Reconstructing the Vocal Tract with Differentiable Acoustic Simulation

Eric M. Chen
MIT CSAIL
echen01@mit.edu

Jin Woo Lee
MIT RLE & KAIST GSCT
jnlee@mit.edu

Vincent Sitzmann
MIT CSAIL
sitzmann@mit.edu

Abstract

The vocal tract is the region of the human body responsible for filtering one's voice to create speech. In this paper, we present a differentiable and GPU accelerated acoustic simulator for the vocal tract. The differentiable simulator synthesizes speech by propagating sound along an acoustic tube model of the vocal tract, and via its gradients, can solve the inverse problem: reconstructing the shape of the vocal tract solely from the sound it produces. Although the inverse mapping between geometry and sound is notoriously non-convex, we discover that gradient descent succeeds with three technical contributions: (1) we design a frequency domain formulation of the vocal tract's fluid dynamics that is 70x more GPU parallelizable than finite differences in time, (2) we integrate a differentiable model for turbulence to synthesize consonants, and (3) similar to prior work in implicit neural representations (INRs) and neural fields, we find that parameterizing the geometry with a neural network accelerates convergence and escapes local minima that trap discrete representations. Because the simulator is differentiable, it is readily integrated with other deep learning pipelines to enable novel linguistics and medical imaging applications. (1) We demonstrate self-supervised autoencoding of vocal tract shapes across 11 languages, and (2) we couple our simulator with a generative model of MRI (magnetic resonance imaging) images to reconstruct one's moving vocal tract from only their speech without paired data.

1 Introduction

Learning to speak is fundamentally an inverse problem. An infant babbles to discover the non-linear mapping between their articulatory motor commands and their speech, learning the controls required to produce language. Most remarkably, children solve this inverse problem without ever seeing paired data between the muscle configuration of their vocal tract and the speech it produces.

In domains like robotics, this type of sensorimotor learning is accelerated by differentiable simulators. Differentiable simulators, such as Brax [22], Taichi [25], and Warp [38], are computer graphics engines which allow gradients from a physical environment to backpropagate directly to a policy, enabling efficient learning of locomotion and manipulation. But to picture the physical actions of your mouth, tongue, etc. that form language, no equivalent computational infrastructure exists.

Such a method to visualize one's vocal tract would have significant impact in language, such as studying the cognitive processes of language acquisition [2, 8], in music, as a tool for voice coaching or singing synthesis, as well as in healthcare such as speech pathology and speech brain-computer interfaces [41]. Although several differentiable acoustic simulators [60, 20, 34, 35] have been recently proposed for room acoustics, these rely on ray-based methods for stationary systems that are ill-suited for capturing the time-varying morphological transformations of the vocal tract [33, 23]. On the other hand, while neural vocoders [45, 30, 61] achieve high acoustic realism, they operate as black boxes that ignore the underlying physics. Conversely, classical vocal tract simulators [27, 39] are

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

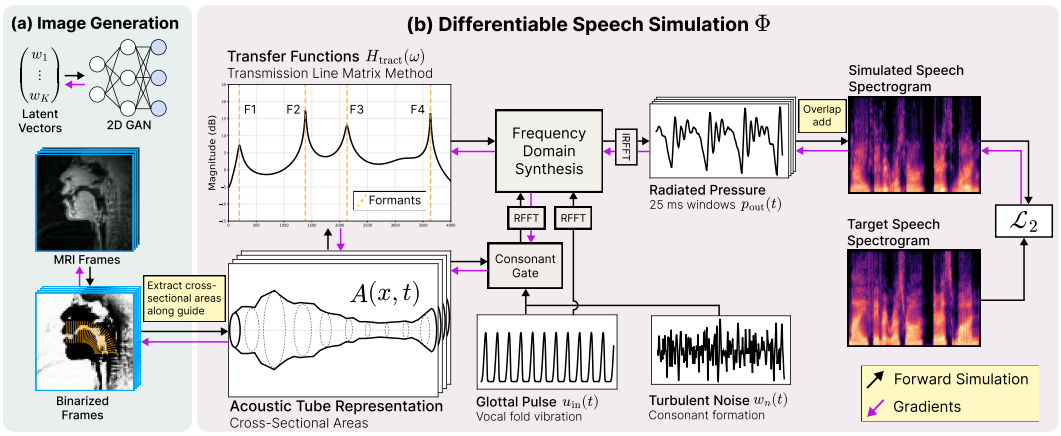


Figure 1: **Method Overview.** We introduce a method to reconstruct a MRI video of a person's moving vocal tract from their speech alone. **(a)** Given an MRI image of a person's vocal tract, we extract cross-sectional areas along the length of the tract. **(b)** These areas define an *acoustic tube*. Our differentiable simulator Φ propagates sound through the acoustic tube to synthesize speech. Gradients from the loss flow back through Φ and the generative model into a set of latent vectors, reconstructing the MRI video from the target speech signal.

computationally expensive and CPU-bound, rendering them incompatible with modern deep learning pipelines and making them ineffective at solving the speech-to-motor inverse problem.

We address this gap by introducing a differentiable, GPU-accelerated physical simulator of speech production. Our simulator produces speech by propagating sound along an acoustic tube representation of a person's vocal tract, and via its gradients, can visualize the shape of the vocal tract from only audio. We adopt a frequency-domain solution tailored for GPUs. This approach is $70\times$ more parallelizable on GPUs and, crucially, avoids high-frequency noise inherent to sequential time-stepping (Section 5.1). And to support end-to-end learning, we introduce a gradient-friendly formulation of turbulent noise for modeling consonants (Section 5.2). Although acoustic inverse problems are non-convex, leading to sub-optimal solutions [57, 47], we find that, much like in neural rendering, representing the geometry with a neural network acts as a regularizer that stabilizes the learning dynamics (Section 6.1). We validate these design choices empirically.

We demonstrate the effectiveness of our simulator through two novel applications. First, we use our simulator to train a *self-supervised* autoencoder which maps raw audio to the vocal tract shape (Section 6.2). Second, we provide a tool to visualize an MRI (magnetic resonance imaging) video of a person's moving vocal tract from their speech, without training on any paired speech-MRI data. This is a task, to the best of our knowledge, that has not been previously attempted. As outlined in Figure 1, we connect a generative model of MRI images to our differentiable acoustic simulator. Our method then jointly recovers an MRI video and audio. Because our model is geometrically grounded, it outperforms the closest prior work trained on paired data [43].

All in all, by establishing a first-of-its-kind differentiable link between vocal tract geometry and speech, we provide the infrastructure to treat speech as a physically-grounded learning problem.

2 Related Work

Articulatory Synthesis. Many physics-based simulators for the vocal tract have been introduced over the last century [27, 39, 55, 4], the most well-known being the Kelly-Lochbaum method [27], and the state of the art being VocalTractLab (VTL) [4]. However, these models are difficult to directly backpropagate through because they rely on finite differences to model a differential equation of sound propagation. Furthermore, they use separate forward models for consonants and vowels, causing a discontinuity in their control flow.

Black Box Methods. To avoid backpropagating through an acoustic simulation, recent works have proposed to solve the inverse problem by training neural networks on paired data between speech and articulatory measurements. However, because these networks are black boxes, they are difficult to

interpret physically. Furthermore, their quality fundamentally depends on paired training data, which are sparse. TensorTract2 [31] trains a neural network to map articulatory controls to synthetic sounds generated from VocalTractLab [4]. Black box models have also been trained on biomedical data such as vocal tract MRI [43] and electromagnetic articulography (EMA) [9]. In contrast, our method does not require paired articulatory and speech data.

Another related line of work is differentiable digital signal processing (DDSP) [15]. DDSP models been used to model human speech, such as in neural source filter [61] and Schulze-Forster et al. [51]. However, unlike our method, these models are not constrained by physics. Schulze-Forster et al. [51], for instance, model the vocal tract filter with an all-pole filter via line spectral frequencies. On the other hand, our physical simulator can model both poles and zeros, allowing for realistic modeling of radiation and viscous losses.

Differentiable Acoustic Simulation with Neural Fields. Following the success of differentiable rendering and neural fields for solving inverse problems in 3D computer vision [42, 54], several works have also proposed differentiable simulators and neural fields for sound propagation [37, 60, 34, 35]. These works focus primarily on room-scale environments, where sound waves are approximated as rays for efficiency. However, such ray-based approximations are inappropriate for the human voice. This is because the fundamental pitch of the human voice (80–255 Hz) has much longer wavelength than the diameter of the vocal tract (< 10 cm), so the wave effects become dominant. Instead, our simulator explicitly leverages physics-based modeling suitable for time-varying acoustic tubes such as the human vocal tract.

Südholt et al. [57] have also recently proposed to reconstruct vocal tract areas by backpropagating through a differentiable simulator. However, instead of parameterizing the vocal tract's area with an implicit neural representation, they adopt an explicit parameterization. Furthermore, they only demonstrate results on the limited setting of static vowels and consonants. Meanwhile, our work shows that neural fields are an effective way to reconstruct both the geometry and movement of the vocal tract, enabling the synthesis of full words.

3 Background: The Acoustics of Speech

Speech production is commonly framed as a source-filter process. First, a source signal is produced by either vibrations of the vocal folds or turbulent noise formed at narrow constrictions. Then, the vocal tract's geometry acts as an acoustic resonator, filtering the audio source to shape the different phonemes we interpret as speech.

