SONY

BigVSAN: Enhancing GAN-based Neural Vocoders with Slicing Adversarial Network

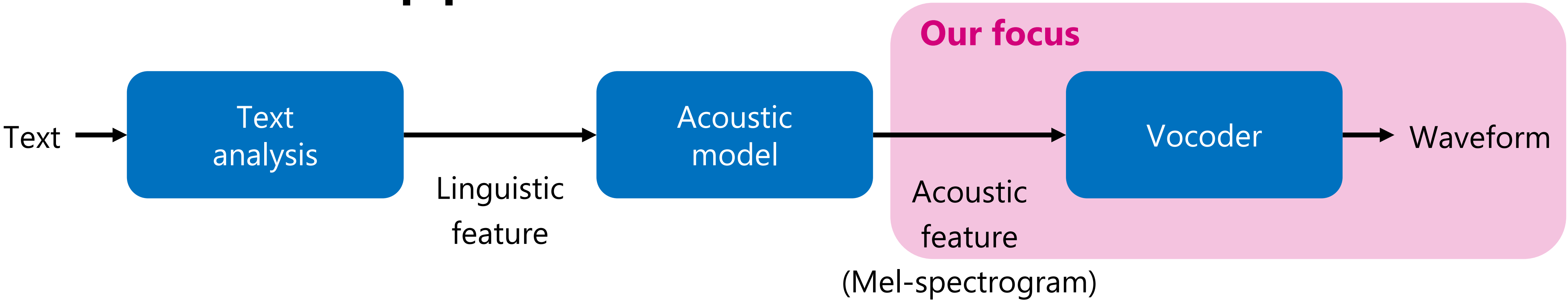
Takashi Shibuya, Yuhta Takida, Yuki Mitsufuji

ICASSP 2024

Sony AI

Introduction: Text-to-speech (TTS) models

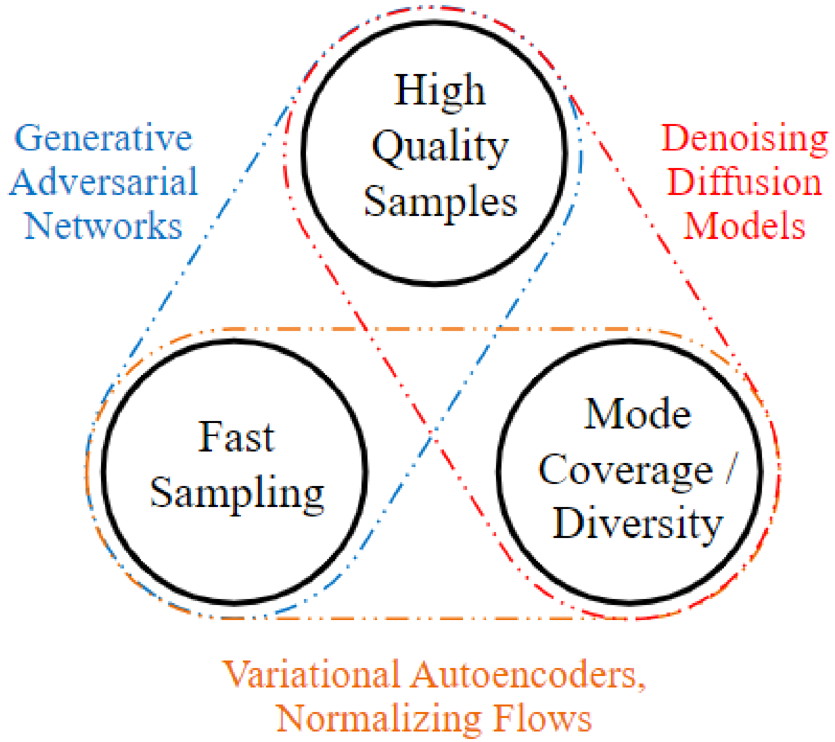
■ Conventional TTS pipeline



■ End-to-end TTS

Introduction: Deep generative models

■ Trilemma of generative models



Xiao et al., "Tackling the Generative Learning Trilemma with Denoising Diffusion GANs," ICLR 2022.

- Existing basic generative models compromise between three key requirements

In image generation or text-to-audio generation,

- Diffusion models are popular because "quality" and "diversity" are important. Many methods of accelerating diffusion models are being studied.

