Generative Models for Speech Time Scale Modification

Jayin Khanna[1] Dr Niteesh Sahni[1] Dr Prasanta Kumar Ghosh[2]

[1]Department of Mathematics, Shiv Nadar Institution of Eminence [2]Department of Electrical Engineering, Indian Institute of Science

Generative Models

Setup. Let the dataset be $\mathcal{D} = \{x_1, x_2, \ldots, x_n\} \overset{\text{iid}}{\sim} P_X$ (unknown), where $x_i \in \mathbb{R}^d$.

Goal. Given $\mathcal{D}$: **(1)** estimate P_X, and **(2)** learn to sample from it.

General Principle.

1. Assume a parametric family P_θ (represented via a deep neural network).
2. Define a divergence metric $\mathcal{D}(P_X \| P_\theta)$.
3. Minimize over θ:

$$\theta^* = \arg\min_\theta \mathcal{D}(P_X \| P_\theta)$$

Once optimized, draw $z \sim \mathcal{N}(0, I)$ and pass through $g_\theta(z)$ to sample from P_X.

1. Generative Adversarial Networks (GANs)

Methodology. GANs choose the f-divergence corresponding to the Jensen–Shannon divergence and minimize it via a minimax game between two networks:

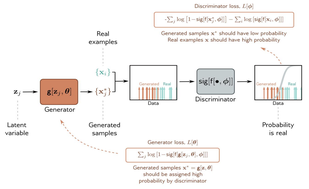
- **Generator** g_θ: maps $z \sim \mathcal{N}(0, I)$ to fake samples $\tilde{x} = g_\theta(z)$.
- **Discriminator** D_ω: outputs the probability that an input is real.

Saddle-Point Objective.

$$\min_\theta \max_\omega \mathbb{E}_{x \sim P_X}[\log D_\omega(x)] + \mathbb{E}_z[\log(1 - D_\omega(g_\theta(z)))]$$

Training Loop.

- *Discriminator step* (fix θ): gradient ascent on $\mathcal{J}_{\text{GAN}}$.
- *Generator step* (fix ω): gradient descent; loss minimised when $D_\omega(g_\theta(z)) \to 1$.



$$z \sim \mathcal{N}(0, I) \xrightarrow{g_\theta} \tilde{x} \xrightarrow{D_\omega} [0, 1]$$

2. Variational Autoencoders (VAEs)

Methodology. VAEs are latent variable models $p_\theta(x) = \int p_\theta(x, z)\, dz$. Since the marginal is intractable, maximising $\log p_\theta(x)$ is replaced by maximising the **Evidence Lower Bound (ELBO)**:

$$\mathcal{J}_\theta(\phi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big]}_{\text{reconstruction}} - \underbrace{D_{\text{KL}}\big(q_\phi(z|x) \,\|\, p_\theta(z)\big)}_{\text{regulariser}}$$

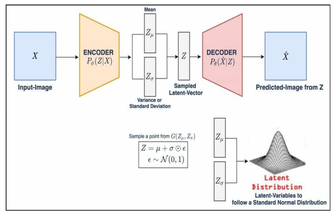
Networks.

- Encoder $q_\phi(z|x)$: maps data to a posterior over latents.
- Decoder $p_\theta(x|z)$: reconstructs data from latent samples.

Reparameterisation Trick. Sample $z = \mu_\phi(x) + \sigma_\phi(x) \odot \varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$, to allow gradients to flow through the sampling step.

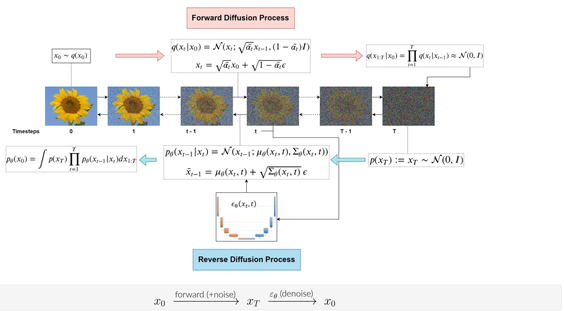
Optimisation.