Acoustic Tubes. Although the space of all possible speech sounds is large, the vocal tract's geometry moves in relatively fewer dimensions over time. Following many prior works [3, 27, 39, 4], we represent the vocal tract as an *acoustic tube*: a one-dimensional tube whose cross-sectional area $A(x)$ varies along its length. Pictured in Figure 1, the size of this tube is dynamically controlled by articulators such as the tongue, mouth and jaw. Sound propagates as waves along this tube, whose physics are governed by the linearized Euler equations:

$$\frac{A(x)}{\rho c^2}\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}u(x,t) \qquad \frac{\rho}{A(x)}\frac{\partial}{\partial t}u(x,t) = -\frac{\partial}{\partial x}p(x,t) \tag{1}$$

$p(x,t)$ is sound (pressure), $u(x,t)$ is the volume velocity of airflow, ρ is the density of air, and c is the speed of sound. x ranges over $[0, L]$, where L is the length of the vocal tract, and t ranges over $[0, T]$, where T is the length of the speech signal. Most importantly, $p(L,t)$, which we name $p_{\text{out}}(t)$, is what we perceive as speech. For brevity, material terms are omitted, but the equations can be augmented with viscous damping and wall vibration terms, summarized in Appendix A. To synthesize speech, boundary conditions are set at each end of the tract as Eq. (22)-(23). At $x = 0$, the vocal folds vibrate, leading to a glottal pulse: $u(0,t) \triangleq u_{\text{in}}(t)$. We represent $u_{\text{in}}(t)$ with the Liljencrants–Fant (LF) model [18]. At $x = L$, a radiation boundary is set for the lips.

Acoustic Tubes as Source-Filter Models. In this problem, the primary object of interest is determining the lip radiation pressure $p_{\text{out}}(t)$ given a source $u_{\text{in}}(t)$. Considering a sufficiently small time window compared to the changes in oral cavity, $A(x)$ can be regarded as quasi-static, and the relationship between p_{out} and u_{in} is time-invariant. Solving the linearized Euler equations for a given $A(x)$ is thus analogous to deriving a linear time-invariant (LTI) filter h_{tract}, such that $p_{\text{out}} = h_{\text{tract}} * u_{\text{in}}$, or equivalently $P_{\text{out}} = H_{\text{tract}} U_{\text{in}}$ in the frequency domain. The peaks of H_{tract}'s frequency response

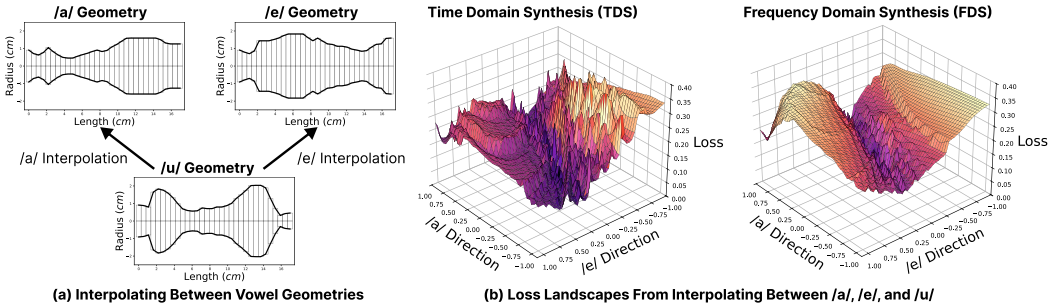


Figure 2: **FDS Leads to Smoother Loss Landscapes. (a)** Starting from the vowel /u/, we interpolate its area in the direction of the /a/ geometry and /e/ geometry. **(b)** We plot the log Mel spectrogram loss between the sound produced by the interpolated geometries and the sound produced by /u/. The loss from time domain synthesis is noiser than the loss from frequency domain synthesis.

function are called its formants, which are crucial auditory cues for humans to perceive vowels (illustrated in Figure 1). Although representing the system in either the time domain or frequency domain may seem equivalent, from an optimization perspective, the choice of domain significantly impacts the convergence of gradient-based approaches as we shall see in Section 5.

Discretization. In accordance with the conventions of prior studies [27, 12], the area function $A(x,t)$ of the dynamic oral cavity is simplified into a quasi-static and piecewise-constant geometry. Specifically, the area function $A(x,t)$ of a vocal tract is discretized into uniform $N \times K$ lattice with spatial sections of length L/N and temporal windows of length T/K, where each windowed section is considered to have constant area, *i.e.*, $A[n,k] = A\left(\frac{nL}{N}, \frac{kT}{K}\right) \approx \text{const}$. We call the simulator which solves the Euler equations in the time-varying case $\Phi : (u_{\text{in}}, A) \mapsto p_{\text{out}}$.

4 Problem Statement and Method Overview

We outline our method in Figure 1. Given a target speech sample p_{tgt}, our objective is to infer the geometry of the vocal tract that produced p_{tgt}. We first estimate the vocal fold vibration $u_{\text{in}}(t)$ with a pitch tracker (Appendix C), and set it as the input boundary condition. Then, our strategy is to leverage gradients from a differentiable simulator Φ to find an area function A_θ that minimizes the squared distance, $\mathcal{L}_2$, between spectrograms of the output $p_{\text{out}} \triangleq \Phi(u_{\text{in}}, A_\theta)$ and the target p_{tgt}. Enabling both efficient forward simulation and gradient-based reconstruction requires addressing a series of challenges, each of which motivates a technical contribution of our work.

How do we make Φ and its gradients efficient to compute? Section 5.1 demonstrates how solving the governing equations in the frequency domain enables GPU parallelization which is $70\times$ faster than time domain synthesis, and more importantly, enables stable gradient flow.

How do we model non-linear effects? Turbulence, which forms consonants, is a non-linear effect not captured by the linearized Euler equations. Section 5.2 introduces a differentiable turbulence model that unifies vowel and consonant synthesis in a single forward pass.

How do we avoid sub-optimal solutions for A_θ? The speech-to-geometry inverse problem is non-convex, and a differentiable forward model alone does not guarantee a good solution. Drawing on recent work in neural fields, we find that parameterizing A_θ with a neural network accelerates convergence. Furthermore, coupling our simulator with neural networks unlocks new applications in linguistics and medical imaging, described in Chapter 6.

5 Differentiable Speech Simulation

5.1 Frequency Domain Synthesis (FDS)

Various numerical methods such as finite differences can be employed to obtain solutions of the linearized Euler equations. Even though backpropagating the gradients through the finite differences is possible, we find that the feasibility of this gradient signal is limited by challenges such as noise

Table 1: **Speech Reconstruction Metrics.** When reconstructing real speech examples from LibriTTS-R [29], the audio from frequency domain synthesis (FDS) is significantly more perceptible than the results from time domain synthesis (TDS). To balance fidelity and efficiency, we use a window of 25 ms and a hop ratio of $1/4$. FDS is also 71.5x faster than real time (RT) while TDS is only 1.08x faster.

	TDS	FDS (25 ms window)		FDS (50 ms window)	
	—	1/8 Hop	1/4 Hop	1/8 Hop	1/4 Hop
SI-SDR	7.32 ±2.42	17.83 ±2.19	16.39 ±2.42	17.47 ±1.94	15.82 ±2.47
STOI	0.77 ±0.03	0.93 ±0.02	0.93 ±0.02	0.93±0.02	0.93 ±0.02
PESQ	1.55 ±0.14	2.02 ±0.31	1.98 ±0.26	1.97 ±0.26	1.98 ±0.25
RT Factor	1.08x	35.0x	71.5x	36.0x	70.0x

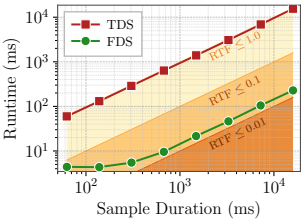


Figure 3: **Runtime.** FDS is 70x faster than TDS on GPU. Tested on a RTX 4090 with a 25 ms window and $1/4$ hop.

accumulation from extensive temporal recursion, leading to noisy loss landscapes that cause neural network training to fail in many cases. Assuming the linear time invariance as described in Section 3, the system admits separable eigensolutions $p(x,t) = P(x,\omega)e^{-j\omega t}$ and $u(x,t) = U(x,\omega)e^{-j\omega t}$, and solving the linearized Euler's equations essentially reduces to the Helmholtz eigenvalue problem:

$$\frac{j\omega A(x)}{\rho c^2} P(x,\omega) = \frac{\partial}{\partial x} U(x,\omega) \qquad \frac{j\omega \rho}{A(x)} U(x,\omega) = \frac{\partial}{\partial x} P(x,\omega) \tag{2}$$