In vocoding,

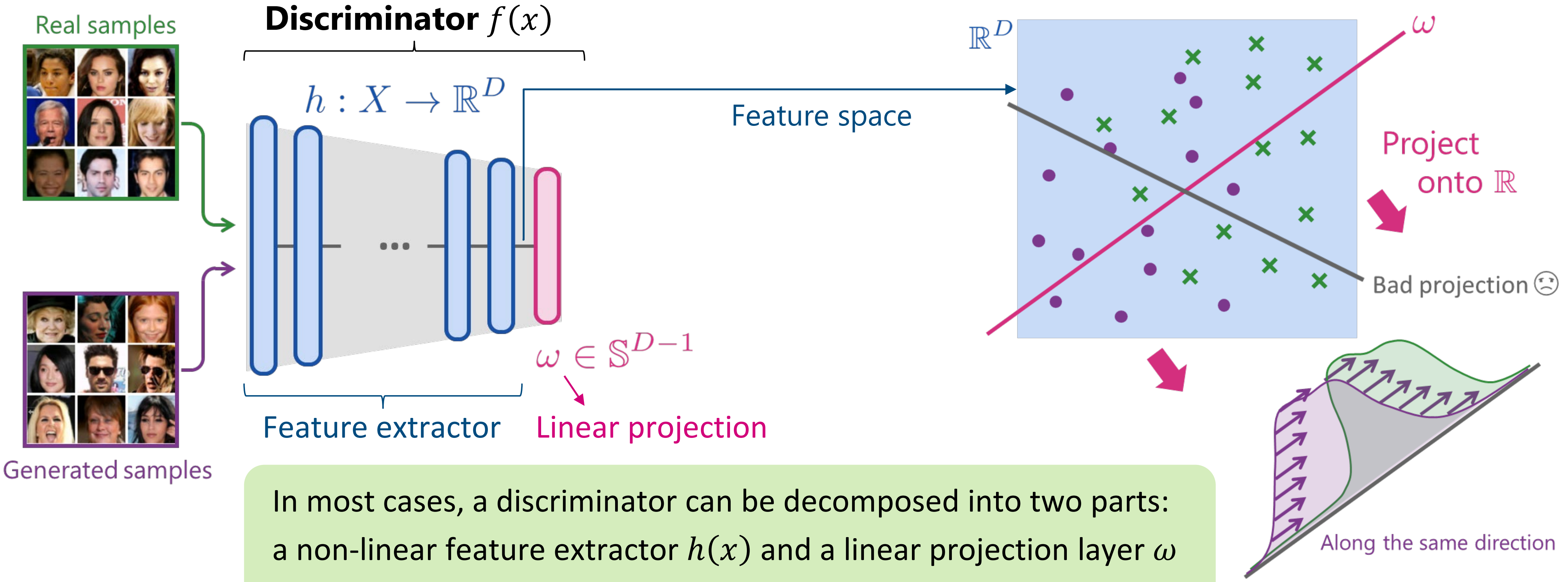
- "Diversity" is not so important because a vocoder is required to synthesize a waveform corresponding to a given mel-spectrogram

⇒ **GAN** is still a reasonable choice

e.g., BigVGAN, HiFi-GAN, Parallel WaveGAN, etc.

Problem in GANs [Takida et al., ICLR 2024]

■ **Decompose a discriminator $f(x)$ into $f(x) = \langle \omega, h(x) \rangle$**



In most cases, a discriminator can be decomposed into two parts: a non-linear feature extractor $h(x)$ and a linear projection layer ω

However,

most conventional GAN frameworks fail to find the projection that can best distinguish real and generated data samples

Takida et al., "SAN: Inducing Metrizability of GAN with Discriminative Normalized Linear Layer," ICLR 2024 (to appear).

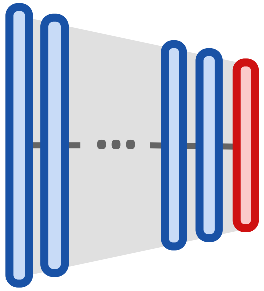
SAN (slicing adversarial network) [Takida et al., ICLR 2024]

φ: feature extractor's parameter
θ: generator's parameter

Conventional GAN

Objective for discriminator: $\max_{\varphi,\omega} \mathcal{V}_{\mathrm{GAN}}(\varphi,\omega;\theta) = \mathrm{E}_{p_X}[R_1(\langle \omega, h_\varphi(x)\rangle)] + \mathrm{E}_{p_S}[R_2(\langle \omega, h_\varphi(g_\theta(s))\rangle)]$