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} \mathbb{E}_{x \sim \mathcal{D}}\big[\mathcal{J}_\theta(\phi)\big]$$



$$x \xrightarrow{q_\phi} (\mu, \sigma) \xrightarrow{\varepsilon} z \xrightarrow{p_\theta} \hat{x}$$

3. Denoising Diffusion Probabilistic Models (DDPMs)



$$x_0 \xrightarrow{\text{forward (+noise)}} x_T \xrightarrow{\varepsilon_\theta\ \text{(denoise)}} x_0$$

Methodology. DDPMs are hierarchical latent variable models with T latent spaces. The **forward process** is fixed: it gradually destroys data by injecting Gaussian noise over T steps:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t;\ \sqrt{\alpha_t}\, x_{t-1},\ (1-\alpha_t)I\big)$$

Marginalising gives a closed form: $q(x_t|x_0) = \mathcal{N}\big(x_t;\ \sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t)I\big)$.

Reverse Process. A neural network ε_θ learns to *denoise* at each step. The simplified training loss is:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t, x_0, \varepsilon}\big[\,\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\big]$$

Generation. Start from $x_T \sim \mathcal{N}(0, I)$ and iteratively apply:

$$x_{t-1} = \mu_\theta(x_t, t) + \sqrt{\beta_t}\, z, \quad z \sim \mathcal{N}(0, I)$$

4. Flow Matching

Methodology. Flow Matching learns a time-dependent vector field $v_\theta(x, t)$ that transports a source distribution $p_0 = \mathcal{N}(0, I)$ to the data distribution $p_1 = P_X$ along continuous ODE trajectories. Given a sample x_t interpolated between noise x_0 and data x_1:

$$x_t = (1-t)\, x_0 + t\, x_1, \quad t \in [0, 1]$$

the **conditional** target vector field pointing from noise to data is:

$$u_t(x \mid x_1) = x_1 - x_0$$

Training Objective. Minimise the **Conditional Flow Matching (CFM)** loss, regressing v_θ onto the target field:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_0, x_1}\big[\,\|v_\theta(x_t, t) - u_t(x_t \mid x_1)\|^2\big]$$

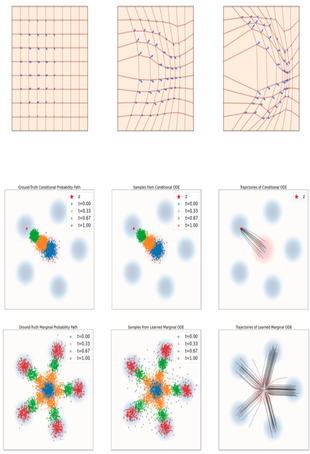
Generation. Integrate the learned ODE forward from $t = 0$ to $t = 1$ using any standard solver (e.g. Euler, RK4):

$$\frac{dx}{dt} = v_\theta(x, t), \quad x(0) \sim \mathcal{N}(0, I) \implies x(1) \approx x_1 \sim P_X$$

Unlike DDPMs, generation requires far fewer function evaluations (as few as 1–10 steps) since the learned trajectories are nearly straight lines.

Connection to DDPMs. DDPMs can be viewed as a special case where the interpolation path is curved (Gaussian noise schedule); Flow Matching straightens these trajectories, reducing discretisation error and enabling faster sampling.

$$x_0 \sim \mathcal{N}(0, I) \xrightarrow{\text{ODE}} x_1 \sim P_X \quad \frac{dx}{dt} = v_\theta$$



Audio Representations: Waveform to Mel-Spectrogram

Why Not Raw Waveforms? A waveform shows amplitude vs. time — sufficient for loudness and duration but blind to frequency content. Speech is non-stationary: vowels, consonants, and transients each carry distinct frequency signatures evolving on the order of milliseconds. Time–frequency representations contain these cues.

Step 1 — Short-Time Fourier Transform (STFT). Break the waveform into short overlapping frames (e.g. 25 ms, 10 ms hop), apply a Hann window, and compute the DFT of each frame:

$$X_m[k] = \sum_{n=0}^{L-1} x_m[n]\, e^{-j\frac{2\pi kn}{L}}, \quad k = 0, 1, \ldots, K-1$$

The spectrogram is the magnitude $|X_m[k]|$ stacked across all frames m.