The boundary conditions simplify to $U(0,\omega) = U_{\text{in}}(\omega)$, and $P(L,\omega) = Z_{\text{lips}}(\omega)U(L,\omega)$, where $Z_{\text{lips}}(\omega)$ is the impedance defined in Eq. (36).The time-independence achieved by transforming the governing equation into a spatial ODE offers distinct advantages by not only eliminating temporal recursion but also enabling independent analysis of the geometries for each frequency ω. When using central differences for the spatial derivatives, as $\frac{\partial}{\partial x} U(x,\omega) \mid_{n+\frac{1}{2}} = \frac{U_{n+1} - U_n}{\Delta x}$ and $\frac{\partial}{\partial x} P(x,\omega) \mid_{n+\frac{1}{2}} = \frac{P_{n+1} - P_n}{\Delta x}$, the relationship between neighboring sections n and $n+1$ can be expressed as a 2×2 chain matrix, $\mathbf{K}_n$. Multiplying successive $\mathbf{K}_n$ matrices over N sections derives a closed-form solution for the vocal tract filter, $H_{\text{tract}}(\omega)$:

$$\begin{bmatrix} P_{n+1} \\ U_{n+1} \end{bmatrix} = \mathbf{K}_n \begin{bmatrix} P_n \\ U_n \end{bmatrix}, \qquad \mathbf{K}_{\text{tot}}(\omega) = \prod_{n=0}^{N} \mathbf{K}_n(\omega), \qquad \begin{bmatrix} P_{N+1} \\ U_{N+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}_{\mathbf{K}_{\text{tot}}(\omega)} \begin{bmatrix} P_0 \\ U_0 \end{bmatrix} \tag{3}$$

$$H_{\text{tract}}(\omega) = \frac{Z_{\text{lips}}(\omega)}{\mathbf{A}(\omega) - \mathbf{C}(\omega) Z_{\text{lips}}(\omega)} = \frac{P_{\text{out}}(\omega)}{U_{\text{in}}(\omega)} \tag{4}$$

The output pressure spectrum is then $P_{\text{out}} = H_{\text{tract}} U_{\text{in}}$, and the synthesized speech p_{out} can be recovered with an inverse Fourier transform. This technique is an instance of the *transmission line matrix* (TLM) method [5], because its equations describe an equivalent acoustic circuit consisting of $\mathbf{K_n}$ components. We detail its derivation in Appendix A.3. Two properties of this formulation are critical. First, the closed-form solution for H_{tract} is independent of temporal discretization. Second, the $\mathbf{K}_n$ matrices can be solved simultaneously for each frequency, enabling parallelization.

Modeling Time-Varying Vocal Tracts. To synthesize speech from time-varying vocal tracts, we compute p_{out} independently for successive windows then combine them using the overlap-add method. We apply a Tukey analysis window with $\alpha = 0.25$ to u_{in}, and a Hann synthesis window to p_{out}.

5.2 Differentiable Gating of Consonants

Consonants as Turbulence. Although the linearized Euler equations can be solved efficiently in the frequency domain, they do not describe consonant production, which is non-linear. Consonants are formed when steady air is pushed through narrow constrictions in the vocal tract, breaking down into turbulence. This transition from steady to turbulent flow is modeled by the squared Reynolds number. At section A_n it is defined as

$$\text{Re}_n^2 = \frac{4\rho^2}{\pi \mu^2} \frac{u_n(t)^2}{A_n}, \tag{5}$$

where μ is the dynamic viscosity of air, and u_n is the volume velocity at section n. u_n is calculated in the frequency domain as $U_n = H_{0\mapsto n} U_{\text{in}}$ [1]. Notice that as A_n becomes smaller, such as the case when forming consonants by constricting one's teeth, tongue, or lips, the Reynolds number increases.

Discontinuities in Turbulence Models. Since directly solving the compressible Navier–Stokes equations to capture turbulence is computationally prohibitive, a widely adopted approach is to use auxiliary models for the turbulence. However these auxiliary models have discontinuities that prevent gradient-based optimization. Prior works [6, 55, 39] adopt consonant formation by injecting a noise source into the vocal tract at the smallest point of constriction: $A_{\text{min}} = \min\{A_1, \ldots, A_N\}$, with amplitude $u_{\text{noise}} = \max\{0, \alpha \cdot z_n(\text{Re}_n^2 - \text{Re}_{\text{crit}}^2)\}$. α is a gain, z_n is noise, and Re_{crit} is the critical Reynolds number. Re_{crit} is a constant that gates the effect of turbulence.

Removing the Discontinuities. We replace the $\min$ and $\max$ operations with smooth approximations. First, to differentiate through the position of the constriction, we simply replace the argmin operator with $\text{softmax}(-A_n)$. By doing so, gradients flow backward into every area element, and turbulence is weighted by how small the area is at A_n. Second, we replace the $\max$ gate with a softplus function, so the overall noise injected at section n is

$$u_{\text{noise}_n} = \alpha \cdot z_n \cdot \text{softmax}(-A_n) \cdot \text{softplus}(\text{Re}_n^2 - \text{Re}_{\text{crit}}^2) \tag{6}$$

We use fractal Perlin noise [49] as the noise source, which has been widely adopted to add turbulent textures in various fields such as computer graphics [7, 56] and sound synthesis [24]. Re_{crit} is set to 3500 as experimentally determined by Sondhi and Schroeter [55].

Unified Vowel and Consonant Synthesis. We represent the final form of our vocal tract transfer function as follows.

$$P_{\text{out}}(\omega) = H_{\text{tract}}(\omega) U_{\text{in}}(\omega) + \sum_{n=1}^{N} H_{n\mapsto N+1}(\omega) U_{\text{noise}_n}(\omega) \tag{7}$$

This harmonic-plus-noise spectral modeling can also be viewed as encapsulating our vocal tract transfer function within a differentiable DSP pipeline [15], thereby not only enabling dynamic transition between vowels and consonants, but also preserving differentiability. Although turbulence is computed in the time domain, this technique is still 70x faster than real time (Figure 3).

5.3 Comparing Frequency Domain Synthesis (FDS) to Time Domain Synthesis (TDS)

FDS Smooths Gradients. To validate our choice of formulation, we compare frequency domain synthesis (FDS) against time domain synthesis (TDS) based on the methods of [5]. The TDS method is a semi-implicit scheme described in Appendix B. While the two methods may be theoretically equivalent in continuous time, we find that TDS has significantly noiser gradients in implementation. To illustrate this, in Figure 2, we visualize the loss landscapes of $\mathcal{L}_2$ when interpolating between three vowels. FDS produces a consistently smooth loss landscape, while TDS suffers from noise.

FDS Improves Acoustic Reconstruction. The differences in loss landscape has direct consequences for optimization. Table 1 reports perceptual speech quality metrics for area functions fit to 25 speech samples from LibriTTS-R [29]. The experimental setup is described in Appendix C. Using TorchAudio-Squim [32], we measure SI-SDR (scale-invariant signal-to-distortion ratio), STOI (short-time objective intelligibility), and PESQ (wideband perceptual evaluation of speech quality). Across all metrics, FDS substantially outperforms TDS. The SI-SDR for TDS is almost 10 dB lower than FDS across all configurations. Notably, FDS is also robust to window and hop length. To balance fidelity and efficiency, we use 25 ms windows and a $1/4$ hop, resulting in a hop length of 6.25 ms.

FDS Enables GPU Parallelization. Because solutions for H_{tract} are independent for each frequency ω, FDS can be efficiently parallelized on a GPU. We compare the forward runtimes for FDS and TDS on an RTX 4090 with a sampling rate of 16 kHz and $N = 32$ sections in Figure 3. Given a glottal pulse u_{in}, TDS takes 1.08 seconds to synthesize a 1-second utterance. However, in the frequency domain, this takes only 14.2 ms. The runtime tends to increase linearly as the number of windows in the time-domain lattice grows. On average, FDS is 70x faster with a 25 ms window and $1/4$ hop.

[1] The notation $H_{i\mapsto j}$ denotes the transfer function from section i to j. Formulas are provided in Appendix A.3.

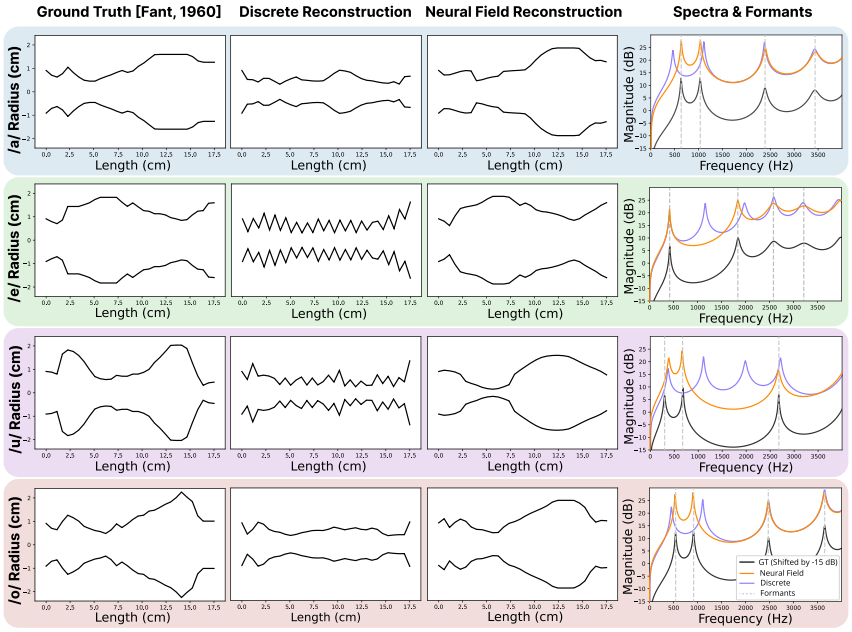


Figure 4: **Neural Fields Smoothly Reconstruct Acoustic Tubes.** Directly optimizing a discrete acoustic tube fails to fit basic vowels. Optimization is trapped around the initialization, and the recovered geometries sometimes suffer from high-frequency oscillation. In comparison, neural fields couple the geometry of each section with shared weights leading to smooth reconstructions.