Objective for generator: $\min_{\theta} \mathcal{V}_{\mathrm{GAN}}(\theta;\varphi,\omega) = \mathrm{E}_{p_S}[R_3(\langle \omega, h_\varphi(g_\theta(s))\rangle)]$



$f_\phi(x) = \langle h_\varphi(x), \omega\rangle$

SAN

The objective for discriminator is modified

Objective for discriminator: $\max_{\varphi,\omega} \mathcal{V}_{\mathrm{SAN}}(\varphi,\omega;\theta) = \mathrm{E}_{p_X}[R_1(\langle \omega^-, h_\varphi(x)\rangle)] + \mathrm{E}_{p_S}[R_2(\langle \omega^-, h_\varphi(g_\theta(s))\rangle)]$

$-\mathrm{E}_{p_X}[R_3(\langle \omega, h_\varphi^-(x)\rangle)] + \mathrm{E}_{p_S}[R_3(\langle \omega, h_\varphi^-(g_\theta(s))\rangle)]$

Objective for generator: $\min_{\theta} \mathcal{V}_{\mathrm{SAN}}(\theta;\varphi,\omega) = \mathrm{E}_{p_S}[R_3(\langle \omega, h_\varphi(g_\theta(s))\rangle)]$

$()^-$: stop-gradient operator

SAN outperforms GAN in many combinations of architectures and image datasets.

SAN achieved SOTA results on several image generation benchmarks.

Sony AI

Takida et al., "SAN: Inducing Metrizability of GAN with Discriminative Normalized Linear Layer," ICLR 2024 (to appear).

SAN-ify (Apply SAN to) GAN-based vocoders

All we propose in this paper is on this slide

SAN

Objective for discriminator: $$\max_{\varphi,\omega} \mathcal{V}_{\mathrm{SAN}}(\varphi,\omega;\theta) = \mathrm{E}_{p_X}\left[R_1\left(\langle \omega^-, h_\varphi(x)\rangle\right)\right] + \mathrm{E}_{p_S}\left[R_2\left(\langle \omega^-, h_\varphi(g_\theta(s))\rangle\right)\right]$$
$$-\mathrm{E}_{p_X}\left[R_3\left(\langle \omega, h_\varphi^-(x)\rangle\right)\right] + \mathrm{E}_{p_S}\left[R_3\left(\langle \omega, h_\varphi^-(g_\theta(s))\rangle\right)\right]$$

Objective for generator: $$\min_{\theta} \mathcal{V}_{\mathrm{SAN}}(\theta;\varphi,\omega) = \mathrm{E}_{p_S}\left[R_3\left(\langle \omega, h_\varphi(g_\theta(s))\rangle\right)\right]$$

$()^-$: stop-gradient operator

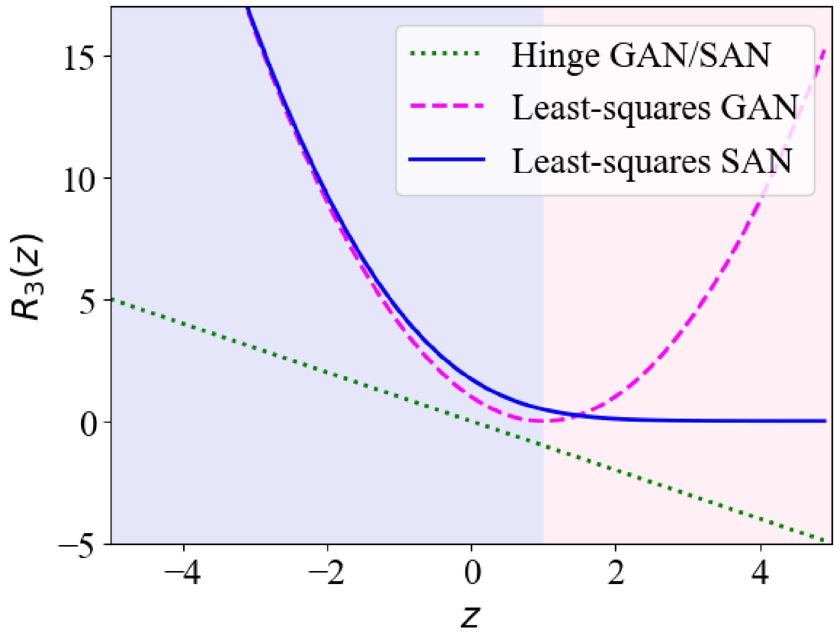
Most GAN-based vocoders rely on Least-squares GAN.

