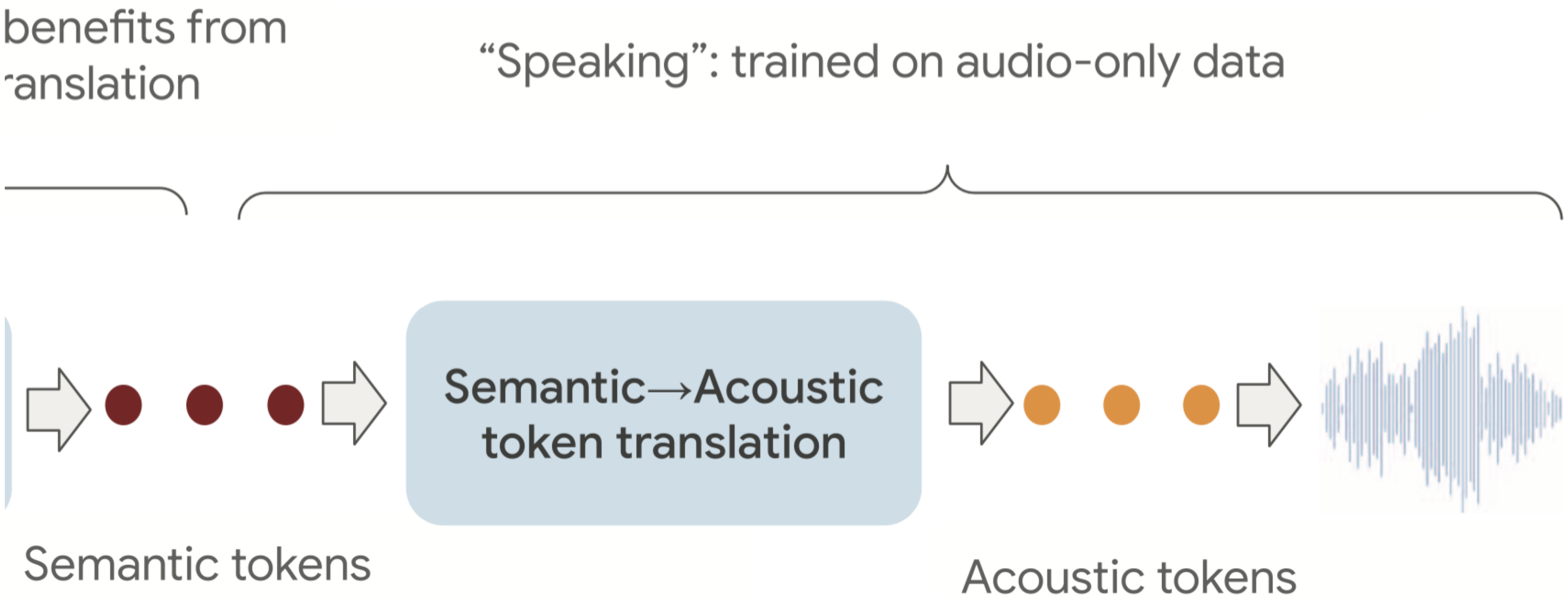
•**converts the audio** into a sequence of **semantic tokens**.

•Now you have:

➔ **Text** ↔ **Semantic tokens**

Two-Stage Architecture: Read & Speak

- Stage 2: Semantic Tokens → Acoustic Tokens → Audio (using soundstream)



Stage 2: What happens with tokenization?

Training Time:

- **raw audio**.
- audio through a **pretrained neural codec — SoundStream**.
- **audio into acoustic tokens**.
- prosody, speaker style, accents, emotions, etc.
- **➔ Semantic tokens → Acoustic tokens**

Motivation for Data Efficiency Techniques

🚨 The Problem: Labeled Data is Expensive

Traditional TTS systems (like Tacotron or FastSpeech2) need **hundreds of hours** of this data

🎤 But... Unlabeled Audio is Everywhere

How can we use this abundant, unlabeled audio to train powerful TTS models?