6 Experiments

We now describe how our differentiable simulator can be used for solving the speech-to-geometry inverse problem. Because the mapping $\Phi(u_{\text{in}}, A_\theta) \mapsto p_{\text{out}}$ is non-convex, classical reconstruction methods require techniques like second-order optimizers, regularizers, and discrete codebooks to find good solutions [47]. However, inspired by recent work in neural rendering [42, 54], we discover that we can simply use first-order gradient descent without regularizers by parameterizing A_θ with a *neural network*. Learning from this, we propose three neural network parameterizations for the vocal tract:

1. ***A neural field***, which demonstrates how parameterizing the area function with a neural network can escape the local minima which trap discrete optimization.
2. ***A self-supervised autoencoder***, that encodes an unlabeled speech sample into an area function, then decodes it back to speech with our simulator.
3. ***A GAN***, used to synthesize realistic MRI videos of a person while speaking. To our knowledge, this is the first method to infer a MRI video from speech audio without paired speech-MRI data.

Each enables novel applications in language and medical imaging. For results of over 30 reconstructed utterances, area functions, MRIs, and singing samples, please find our videos on our website.

6.1 Neural Networks Escape Sub-optimality

Neural Fields Break Gradient Deadlock. As common in classical iterative optimization, we initially attempted to directly optimize a discrete array representing the area function $A_\theta[n, k]$ across N spatial segments and K time windows. However, this approach proved unsuccessful. Upon further investigation, we find that parameterizing $A_\theta(x, t)$ with a neural network regularizes the optimization space and better fits high-frequency detail, thereby enabling efficient optimization. Consider, for instance, the example shown in Figure 4. In this example, we initialize the model with a uniform tube and employ the Adam optimizer [28] to fit A_θ to four distinct vowel shapes. The ground truth is from the X-ray data of Fant [16]. When using discrete reconstruction, the optimization stalls as the solutions are often stuck in oscillations around the initial state. It is almost as if the individual

A_n sections of the tube are competing against one another with conflicting gradient directions. Conversely, with a continuous neural field, $A_\theta = F_\theta(x, t)$, the model is able to recover the general morphology of the area function. Even when the recovered shapes are not flawless, the formants align accurately with the ground truth. We attribute this outcome to the fact that neural fields, and neural networks more generally, share weights between the spatial sections, allowing optimization to succeed in a coarse-to-fine manner. We test two neural fields for F_θ: a fully-connected network with random Fourier feature (RFF) encoding [59], and a multiplicative filter network (MFN) [19]. The results in Table 2 show that both neural fields significantly outperform the discrete baseline.

Table 2: **Area Reconstruction.** We report the mean squared error of each area parameterization in cm^4. Both neural field methods, RFF and MFN, significantly outperform the discrete baseline.

	/ɑ/	/e/	/i/	/ɨ/	/o/	/u/
Discrete	14.66 ±2.25	26.22 ±1.36	27.66 ±2.04	29.65 ±2.34	27.42 ±3.60	28.19 ±2.54
RFF	**1.96** ±4.17	**1.29** ±0.90	**2.46** ±4.01	**0.47** ±0.32	**1.63** ±0.40	27.26 ±22.30
MFN	5.06 ±3.64	15.00 ±7.06	5.96 ±1.37	5.79 ±10.52	8.83 ±3.22	**19.98** ±5.04

6.2 Self-supervised Autoencoding of Speech

Since the process of fitting a new neural field to each utterance is computationally expensive, we also train an encoder $\mathcal{E}_\theta : p_{\text{tgt}} \mapsto A$ that maps a speech recording directly to an area function. The area function is then decoded back to speech with our simulator, Φ. We use the Wav2Vec 2.0 architecture [1] for $\mathcal{E}_\theta$. The encoder is trained end-to-end with the loss $\mathcal{L}_2(p_{\text{tgt}}, \Phi(u_{\text{in}}, \mathcal{E}_\theta(p_{\text{tgt}}))$. No paired data between p_{tgt} and $A(x, t)$ is required since Φ itself physically constrains the latent space of the autoencoder to be plausible. Note that unlike prior articulatory autoencoders [31, 9] that require paired speech and articulatory data, our model is self-supervised.

Analysis-by-Synthesis Enables Scalable Training. While there is no prior self-supervised articulatory autoencoder to compare to, we can still evaluate our work against TensorTract2 (TT2) [31], an articulatory autoencoder trained on paired data. TT2 is trained on synthetic consonant-vowel clusters generated by VocalTractLab [4]. In comparison, our autoencoder benefits from training on real-world speech datasets because it is self-supervised. Described in Appendix C.5, we train our autoencoder on 11 different languages. 200 synthesized utterances from the test set of each language are then input to Whisper [50] to evaluate their word error rates (WER) and character error rates (CER). We also report cosine similarity between ECAPA-TDNN [13] speaker identity embeddings. Across 11 languages, Table 3 demonstrates that our simulator outputs utterances that are more intelligible and better preserve identity.

Vocal Tract Area Functions Encode a Linguistic Spectrum. Consisting of only 32 spatial sections, area functions are an exceptionally compact representation of speech. To investigate what linguistic information they retain—if any—we plot a UMAP [40] projection of the area functions and color each point by its phoneme (Figure 5). Despite receiving no phonetic labels during training, the area functions remarkably form a continuous spectrum of phonemes, ordered by place of articulation. On the left are vowels like /a/, /i/, and /u/, while constrictions like /s/ and /k/ occupy the right. Although phonemes are often thought of as discrete symbols, our encoder naturally embeds them in a low-dimensional continuous space. This is a fully anticipated phenomenon, considering that the transitions in oral structure between phonemes occur smoothly in spoken language.

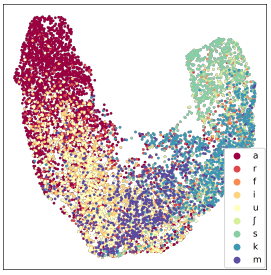


Figure 5: Encoded areas form a spectrum.

6.3 Physically-Grounded MRI Visualization from Speech

A visual representation of how one's lip position, tongue geometry, etc. affect their speech would be useful in applications such as language learning and singing instruction. However, directly capturing this data with an MRI machine is impractical outside of specialized settings. Our differentiable simulator thus enables a task that, to our knowledge, has not previously been attempted: visualizing an MRI of a person's vocal tract from their speech, without any paired speech-MRI training data.

Our method is shown in Figure 1. First, we pre-train a StyleGAN2 network [26] on MRI scans from [36]. The trained GAN G maps a latent vector $w \in \mathbb{R}^{512}$ to an MRI I. For reconstruction, we

Table 3: **Speech Autoencoding Metrics.** We report WER, CER and ID metrics, with their 95% confidence intervals, across 11 languages. Our self-supervised autoencoder better encodes intelligible utterances than TensorTract2, an autoencoder trained on synthetic consonant-vowel clusters.

	Ground Truth		**TensorTract2** [31]			**Ours**		
	WER	CER	WER	CER	Speaker ID	WER	CER	Speaker ID
English	3.91 ±0.84	2.05 ±0.70	13.00 ±1.82	7.37 ±1.11	16.86 ±1.01	**5.30** ±0.94	**2.82** ±0.75	**52.0** ±1.39
German	5.53 ±0.78	2.40 ±0.47	17.59 ±1.15	10.85 ±0.73	14.07 ±1.42	**12.49** ±1.54	**7.89** ±0.74	**54.9** ±1.45
Dutch	9.24 ±0.87	2.59 ±0.30	27.08 ±1.63	12.89 ±0.91	26.97 ±1.15	**17.19** ±1.27	**7.77** ±0.65	**59.3** ±1.17
French	4.72 ±0.79	2.27 ±0.59	19.65 ±1.71	11.37 ±1.17	15.11 ±1.34	**12.91** ±1.38	**7.69** ±0.85	**47.4** ±1.46
Spanish	3.35 ±0.69	1.37 ±0.44	9.71 ±0.93	5.13 ±0.44	14.74 ±1.49	**9.27** ±1.25	**4.43** ±0.59	**52.9** ±1.81
Italian	9.96 ±1.18	2.28 ±0.32	27.61 ±1.96	10.10 ±0.95	10.34 ±1.21	**22.01** ±1.94	**6.98** ±0.70	**53.9** ±1.44
Portuguese	6.23 ±0.79	2.43 ±0.44	26.61 ±2.50	14.15 ±2.40	17.32 ±0.99	**23.30** ±9.33	**12.68** ±5.90	**49.1** ±1.24
Polish	4.14 ±0.63	0.98 ±0.25	20.95 ±1.30	7.44 ±0.50	18.67 ±0.88	**12.88** ±1.08	**4.79** ±0.42	**49.6** ±2.43
Korean	8.00 ±2.42	2.10 ±0.79	32.32 ±4.27	14.58 ±2.42	23.51 ±0.77	**19.45** ±3.49	**6.05** ±1.25	**63.4** ±0.94
Chinese	–	13.54 ±3.63	–	53.27 ±42.97	17.06 ±1.12	–	**30.11** ±13.0	**47.9** ±1.52
Japanese	–	6.89 ±1.19	–	15.01 ±2.38	23.76 ±1.41	–	**11.88** ±1.57	**53.9** ±1.44