However, SAN requires R_3 to be a monotonically decreasing function.

In Least-squares GAN: $R_3(z) = (1-z)^2$ Not monotonic

In **Least-squares SAN** (ours): $\tilde{R}_3(z) = \varsigma(1-z)^2$ Monotonic

where $\varsigma(a) = \ln(1+e^a)$: softplus function



Sony AI

Experiments: large-scale vocoder training (1/2)

■ SAN-ify BigVGAN [Lee et al., ICLR 2023]

We trained a BigVGAN vocoder with **Least-squared SAN** on the LibriTTS dataset.

We followed their experimental setups, including data split, training hyperparameters, and evaluation protocol.

Result

M-STFT: spectral distance, **PESQ**: perceptual evaluation of speech quality, **MCD**: difference b/w mel cepstra
Periodicity: difference b/w periodicity scores, **V/UV F1**: F1 score of voiced/unvoiced classification

Table 1. Objective and subjective evaluations on LibriTTS. Objective results are obtained from a subset of its `dev` set. Subjective evaluations are based on a 5-scale mean opinion score (MOS) with 95% confidence interval (CI) from a subset of its `test` set.

Model	M-STFT (↓)	PESQ (↑)	MCD (↓)	Periodicity (↓)	V/UV F1 (↑)	MOS (↑)
Ground truth	–	–	–	–	–	3.81±1.89
BigVGAN (Lee *et al.* [20])	0.7997	4.027	<u>0.3745</u>	0.1018	0.9598	–
BigVGAN (our reproduction)	0.8382	3.862	<u>0.3711</u>	0.1155	0.9540	3.19±2.21
BigVSAN	**0.7881**	<u>4.116</u>	**0.3381**	<u>0.0935</u>	<u>0.9635</u>	<u>3.24</u>±1.95
BigVSAN w/ snakebeta activation	<u>0.7992</u>	**4.120**	0.4129	**0.0924**	**0.9644**	**3.43**±2.04

BigVSAN outperforms BigVGAN in terms of five objective metrics!

(*) We tried two activation functions for the generator

1) Snake activation: $f_\alpha(x) = x + \alpha^{-1}\sin^2(\alpha x)$ (Mentioned in the BigVGAN paper)

2) Snakebeta activation: $f_{\{\alpha,\beta\}}(x) = x + e^{-\beta}\sin^2(e^{\alpha}x)$ (Default in the BigVGAN repository)

Experiments: large-scale vocoder training (2/2)

■ SAN-ify BigVGAN [Lee et al., ICLR 2023]

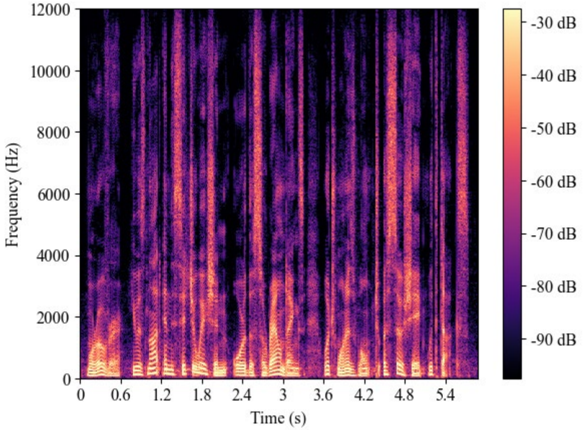
Result

Table 1. Objective and subjective evaluations on LibriTTS. Objective results are obtained from a subset of its `dev` set. Subjective evaluations are based on a 5-scale mean opinion score (MOS) with 95% confidence interval (CI) from a subset of its `test` set.