⚡ The Solution: Data Efficiency Techniques

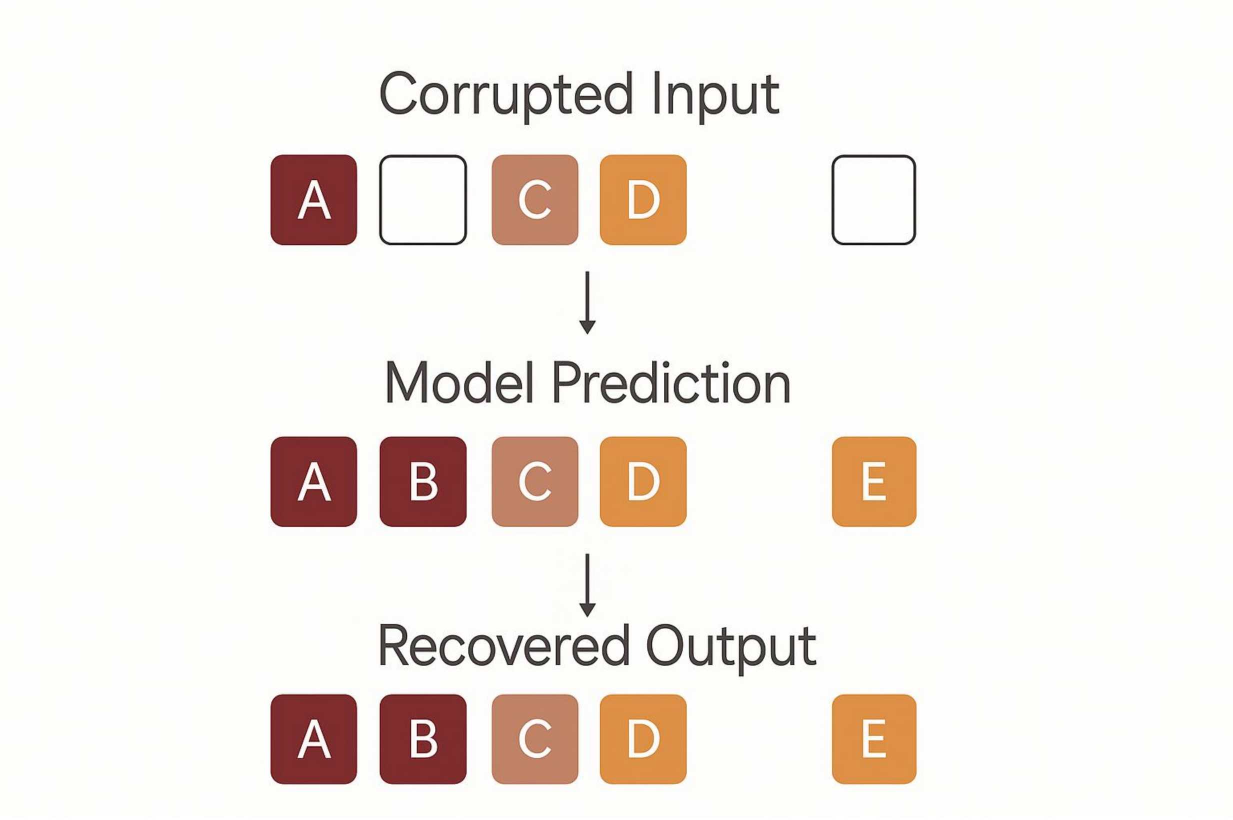
SPEAR-TTS introduces clever strategies to **reduce reliance on supervised data**

Data Efficiency Techniques

- BART-style pretraining
- Backtranslation for synthetic data
- Example prompting for voice control

Data Efficiency Techniquesy: BART

- The inp
  or shuff
- The mo
  uncorr



, deleting,

Reference:
Lewis et al., "*BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation*", arXiv:1910.13461, 2020.