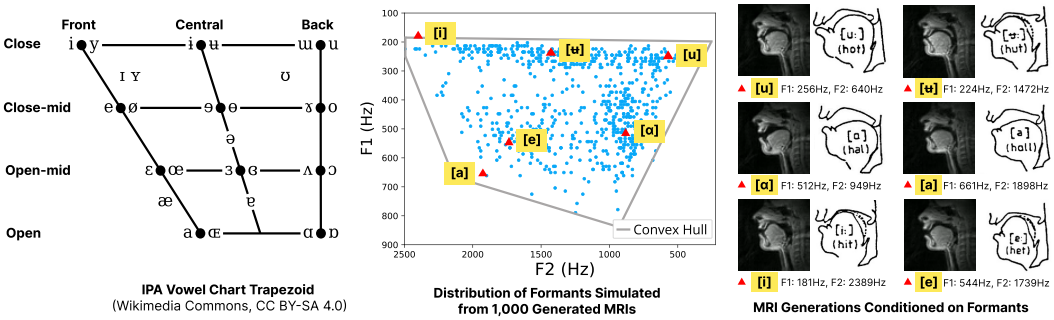


Figure 6: **Phonetic Structure Emerges without Supervision.** We randomly generate 1,000 MRIs from our GAN, then compute their formants with our simulator. The convex hull of formants resemble a trapezoid corresponding almost exactly to the IPA vowel chart. The anatomy of each generated MRI corresponds closely to textbook illustrations of vowels from Fant [17].

start by initializing a sequence of latents, $\theta = w_1, \ldots, w_K$, corresponding to a video $I_1, \ldots, I_K = G(w_1), \ldots, G(w_K)$. We then extract the cross-sectional area from each I, denoted with $\psi(I_k) = A[1\!:\!N, k]$. And finally, we run simulation on A to generate an utterance p_{out}. Overall, $p_{\text{out}} = \Phi(u_{\text{in}}, \psi(G(w_1)), \ldots, \psi(G(w_K)))$. To fit an utterance, we backpropagate the entire process end-to-end, deriving gradients for each w_k. We differentiably segment the cross-sectional areas from each MRI frame by first softly binarizing it with a sigmoid function, then extracting the areas along a guide. The guide, pictured in orange in Figure 1, simply consists of a quarter circle arc and a line segment, connecting the glottis to the lips. Along each orange line, the cross-sectional areas are approximated by summing the binarized pixels. Because this technique does not require paired speech-MRI data it is fully self-supervised.

Phonetic Structure Emerges without Supervision. We explore how physically grounded the audio-visual representation is. In Figure 6, we visualize the formant space generated from our learned representation. We randomly sample 1,000 MRI frames from the GAN, use our simulator to synthesize vowels extracted from their cross-sectional areas, then plot their first two formants. Despite the absence of any explicit supervision, the formant space's convex hull forms a trapezoid almost exactly resembling the trapezoidal vowel chart from the International Phonetic Alphabet (IPA) chart. When compared to vowel examples from [17], we observe a very close correspondence in tongue positioning as well. The fact that the IPA emerges spontaneously from this audio-visual representation gives us confidence that our simulator captures geometrically meaningful vocal tract configurations.

Acoustic Grounding Prevents Video Collapse. Without a physical model to connect geometry to speech, prior works that tackle this task act as black boxes. For example, Speech2rtMRI [43], the closest prior work, trains a diffusion model on paired data between speech and MRI. The quality of this diffusion model is limited by MRI data, which is difficult to collect. Furthermore, because Speech2rtMRI does not leverage a physical representation, the generated videos may lack physical

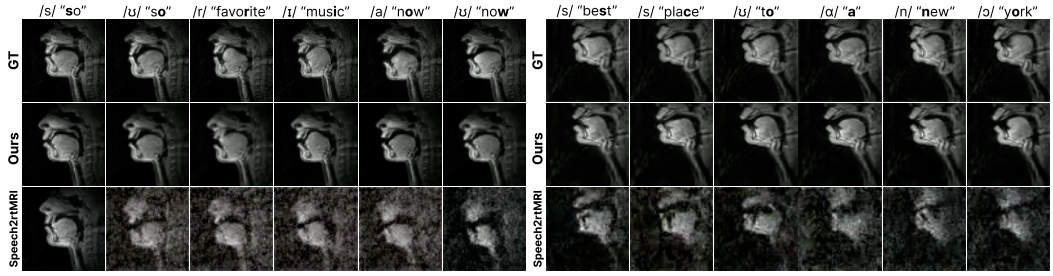


Figure 7: **Acoustic Grounding Prevents Video Collapse.** Because Speech2rtMRI [43] does not leverage a physical representation, it tends to degenerate over time. Our method remains stable.

plausibility. It can only generate 10-20 frames of video. Our "analysis by synthesis" technique on the other hand makes MRI visualization tractable for the first time by physically constraining the video generations with audio. We compare our methods on 40 sequences of 3 seconds, corresponding to 9,900 video frames.

We evaluate each method on their acoustic and visual quality. According to the experimental results, Speech2rtMRI suffers from noise degradation. As illustrated in Figure 7, the diffusion model collapses and the generated images exhibit severe noise and unrealistic artifacts. The quantitative results show that the diffusion model achieves a Fréchet Video Distance (FVD) of 2949, an order of magnitude higher than our value of 623. Despite begin trained on unlabeled data, our method still outperforms Speech2rtMRI on SSIM (structural similarity index measure) and LPIPS (learned perceptual image patch similarity) [62] as well. These two image metrics measure how accurately the MRI's structure matches the ground truth. Our method also simultaneously simulates audio. Despite the challenging task of backpropagating through an MRI generator, it is still able to produce intelligible audio, something not previously possible.

Table 4: **MRI Metrics**

	Ours	Speech 2rtMRI
Audio		
SI-SDR ↑	**13.08**	N/A
STOI ↑	**0.92**	N/A
PESQ ↑	**2.09**	N/A
Visual		
FVD ↓	**623**	2949
SSIM ↑	**0.352**	0.317
LPIPS ↓	**0.159**	0.355

7 Conclusion

We have introduced a differentiable and GPU parallelizable physics-based model of human speech. We have discovered several technical insights along the way: (1) Modeling acoustics in the frequency domain enables stable gradient flow, and is 70x more GPU parallelizable than finite differences in time. (2) We introduce a differentiable model for consonant formation. (3) Parameterizing the area functions with a neural field alleviates the sub-optimality failures common in acoustic inverse problems. Beyond the simulator itself, we enable two applications that were not previously possible. (1) We train a self-supervised autoencoder that predicts the vocal tract's area function directly from raw audio, without any paired articulatory data. (2) We provide a way to visualize an MRI video of one's vocal tract from their speech, without paired speech-MRI training data.

Limitations. We note several directions for future improvement. As shown in Figure 7, our MRI images sometimes do not correspond directly to the ground truth. Part of the reason is that the inverse problem is inherently ill-posed: different geometries can produce the same sound. Furthermore, the MRI dataset used does not image the nasal tract. Reconstructing the nasal tract would produce more faithful formants and anti-formants.

Broader Impacts. While this work is primarily focused on acoustics, we believe that this work opens the door to exciting applications beyond acoustics. Our acoustic simulator can be used, for instance, in music, as a tool for voice coaching or singing synthesis; in cognitive science, such as the study of language acquisition [2]; and in medicine, such as speech pathology. Another direction for future work is to extend our model to other animals, such as sperm whales [52] or zebra finches.

Acknowledgments

This work would not have been possible without discussions with Morris Alper about phonetics; with Mark Rau about acoustic simulation; and with Matthew Caren and Kartik Chandra about their work on vocal imitation. EC is supported by the NSF Graduate Research Fellowship Program.

Special thanks to our foreign language contributors: Clément Jambon for French, David Charatan for German, Chonghyuk (Andrew) Song for Korean, and Emerald Liu for Chinese.