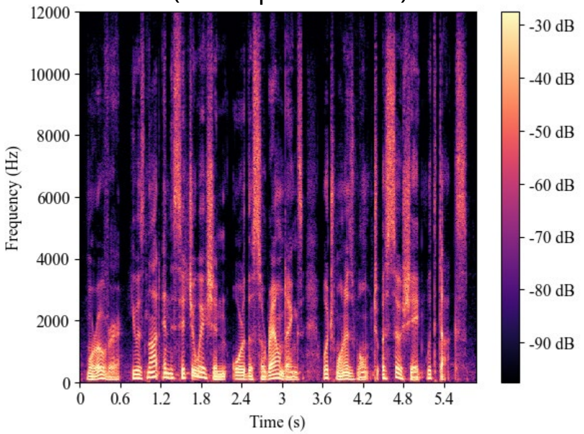
Model	M-STFT (↓)	PESQ (↑)	MCD (↓)	Periodicity (↓)	V/UV F1 (↑)	MOS (↑)
Ground truth	–	–	–	–	–	3.81±1.89
BigVGAN (Lee *et al.* [20])	0.7997	4.027	0.3745	0.1018	0.9598	–
BigVGAN (our reproduction)	0.8382	3.862	0.3711	0.1155	0.9540	3.19±2.21
BigVSAN	**0.7881**	4.116	**0.3381**	0.0935	0.9635	3.24±1.95
BigVSAN w/ snakebeta activation	0.7992	**4.120**	0.4129	**0.0924**	**0.9644**	**3.43**±2.04

Samples

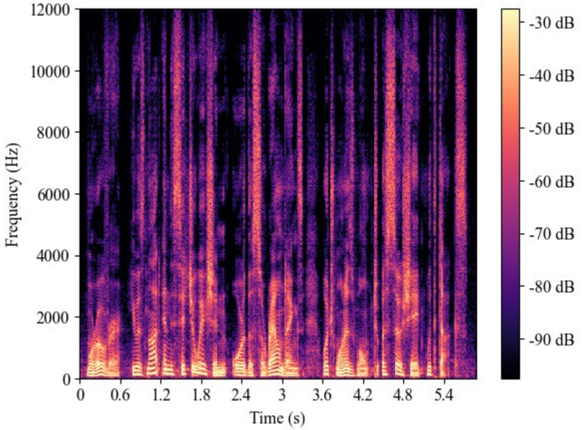
Ground truth



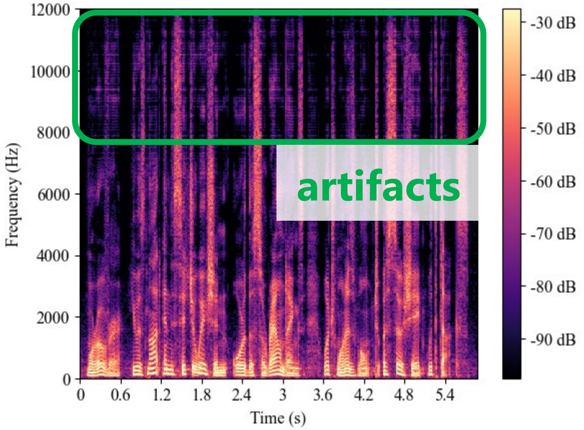
BigVGAN
(our reproduction)



BigVSAN w/ snake



BigVSAN w/ snakebeta



Experiments: moderate-sized vocoder training

■ SAN-ify MelGAN and Parallel WaveGAN

We trained MelGAN and Parallel WaveGAN vocoders with **Least-squared SAN**

We used the VocBench framework [Albadawy et al., ICASSP 2023], which provides a shared environment where we can train/evaluate different vocoders on three public dataset: LJ speech, LibriTTS, and VCTK.

Result

Table 2. Results for Fréchet Audio Distance (FAD) evaluated on three datasets: LJ Speech, LibriTTS, and VCTK. Scores marked with † are reported in the VocBench paper [38].

Dataset	MelGAN†	MelSAN	Parallel WaveGAN†	Parallel WaveSAN
LJ Speech	1.51	**1.34**	0.92	**0.84**
LibriTTS	2.95	**2.91**	1.41	**0.87**
VCTK	1.76	**1.69**	1.22	**0.76**

SAN outperforms GAN in all combinations of vocoder model and dataset!

FAD: distance between the distribution of real recorded speech and that of synthesized speech (the lower, the better)

Conclusion

Recap

- We applied SAN (the improved GAN training framework) to GAN-based vocoders
  - SAN can find the projection that can distinguish real and generated data samples
  - We designed a new loss function for satisfying SAN's requirements
- We demonstrated SAN boosts the performance of existing vocoders, including BigVGAN