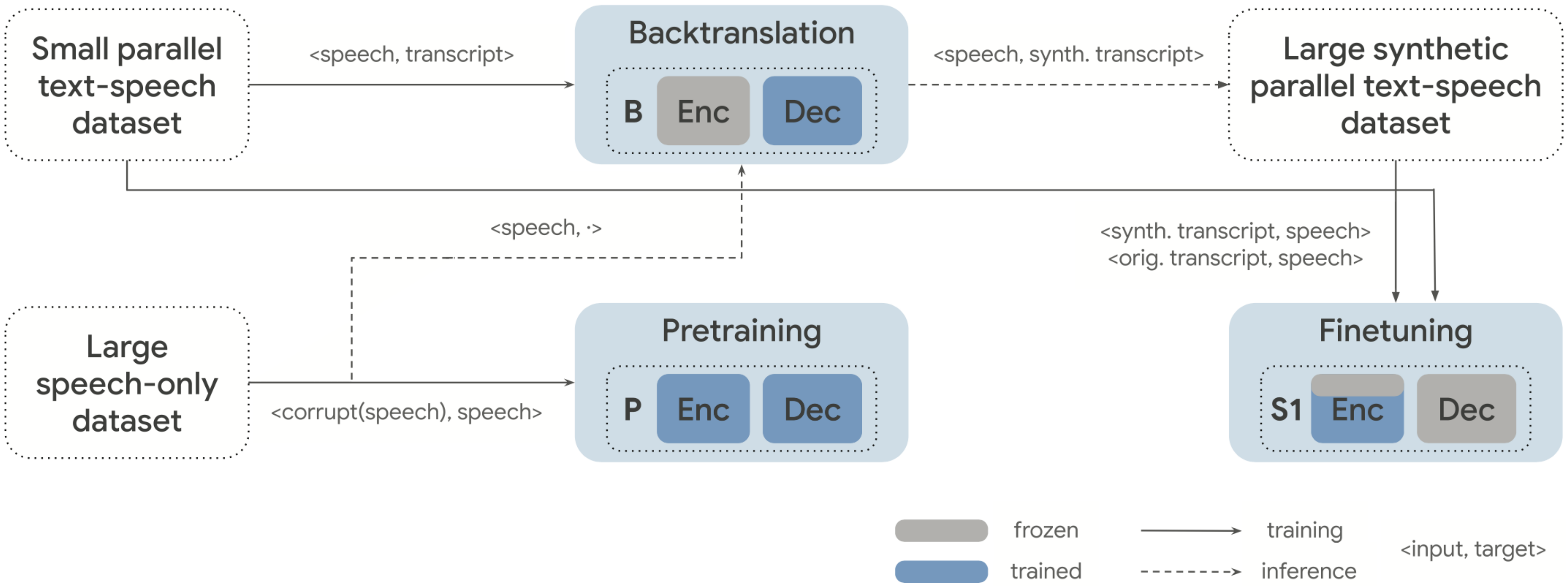
Data Efficiency Techniques

- BART-style pretraining
- Backtranslation for synthetic data
- Example prompting for voice control

Data Efficiency Techniques: Backtranslation

- Leverages unlabeled audio-only data
- Generate synthetic text from audio using a reverse model
- Create new pseudo-parallel data (Text ↔ Audio)
- Benefits from the ASR models