References

[1] Alexei Baevski, Henry Zhou, Abdel rahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *ArXiv*, abs/2006.11477, 2020. URL https://api.semanticscholar.org/CorpusID:219966759. 8, 22

[2] Gašper Beguš, Alan Zhou, Peter Wu, and Gopala K Anumanchipalli. Articulation gan: Unsupervised modeling of articulatory learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 1, 10

[3] Stefan Bilbao. *Acoustic Tubes*, chapter 9, pages 249–286. John Wiley & Sons, Ltd, 2009. ISBN 9780470749012. doi: https://doi.org/10.1002/9780470749012.ch9. 3, 16

[4] Peter Birkholz. Modeling consonant-vowel coarticulation for articulatory speech synthesis. *PLOS ONE*, 8(4):1–17, 04 2013. doi: 10.1371/journal.pone.0060603. URL https://doi.org/10.1371/journal.pone.0060603. 2, 3, 8, 16

[5] Peter Birkholz and Dietmar Jackel. Influence of temporal discretization schemes on formant frequencies and bandwidths in time domain simulations of the vocal tract system. In *Interspeech*, 2004. URL https://api.semanticscholar.org/CorpusID:15404079. 5, 6, 20

[6] Peter Birkholz and Dietmar Jackel. Noise sources and area functions for the synthesis of fricative consonants. 2006. URL https://api.semanticscholar.org/CorpusID:12604988. 6

[7] Robert Bridson, Jim Houriham, and Marcus Nordenstam. Curl-noise for procedural fluid flow. *ACM SIGGRAPH 2007 papers*, 2007. URL https://api.semanticscholar.org/CorpusID:10174968. 6

[8] Matthew Caren, Kartik Chandra, Joshua Tenenbaum, Jonathan Ragan-Kelley, and Karima Ma. Sketching with your voice: "non-phonorealistic" rendering of sounds via vocal imitation. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. URL https://doi.org/10.1145/3680528.3687679. 1

[9] Cheol Jun Cho, Peter Wu, Tejas S. Prabhune, Dhruv Agarwal, and Gopala K. Anumanchipalli. Coding speech through vocal tract kinematics. *IEEE Journal of Selected Topics in Signal Processing*, 18(8):1427–1440, 2024. doi: 10.1109/JSTSP.2024.3497655. 3, 8

[10] Eleanor Chodroff, Blaz Pazon, Annie Baker, and Steven Moran. Phonetic segmentation of the ucla phonetics lab archive. In *International Conference on Language Resources and Evaluation*, 2024. URL https://api.semanticscholar.org/CorpusID:268732610. 22

[11] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k. 22

[12] John R Deller Jr, John G Proakis, and John H Hansen. *Discrete time processing of speech signals*. Prentice Hall PTR, 1993. 4

[13] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020*, pages 3830–3834, 2020. 8

[14] Hugh K Dunn. The calculation of vowel resonances, and an electrical vocal tract. *The Journal of the Acoustical Society of America*, 22(6):740–753, 1950. 17

[15] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. *ArXiv*, abs/2001.04643, 2020. URL https://api.semanticscholar.org/CorpusID:210473083. 3, 6

[16] Gunnar Fant. *Acoustic theory of speech production, with calculations based on X-ray studies of Russian articulations.* Mouton and Co. N.V., The Hague, 1960. 7

[17] Gunnar Fant. *Speech Acoustics and Phonetics*. Text, Speech and Language Technology. Springer Dordrecht, 2004. doi: 10.1007/978-1-4020-5746-5. URL https://api.semanticscholar.org/CorpusID:60121294. 9

[18] Gunnar Fant, Johan Liljencrants, and Qi-guang Lin. A four-parameter model of glottal flow. *STL-QPSR*, 4(1985):1–13, 1985. 3, 17, 21

[19] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2021. URL https://api.semanticscholar.org/CorpusID:235613628. 8, 21, 22

[20] Ugo Finnendahl, Markus Worchel, Tobias Jüterbock, Daniel Wujecki, Fabian Brinkmann, Stefan Weinzierl, and Marc Alexa. Differentiable geometric acoustic path tracing using time-resolved path replay backpropagation. *ACM Transactions on Graphics (TOG)*, 44(4), 2025. doi: 10.1145/3730900. 1

[21] James L. Flanagan. *Speech Analysis Synthesis and Perception*. Communication and Cybernetics. Springer Berlin, Heidelberg, 2 edition, 1972. doi: 10.1007/978-3-662-01562-9. 16, 17

[22] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation, 2021. URL http://github.com/google/brax. 1

[23] Andrew S Glassner. *An introduction to ray tracing*. Morgan Kaufmann, 1989. 1

[24] James K Hahn, Joe Geigel, Jong Won Lee, Larry Gritz, Tapio Takala, and Suneil Mishra. An integrated approach to motion and sound. *The Journal of Visualization and Computer Animation*, 6(2):109–123, 1995. 6

[25] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan A. Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *ArXiv*, abs/1910.00935, 2019. URL https://api.semanticscholar.org/CorpusID:203626832. 1

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2019. URL https://api.semanticscholar.org/CorpusID:209202273. 8, 22

[27] J. L. Jr. Kelly and C. C. Lochbaum. Speech synthesis. In *Proceedings of the Stockholm Speech Communication Seminar*, 1962. 1, 2, 3, 4, 16

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106. 7, 21, 22

[29] Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. Libritts-r: A restored multi-speaker text-to-speech corpus. *ArXiv*, abs/2305.18802, 2023. URL https://api.semanticscholar.org/CorpusID:258967444. 5, 6, 21, 22

[30] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *NeurIPS*, volume 33, pages 17022–17033, 2020. 1

[31] Paul Konstantin Krug, Christoph Wagner, Peter Birkholz, and Timo Stich. Precisely controllable neural speech synthesis. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10890772. 3, 8, 9

[32] Anurag Kumar, Ke Tan, Zhaoheng Ni, Pranay Manocha, Xiaohui Zhang, Ethan Henderson, and Buye Xu. Torchaudio-squim: Reference-less speech quality and intelligibility measures in torchaudio. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. URL https://api.semanticscholar.org/CorpusID:257921409. 6, 21

[33] Heinrich Kuttruff. *Room acoustics*. Crc Press, 2016. 1

[34] Zitong Lan, Chenhao Zheng, Zhiwei Zheng, and Mingmin Zhao. Acoustic volume rendering for neural impulse response fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 3

[35] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Av-nerf: Learning neural fields for real-world audio-visual scene synthesis. *Advances in Neural Information Processing Systems*, 36:37472–37490, 2023. 1, 3

[36] Yongwan Lim, Asterios Toutios, Yannick Bliesener, Ye Tian, Sajan Goud Lingala, Colin Vaz, Tanner Sorensen, Miran Oh, Sarah Harper, Weiyi Chen, Yoonjeong Lee, Johannes Töger, Mairym Lloréns Montesserin, Caitlin Smith, Bianca Godinez, Louis Goldstein, Dani Byrd, Krishna S Nayak, and Shrikanth Narayanan. A multispeaker dataset of raw and reconstructed speech production real-time MRI video and 3D volumetric images. 2 2021. doi: 10.6084/m9.figshare.13725546.v1. URL https://figshare.com/articles/dataset/A_multispeaker_dataset_of_raw_and_reconstructed_speech_production_real-time_MRI_video_and_3D_volumetric_images/13725546. 8, 22

[37] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35: 3165–3177, 2022. 3

[38] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, March 2022. NVIDIA GPU Technology Conference (GTC). 1

[39] Shinji Maeda. A digital simulation method of the vocal-tract system. *Speech Communication*, 1 (3):199–229, 1982. ISSN 0167-6393. doi: https://doi.org/10.1016/0167-6393(82)90017-6. URL https://www.sciencedirect.com/science/article/pii/0167639382900176. 1, 2, 3, 6, 16, 20

[40] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 8, 22

[41] Sean L. Metzger, Kaylo T Littlejohn, Alexander B. Silva, David Aaron Moses, Margaret P. Seaton, Ran Wang, Maximilian E. Dougherty, Jessie R. Liu, Peter Wu, Michael Berger, Inga Zhuravleva, Adelyn P. Tu-Chan, Karunesh Ganguly, Gopala Krishna Anumanchipalli, and Edward F. Chang. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620:1037–1046, 2023. URL https://api.semanticscholar.org/CorpusID:261098775. 1

[42] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 3, 7

[43] Hong Nguyen, Sean Foley, Kevin Huang, Xuan Shi, Tiantian Feng, and Shrikanth S. Narayanan. Speech2rtmri: Speech-guided diffusion model for real-time mri video of the vocal tract during speech. *ArXiv*, abs/2409.15525, 2024. URL https://api.semanticscholar.org/CorpusID:272832441. 2, 3, 9, 10

[44] Lars Nieradzik. Swiftf0: Fast and accurate monophonic pitch detection, 2025. URL https://arxiv.org/abs/2508.18440. 21

[45] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 1

[46] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. URL https://api.semanticscholar.org/CorpusID:2191379. 22

[47] Sankaran Panchapagesan and Abeer Alwan. A study of acoustic-to-articulatory inversion of speech by analysis-by-synthesis using chain matrices and the maeda articulatory model. *The Journal of the Acoustical Society of America*, 129 4:2144–62, 2011. URL https://api.semanticscholar.org/CorpusID:18781420. 2, 7

[48] Kyubyong Park. Kss dataset: Korean single speaker speech dataset, 2018. URL https://kaggle.com/bryanpark/korean-single-speaker-speech-dataset. 22

[49] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325247. URL https://doi.org/10.1145/325165.325247. 6

[50] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:252923993. 8

[51] Kilian Schulze-Forster, Gaël Richard, Liam Kelley, Clement S. J. Doire, and Roland Badeau. Unsupervised music source separation using differentiable parametric source models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1276–1289, 2022. 3

[52] Pratyusha Sharma, Shane Gero, Roger Payne, David F Gruber, Daniela Rus, Antonio Torralba, and Jacob Andreas. Contextual and combinatorial structure in sperm whale vocalisations. *Nature Communications*, 15, 2023. URL https://api.semanticscholar.org/CorpusID:266150065. 10

[53] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. 2015. URL https://arxiv.org/abs/2010.11567. 22

[54] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems (NeurIPS 2020)*, 33:7462–7473, 2020. 3, 7