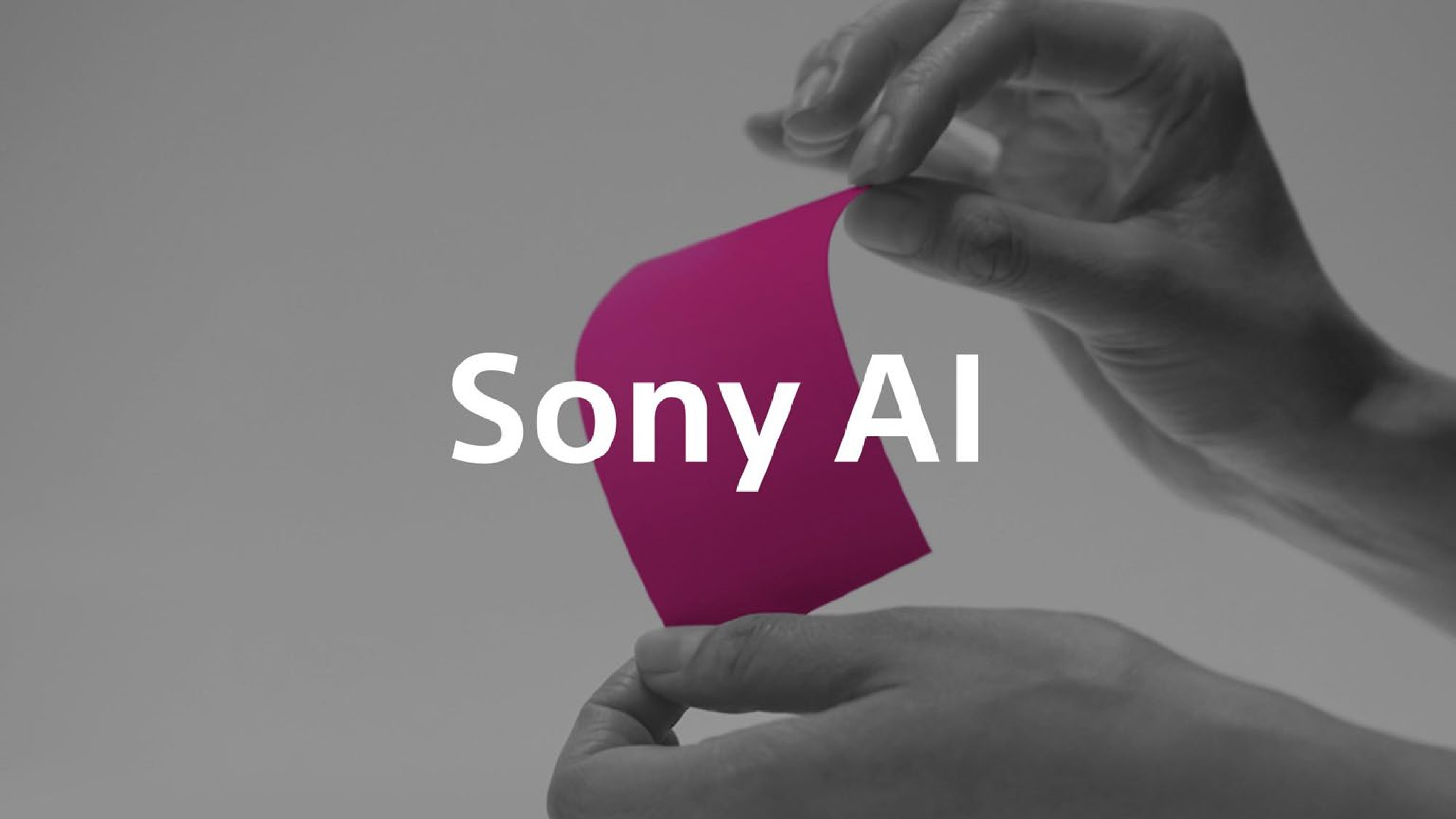
Future directions

- Incorporating the SAN training framework is orthogonal to most types of improvements of discriminator/generator architectures.

  ⇒ SAN can boost other GAN-based vocoders: EVA-GAN [Liao+, arXiv, '24], MusicHiFi [Zhu+, arXiv, '24], etc.

- GAN is used as an auxiliary loss in other tasks
  - Text-to-speech: NaturalSpeech 3 [Ju+, arXiv, '24], StyleTTS 2 [Li+, NeurIPS '23], VITS [Kim+, ICML '21], etc.
  - Audio compression: DAC [Kumar+, NeurIPS '23], EnCodec [Défossez+, TMLR, '23], SoundStream [Zeghidour+, TASLP, '21], etc.

  ⇒ Applying SAN to these types of models is an interesting direction


Sony AI