Data Efficiency Techniques: Backtranslation



Data Efficiency Techniques: Backtranslation

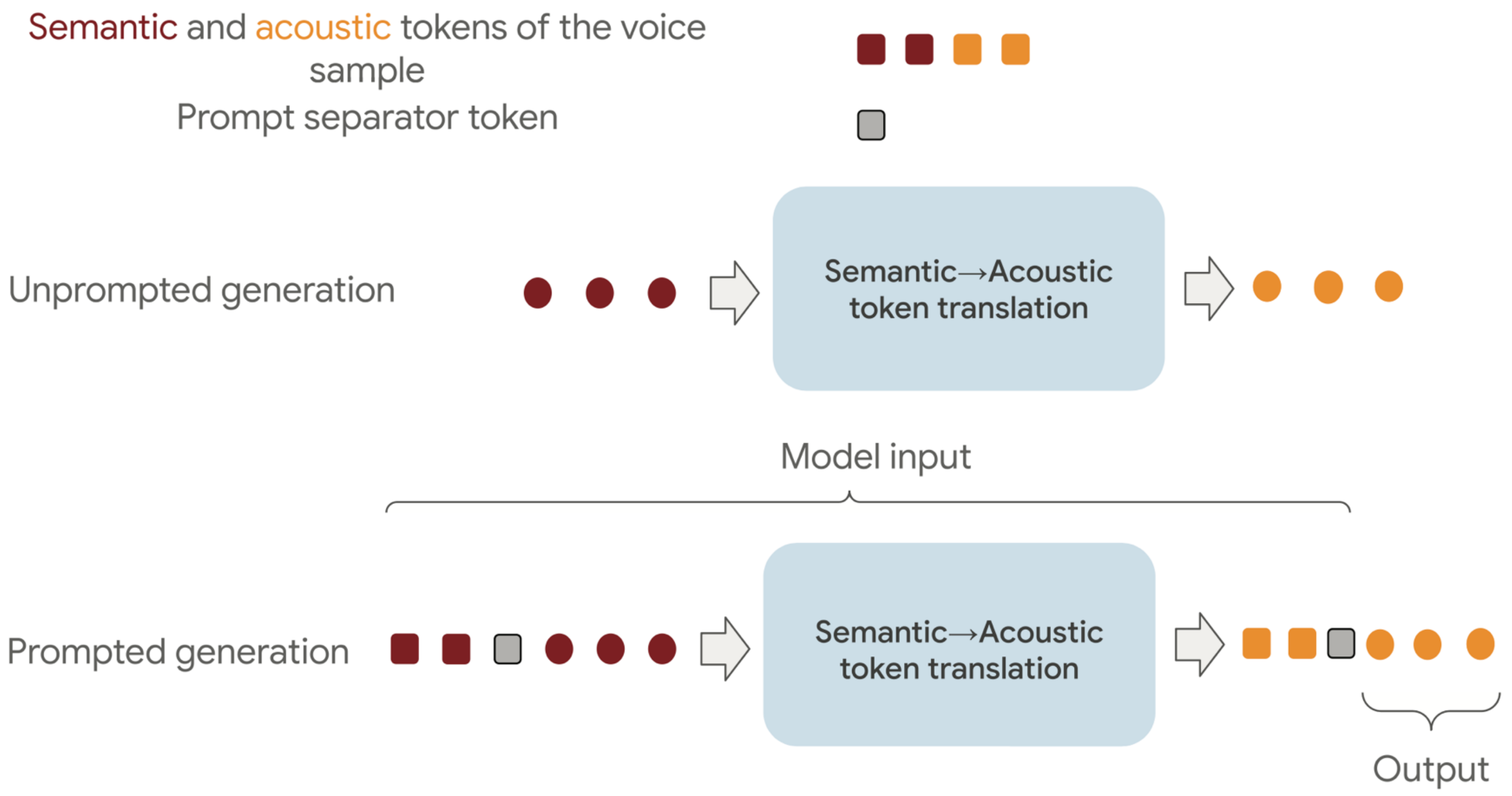
📊 Example Table: Types of Data Pairs in TTS

Type of Data	Text	Audio	Notes
Supervised	"Hello, how are you?"	🎙 Human speech	Real ground truth
Unsupervised	—	🎙 Human speech	No text available
Synthetic Pair	"Hi, how are you?"	🎙 Human speech	Text generated via backtranslation

Data Efficiency Techniques

- BART-style pretraining
- Backtranslation for synthetic data
- Example prompting for voice control

Data Efficiency Techniques: voice control with prompting



Data Efficiency Techniques: voice control with prompting – Demo (the prompt)

Data Efficiency Techniques: voice control with prompting – Demo (the outcome)

Text: "I see a crowd in one corner of the garden everybody standing still and looking up"