[55] Man Sondhi and J. Schroeter. A hybrid time-frequency domain articulatory speech synthesizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(7):955–967, 1987. doi: 10.1109/TASSP.1987.1165240. 2, 6, 16, 20

[56] Jos Stam and Eugene Fiume. Turbulent wind fields for gaseous phenomena. *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993. URL https://api.semanticscholar.org/CorpusID:1618202. 6

[57] David Südholt, Mateo Cámara, Zhiyuan Xu, and Joshua D Reiss. Vocal tract area estimation by gradient descent. 2023. 2, 3

[58] Shinnosuke Takamichi, Kentaro Mitsui, Yuki Saito, Tomoki Koriyama, Naoko Tanji, and Hiroshi Saruwatari. Jvs corpus: free japanese multi-speaker voice corpus. *ArXiv*, abs/1908.06248, 2019. URL https://api.semanticscholar.org/CorpusID:201070145. 22

[59] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 8, 22

[60] Mason Wang, Ryosuke Sawata, Samuel Clarke, Ruohan Gao, Shangzhe Wu, and Jiajun Wu. Hearing anything anywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3

[61] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415, 2019. 1, 3

[62] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. URL https://api.semanticscholar.org/CorpusID:4766599. 10

A Physics of the Vocal Tract

We understand that the physics of speech production may be unfamiliar to the broader machine learning community, so we provide a derivation of the speech simulation method in this section. For a comprehensive introduction to speech, we refer the reader to Flanagan [21], or Bilbao [3] for a more general overview on physical modeling synthesis.

A.1 Derivation of the Linearized Euler Equations

In the most general form, we can model the fluid dynamics of the vocal tract with Euler's equations for compressible flow, over 3D space $\mathbf{x} \in \mathbb{R}^3$ and time $t \in \mathbb{R}$. For a pressure field $p : (\mathbf{x}, t) \mapsto \mathbb{R}$, velocity field $\mathbf{v} : (\mathbf{x}, t) \mapsto \mathbb{R}^3$, and density field $\rho : (\mathbf{x}, t) \mapsto \mathbb{R}_{\geq 0}$, the first equation represents conservation of mass, while the second represents conservation of momentum. Subscripts $_t$ and $_x$ denote partial derivatives in time and space respectively.

$$\rho_t = -\nabla \cdot (\rho \mathbf{v}) \qquad \text{mass} \tag{8}$$

$$-\nabla p = \rho(\mathbf{v}_t + (\mathbf{v} \cdot \nabla)\mathbf{v}) \qquad \text{momentum} \tag{9}$$

The pressure at the outlet of the vocal tract, which we will denote with p_{out}, is what we perceive as speech. These equations are difficult to model numerically, but by applying a series of simplifying assumptions, we can derive an entire hierarchy of governing equations for vocal tract acoustics.

First, when modeling sound propagation, p and ρ may be defined as perturbations around equilibrium states: $\rho = \rho_0 + \rho'$, $p = p_0 + p'$. Substituting these linearizations into Euler's equations, we have

$$(\rho_0 + \rho')_t = -\nabla \cdot ((\rho_0 + \rho')\mathbf{v}) \tag{10}$$

$$-\nabla(p_0 + p') = (\rho_0 + \rho')(\mathbf{v}_t + (\mathbf{v} \cdot \nabla)\mathbf{v}). \tag{11}$$

Because v, ρ' and p' are assumed to be small, all second-order terms may be eliminated. Further, we may assume that $p' = c^2\rho'$, where c is the speed of sound, eliminating the need for ρ'. The resulting equations are a common form of Euler's equations used in acoustics. Note that by linearization, we lose the ability to synthesize turbulent flow.

$$\frac{1}{c^2} p'_t = -\rho_0 \nabla \cdot \mathbf{v} \qquad \text{mass} \tag{12}$$

$$-\nabla p' = \rho_0 \mathbf{v}_t \qquad \text{momentum} \tag{13}$$

For frequencies less than 4 kHz, the acoustics of the vocal tract are dominated by plane waves: waves traveling perpendicular to the vocal tract's cross section $\mathrm{dA} = \mathrm{A(x)}\ \mathrm{d}\hat{\mathbf{n}}$. This is because the wavelength of a 4 kHz wave traveling at the speed of sound (343 m/s) is 8.6 cm, significantly larger than the diameter of the vocal tract (2-3 cm). This is a common simplification used in prior works [27, 39, 55, 4]. We can model planar flow by rewriting Eq. 12 and Eq. 13 in terms of a one-dimensional flux, $u(x,t) = \int_{A(x)} \mathbf{v} \cdot \mathrm{dA}$. To model this flux (also known as volume velocity), we integrate the governing equations over a control volume V.

$$\int_V \frac{1}{c^2} p'_t \ \mathrm{dV} = -\rho_0 \int_V (\nabla \cdot \mathbf{v}) \ \mathrm{dV} \tag{14}$$

$$-\int_V \nabla p' \ \mathrm{dV} = \rho_0 \int_V \mathbf{v}_t \ \mathrm{dV}. \tag{15}$$

In the limit, $\mathrm{dV} = \mathrm{dA}\ \mathrm{dx}$. Further assuming that $p'(\mathbf{x}, t)$ is constant over each dA, the 3D field $p'(\mathbf{x}, t)$ can be replaced with a simpler 1D field: $p'(x, t)$. By applying the divergence theorem and expanding the integrals, we find that

$$A\Delta x \frac{1}{c^2} p'_t = -\rho_0 \int_{A(x)+\Delta x} \mathbf{v} \cdot \ \mathrm{dA} - \rho_0 \int_{\mathrm{A(x)}} \mathbf{v} \cdot \ \mathrm{dA} \tag{16}$$

$$-A\Delta x \ p'_x = \rho_0 \int_x^{x+\Delta x} \int_A \mathbf{v}_t \ \cdot \mathrm{dA}\ \mathrm{dx}. \tag{17}$$

Finally, substituting u, and taking the limit $\Delta x \mapsto 0$, we derive Euler's equations in terms of u and A. For simplicity, p' is often written as just p, and ρ_0 is written as ρ, the ambient density of air.

$$\frac{A}{c^2} p_t = -\rho u_x \qquad \text{mass} \tag{18}$$

$$-A\ p_x = \rho u_t \qquad \text{momentum} \tag{19}$$

These are the basic equations for modeling acoustics in the vocal tract, and can be easily augmented with additional dampening parameters for the vocal tract's material properties [21]. The equations can be written even more compactly by differentiating the mass equation by t, the momentum equation by x, then combining them a single equation, known as Webster's equation:

$$(A(x)p_x)_x = \frac{A(x)}{c^2} p_{tt} \tag{20}$$

Importantly, Webster's equation resembles the wave equation, demonstrating how the geometry of the vocal tract, controlled by $A(x)$, affects the wave propagation of speech. After applying the frequency domain separation of variables $p(x,t) = P(x)e^{-j\omega t}$, $u(x,t) = U(x)e^{-j\omega t}$, Webster's equation becomes

$$(A(x)P(x)_x)_x = -\frac{\omega^2}{c^2} A(x)P(x), \tag{21}$$

resembling the Helmholtz equation. Solving the linearized Euler equations or Webster's equation is thus equivalent to solving the Helmholtz eigenvalue problem in the frequency domain.

To generate vowels, a glottal pulse u_{in} parameterized by the LF model [18] is set as the initial condition, and is propagated to the other end of the tract to synthesize p_{out}. Solving the linearized Euler equations or Webster's equation derives a filter h_{tract}, such that $p_{\text{out}} = h_{\text{tract}} * u_{\text{in}}$.

The following boundary conditions at the glottis and the lip are typically adopted to solve the system.

$$u(0,t) = u_{\text{in}}(t) \qquad \text{Glottis boundary} \tag{22}$$

$$R_{\text{lips}} p(L,t) + L_{\text{lips}} p_t(L,t) = R_{\text{lips}} L_{\text{lips}} u_t(L,t) \qquad \text{Lip boundary} \tag{23}$$

The lip opening area, $A(L)$, controls how much sound is reflected back into the tract versus how much is dissipated outside via an inductance term $L_{\text{lips}} = \frac{8\rho}{3\pi\sqrt{\pi A(L)}}$ and resistance term $R_{\text{lips}} = \frac{128\rho c}{9\pi^2 A(L)}$ (with coefficients derived by Flanagan [21]).

A.2 Circuit Interpretation and the Transmission Line Model

Interpreting the fluid dynamics of the vocal tract as a circuit model has been long studied [14]. Considering a piecewise-constant area function with N cylindrical sections where each n-th section has of length l_n and area A_n and taking the Fourier transform of Eq. (18)-(19) gives the following frequency-domain governing equations:

$$j\omega C_n P_n(\omega) = U_n(\omega) - U_{n+1}(\omega), \tag{24}$$

$$j\omega L_n U_n(\omega) = P_{n-1}(\omega) - P_n(\omega), \tag{25}$$

where $C_n = A_n l_n/(\rho_0 c^2)$ is the acoustic *compliance* and $L_n = \rho_0 l_n/(2A_n)$ is the acoustic *inductance* of the half-section. This describes a lossless ideal fluid flow with purely imaginary impedances.