Step 2 — Mel Filterbank. Human hearing is non-linear: we resolve low frequencies more finely than high ones. Frequencies are mapped to the **mel scale**:

$$m(f) = 1125 \cdot \ln\left(1 + \frac{f}{700}\right)$$

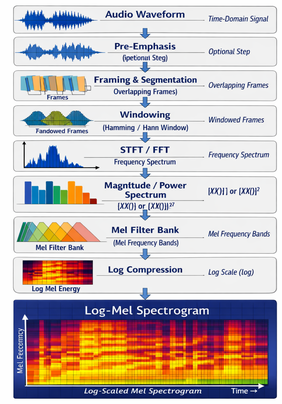
A filterbank matrix $T \in \mathbb{R}^{M \times K}$ of M triangular filters (e.g. $M = 80$) is applied to the power spectrum $P \in \mathbb{R}^{K \times N_{\text{frames}}}$:

$$E = T \cdot P \ \in \ \mathbb{R}^{M \times N_{\text{frames}}}$$

Step 3 — Log Compression. Apply $\log E$ to mimic human loudness perception and compress the dynamic range. The final mel-spectrogram has shape $(M \times N_{\text{frames}})$ and is the standard input to modern speech models (Tacotron, FastSpeech, ScalerGAN).

Pipeline Summary.

$$x[n] \xrightarrow{\text{frame + window}} x_m[n] \xrightarrow{\text{DFT}} |X_m[k]| \xrightarrow{T} E \xrightarrow{\log} \log E$$



Mel-spectrogram of a speech utterance (80 mel bins × time frames)

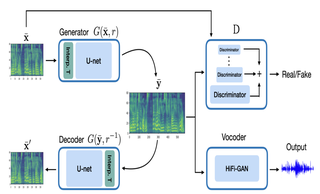
1. ScalerGAN (Cohen et al., 2022)

Methodology. Given mel-spectrogram $\bar{x}$ and desired rate $r \in \mathbb{R}$, ScalerGAN produces a time-scaled spectrogram $\bar{y}$ of length $M = \lceil Nr \rceil$ using a GAN framework trained on unpaired speech — no examples at different speeds are required.

Pipeline.

- **Generator** $G(\bar{x}, r)$: bilinear interpolation T^r stretches the time axis, then a U-Net refines the result.
- **Multi-scale Discriminator** D: 5 sub-discriminators each operating on a differently down-scaled spectrogram; encourages realism at coarse and fine resolutions simultaneously.
- **Decoder** (cycle consistency): the same generator reconstructs the original via $\hat{\bar{x}} = G(\bar{y}, r^{-1})$, preventing mode collapse.

$$\min_G \max_D \ \mathcal{L}_{\text{LS}}(G, D) \ + \ \lambda \big\| G\big(G(\bar{x}, r),\, r^{-1}\big) - \bar{x} \big\|_1$$



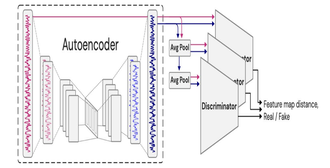
$$\bar{x} \xrightarrow{T^r\text{+U-Net}} \bar{y} \xrightarrow{G(\cdot, r^{-1})} \hat{\bar{x}} \xrightarrow{\text{HiFi-GAN}} \text{waveform}$$

2. TSM-Net (Chu et al., 2023)

Methodology. Instead of operating on waveforms or spectrograms directly, TSM-Net encodes raw audio into a high-level latent representation called the **Neuralgram**, where each vector encodes 1024 audio samples — enough to capture an entire sinusoid of the lowest audible frequency.

Pipeline.

- **Encoder** $\mathcal{A}_E$: mirrors a MelGAN generator in reverse — 5 downsampling stages ($8\times, 8\times, 4\times, 2\times, 2\times$) with dilated convolutions and residual blocks, compressing audio $1024\times$ into the Neuralgram.
- **Spatial Interpolation** $S(\cdot, r)$: cubic interpolation of the Neuralgram along the time axis by factor r
- **Decoder** $\mathcal{A}_D$: symmetric upsampling network reconstructs the time-scaled waveform from the stretched Neuralgram.
- **Multi-scale Discriminators** (D_1, D_2, D_3): trained with hinge GAN loss + feature-matching loss $\mathcal{L}_{FM}$.



$$x \xrightarrow{\mathcal{A}_E} \text{Neuralgram} \xrightarrow{S(\cdot, r)} \text{scaled Neuralgram} \xrightarrow{\mathcal{A}_D} \hat{x}$$

$$\min_{\mathcal{A}} \ \mathbb{E}_x \left[-\sum_{k=1}^{3} D_k(\mathcal{A}(x)) \right] + \lambda \sum_{k=1}^{3} \mathcal{L}_{FM}(\mathcal{A}, D_k)$$

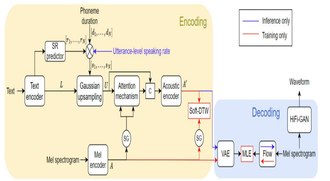
3. Neural ATSM (Lee & Jang et al., 2024)

Methodology. A fully neural ATSM that predicts *sentence-specific, phoneme-level* rates directly from text, eliminating MFA and enabling strong performance on synthetic speech.