Data sets used

Name	Size, hours	Transcripts used?	Used for
LibriLight	60k	✗	Acoustic & semantic tokens, pretraining $\mathcal{S}_1$, and training $\mathcal{S}_2$
LJSpeech	0.25..24	✓	Finetuning $\mathcal{S}_1$ for backtranslation
LibriTTS train	551	✗	Source of backtranslated data
LibriSpeech test-clean (shorter than 10s)	3	✓	Intelligibility evaluation
LibriSpeech train-clean + test-clean	105	✗	Training the voice classifier used in the evaluation

Table 1: **Datasets used in the paper.** For each dataset, we highlight its size, use, and whether textual transcripts are used.

Data sets used

📚 1. LibriTTS *(Supervised Data)*

- **English audiobook recordings** from the public domain (LibriVox project).
- Contains **text + matching audio** (parallel data).
- Designed specifically for TTS tasks.
- High-quality, but limited in size

Data sets used

🎙️ **2. LibriLight** ***(Unsupervised Data)***

•**60,000+ hours** of English audiobook recordings — but with **no transcripts**.
•**audio-only data**.
•For Stage 2 and for backtranslation.

Data sets used

🗣 3. VCTK

•**different English speakers** (diverse accents).

•Contains **text + audio**

•used for evaluating **speaker adaptation**.

Metrics

- CER (Character Error Rate): Faithfulness to input text using ASR transcription
(e.g., 0.98% on original audio)

- Voice Preservation: Measures consistency between prompt and generated speech using a speaker classifier

Metrics

- MOS (Mean Opinion Score): Human-rated naturalness and audio quality on a 1–5 scale, the best evaluation, hard to be objective

Score	Description	Meaning
5	Excellent	Completely natural
4	Good	Mostly natural
3	Fair	Somewhat unnatural
2	Poor	Unnatural, noticeable issues
1	Bad	Completely unnatural

Key Results

Metric	SPEAR-TTS	Reference
CER	2.21%	(15 min parallel data)
MOS	4.96	4.92 (Ground Truth)
Speaker Accuracy	92.4%	Prompted generation
Data Efficiency	240,000× less data	Compared to VALL-E

SPEAR-TTS vs. Previous Models

- FastSpeech2-LR: Needs more data, lower quality
- VALL-E: Requires huge parallel datasets
- SPEAR-TTS: Data-efficient, flexible, high-quality

SPEAR-TTS outperforms older models in low-resource settings.

Demo-time!

- Compare samples from SPEAR-TTS vs FastSpeechLR
- Highlight the MOS gap (4.75 vs. 3.35)

https://google-research.github.io/seanet/speartts/examples/

Demo

"I will go," said Beth, a little frightened at the passionate appeal.				

Demo

Get my things ready, get yours ready. We're departing in two hours."				

Demo

"Nobody but me, till now, has ever heard.				

Strengths & Limitations

+ Data efficiency
+ zero-shot
- large audio-only datasets
- limited to English

WhisperSpeech: Built on SPEAR-TTS, Improved

- Two-stage architecture (SPEAR-TTS)
- Whisper ASR for better backtranslation
- Stronger zero-shot, multilingual
- Less parallel data needed
- More robust semantic understanding

Ethical Concerns

- Voice Cloning & Deepfakes
- Lack of Consent
- Misinformation & Fraud
- Accountability & Detection

✅ **How to Address These Concerns:**
•Develop **detection mechanisms**
•Implement **consent-based systems**
•Establish **legal frameworks**
•Raise **public awareness**

No code/model released due to misuse risk

Questions or thoughts?

Data Efficiency Techniques: Backtranslation

Backtranslation is used in Stage 1 ("Reading")
Because Stage 1 needs to learn how to map **text →** **semantic tokens**, but there’s limited real parallel data.

So, they:

1. Take **audio-only data**.
2. Use a reverse model (**semantic tokens → text**) to generate **synthetic text**.
3. Now they have a **synthetic text + real semantic tokens pair** to train Stage 1.

- This creates **pseudo-parallel data** for Stage 1.