Pipeline.

- **Text Encoder** (Transformer + LConv blocks): extracts linguistic feature L from the phoneme sequence.
- **Mel Encoder** (Transformer): extracts acoustic feature A from the input mel-spectrogram.
- **SR Predictor** (conv + linear): estimates per-phoneme rate r_i; controlled duration $s_i = d_i \cdot r_i$.
- **Gaussian Upsampling + Attention:** upsamples L to mel length; dot-product attention over A yields context-aware feature I, concatenated and fed to the **Acoustic Encoder** to produce A', aligned to A via Soft-DTW loss.
- **Decoding:** VAE estimates $\mathcal{N}(\mu, \sigma)$; Flow (Glow-TTS) generates mel from sampled noise; HiFi-GAN produces the waveform.

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \mathcal{L}_{\text{soft-DTW}} + \mathcal{L}_{\text{sr}}$$



$$\text{text} \xrightarrow{\text{SR Pred.}} s_i \xrightarrow{\text{GaussUp+Attn}} A' \xrightarrow{\text{VAE+Flow}} \text{mel} \xrightarrow{\text{HiFi-GAN}} \text{wave}$$

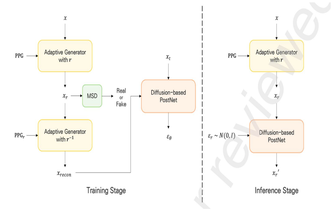
4. DiffATSM (Jang et al., 2024)

Methodology. Operates directly on raw waveforms *without* text transcriptions. Phonetic posteriorgrams (PPG) from HuBERT preserve phonetic content; a diffusion-based PostNet recovers fine spectral detail lost during scaling.

Pipeline.

- **PPG Extraction:** HuBERT extracts PPGs encoding pronunciation without requiring any text.
- **Adaptive Transformation:** voiced sections scaled at $r_v > r$, unvoiced at $r_{uv} < r$ — preserving intelligibility at high speeds by protecting the voiced content.
- **Conditional U-Net Generator:** adaptively transformed mel + PPG → scaled spectrogram x_r; trained with LS-GAN + L1 cycle reconstruction loss via MSD.
- **Diffusion PostNet** (WaveNet backbone): denoises over T reverse steps conditioned on x_{recon}:

$$\mathcal{L} = \mathbb{E}\Big[\Big\|\epsilon - \epsilon_\theta\big(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t,\ x_{\text{recon}}\big)\Big\|^2\Big]$$



$$\text{mel} + \text{PPG} \xrightarrow{\text{Adapt.+U-Net}} x_r \xrightarrow{\text{DiffPostNet}} x'_r \xrightarrow{\text{HiFi-GAN}} \text{wave}$$

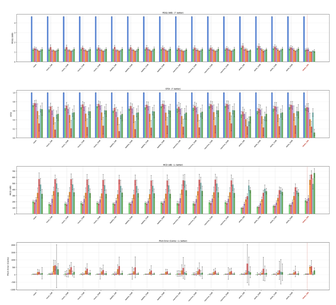
Experiments

Models Evaluated. We benchmark 14 TSM systems spanning classical signal-processing baselines and modern deep-learning approaches:

Category	Models
Time-domain	OLA, SOLA, PICOLA, TD-PSOLA, WSOLA
Spectral-domain	Phase Vocoder, Phase-Locked PV, STRAIGHT, WORLD
Deep learning	ScalerGAN, TSM-Net, ATSM, Neural ATSM, DiffATSM

Acoustic Conditions. Three experimental axes:

- **Sampling Rate:** utterances evaluated at 22.05 kHz (clean) and 16 kHz (clean-16k)
- **Additive Noise:** each utterance mixed with *White*, *Babble*, and *Machine & Music* noise at SNR levels of 5, 10, 15, and 20 dB
- **Music Conditions (5–20 dB):** Music added to speech at 4 SNR levels



Github: https://github.com/jk-2604

jk873@snu.edu.in