Extension with viscous losses. More realistic fluid result from a viscous boundary layer at the cylinder wall. This can be modeled by imposing the no-slip condition at the wall, so the resulting wall shear stress acts as a drag on the fluid. This 'wall friction' due to this boundary layer is known to be well-modeled by introducing an additional dissipative term in the momentum equation:

$$-\nabla p = L_n \dot{u} + r_{\text{wf}}(t) * u \tag{26}$$

where $*$ denotes convolution and $r_{\text{wf}}(t)$ is the impulse response of the wall-friction operator. For a circular cylinder of perimeter S_n and cross-sectional area A_n, the viscous boundary-layer theory gives the frequency-domain expression of this wall-friction operator as:

$$R_n(x,\omega) = \frac{S_n(x)}{2A_n^2(x)} \sqrt{\frac{\rho_0 \omega \mu}{2}} \tag{27}$$

which can be analogously interpreted as a frequency-dependent *resistance* $R_n(\omega) \propto \sqrt{\omega}$. Taking the Fourier transform to Eq. (26) gives the follows.

$$P_{n-1}(\omega) - P_n(\omega) = (j\omega L_n + R_n(\omega))U_n(\omega)$$

The series impedance of the cylinder at the n-th section is therefore

$$Z_{s,n}(\omega) = R_n(\omega) + j\omega L_n. \tag{28}$$

Extension with yielding walls. In addition to modeling the fluid flow, the acoustic modeling of the vocal tract wall can be approached using a damped harmonic oscillator (equivalently the mass-spring-damper). Denoting the wall mass, damping coefficient, and stiffness coefficient as m_w, b_w, and k_w, respectively, the normal displacement of the wall ξ_n satisfies

$$m_w\ddot{\xi}_n + b_w\dot{\xi}_n + k_w\xi_n = p(x,t). \tag{29}$$

In the frequency domain, Eq. (29) gives a wall displacement $\Xi_n = P/(-\omega^2 m_w + j\omega b_w + k_w)$, so the wall presents a shunt impedance to the acoustic field. For n-th section with lateral surface area $S_n l_n$, the lumped wall impedance is

$$Z_{w,n} = \frac{-\omega^2 m_w + j\omega b_w + k_w}{S_n l_n \omega^2} = \frac{R_{w,n} + j\omega L_{w,n} + \frac{1}{j\omega C_{w,n}}}{j\omega}$$

Combined with the acoustic compliance C_n in parallel, the total shunt admittance is

$$Z_{w,n}(\omega) = \left(R_{w,n} + j\omega L_{w,n} + \frac{1}{j\omega C_{w,n}}\right) \parallel \frac{1}{j\omega C_n} \tag{30}$$

where $L_{w,n} = m_w/(S_n l_n)$, $R_{w,n} = b_w/(S_n l_n)$, and $C_{w,n} = (S_n l_n)/k_w$.

A.3 Transfer Functions

Combining Eq. (28) and Eq. (30), each n-th section of the acoustic tube is represented in the frequency domain as the *T-network* where a series arm impedance $Z_{s,n}$ carries momentum losses and a shunt arm $Z_{w,n}$ stores compliance and wall losses. Finally, as described in Eq. (3), the two-port transfer matrix for section n is

$$\begin{bmatrix} P_{n+1} \\ U_{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 + Z_{s,n}/Z_{w,n} & -2Z_{s,n} - Z_{s,n}^2/Z_{w,n} \\ -1/Z_{w,n} & 1 + Z_{s,n}/Z_{w,n} \end{bmatrix}}_{\mathbf{K}_n(\omega)} \begin{bmatrix} P_n \\ U_n \end{bmatrix} \tag{31}$$

with a section matrix

$$\mathbf{K}_n = \begin{bmatrix} \mathbf{A}_n & \mathbf{B}_n \\ \mathbf{C}_n & \mathbf{D}_n \end{bmatrix}.$$

This can be used to construct partial chain matrices by concatenating the individual section matrices:

$$\mathbf{K}_{0\mapsto n} := \mathbf{K}_n\mathbf{K}_{n-1}\cdots\mathbf{K}_1 = \begin{bmatrix} \mathbf{A}_{0n} & \mathbf{B}_{0n} \\ \mathbf{C}_{0n} & \mathbf{D}_{0n} \end{bmatrix}, \tag{32}$$

$$\mathbf{K}_{n\mapsto N+1} := \mathbf{K}_N\cdots\mathbf{K}_{n+1} = \begin{bmatrix} \mathbf{A}_{nN} & \mathbf{B}_{nN} \\ \mathbf{C}_{nN} & \mathbf{D}_{nN} \end{bmatrix}. \tag{33}$$

Note that chain matrices have unit determinant for reciprocal media, *i.e.*, $\det \mathbf{K}_{0\mapsto n} = \det \mathbf{K}_{n\mapsto N+1} = 1$.

Transfer function from glottis to section n. To compute $H_{0\mapsto n}$, the transfer function from glottis to section n, apply Eq. (31) for $\mathbf{K}_{0\mapsto n}$:

$$\begin{bmatrix} P_n \\ U_n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{0n} & \mathbf{B}_{0n} \\ \mathbf{C}_{0n} & \mathbf{D}_{0n} \end{bmatrix} \begin{bmatrix} P_{\text{in}} \\ U_{\text{in}} \end{bmatrix} \tag{34}$$

where P_{in} is the pressure at the glottal plane and U_{in} is the glottal volume velocity. Imposing the boundary condition at the glottis, $P_{\text{in}} = Z_{\text{in}}U_{\text{in}}$,

$$U_n = (\mathbf{C}_{0n}Z_{\text{in}} + \mathbf{D}_{0n})U_{\text{in}},$$

$$G_{0\mapsto n}(\omega) = \frac{U_n}{U_{\text{in}}} = \mathbf{C}_{0n}Z_{\text{in}} + \mathbf{D}_{0n}. \tag{35}$$

The backward impedance seen from section n looking toward the glottis follows from the same substitution.

$$Z_1 = -\frac{P_n}{U_n} = -\frac{\mathbf{A}_{0n}Z_{\text{in}} + \mathbf{B}_{0n}}{\mathbf{C}_{0n}Z_{\text{in}} + \mathbf{D}_{0n}}$$

The sign convention for $-U_n$ indicates that U_n is flowing away from glottis and thus into the Z_1 load when viewed from section n.

Transfer function from section n to lips. From boundary condition Eq. (23), the radiation impedance $Z_{\text{lips}} = R_{\text{lips}} \parallel j\omega L_{\text{lips}}$ can be derived with

$$L_{\text{lips}} = \frac{8\rho_0}{3\pi\sqrt{\pi A_N}} \qquad \text{and} \qquad R_{\text{lips}} = \frac{128\rho_0 c}{9\pi^2 A_N}. \tag{36}$$

Now, apply $\mathbf{K}_{n\mapsto N+1}$ from section n to the lip plane, where the boundary condition $P_{N+1} = Z_{\text{lips}} U_{N+1}$ must be satisfied:

$$\begin{bmatrix} P_{N+1} \\ U_{N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{nN} & \mathbf{B}_{nN} \\ \mathbf{C}_{nN} & \mathbf{D}_{nN} \end{bmatrix} \begin{bmatrix} P_n \\ U_n \end{bmatrix}, \tag{37}$$

giving

$$P_{N+1} = \mathbf{A}_{nN} P_n + \mathbf{B}_{nN} U_n, \tag{38}$$

$$U_{N+1} = \mathbf{C}_{nN} P_n + \mathbf{D}_{nN} U_n. \tag{39}$$

Substituting $P_{N+1} = Z_{\text{lips}} U_{N+1}$ into Eq. (38) and (39) to eliminate P_{N+1}:

$$Z_{\text{lips}} U_{N+1} = \mathbf{A}_{nN} P_n + \mathbf{B}_{nN} U_n. \tag{40}$$

Divide Eq. (40) by (39):

$$Z_{\text{lips}} = \frac{\mathbf{A}_{nN} P_n + \mathbf{B}_{nN} U_n}{\mathbf{C}_{nN} P_n + \mathbf{D}_{nN} U_n} \qquad \Longrightarrow \qquad Z_2 := \frac{P_n}{U_n} = \frac{\mathbf{D}_{nN} Z_{\text{lips}} - \mathbf{B}_{nN}}{\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}}} \tag{41}$$

which is the forward impedance seen from section n towards the lips. Eliminating P_n using $P_n = Z_2 U_n$ yields:

$$U_{N+1} = (\mathbf{C}_{nN} Z_2 + \mathbf{D}_{nN})\, U_n = \frac{\mathbf{C}_{nN}(\mathbf{D}_{nN} Z_{\text{lips}} - \mathbf{B}_{nN}) + \mathbf{D}_{nN}(\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}})}{\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}}} U_n.$$

Because $\mathbf{A}_{nN}\mathbf{D}_{nN} - \mathbf{B}_{nN}\mathbf{C}_{nN} = \det \mathbf{K}_{n\mapsto N+1} = 1$, the numerator simplifies, leaving

$$G_{n\mapsto N+1} = \frac{U_{N+1}}{U_n} = \frac{1}{\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}}}. \tag{42}$$

The pressure transfer function from n to lips is therefore

$$H_{n\mapsto N+1} = \frac{P_{\text{out}}}{U_n} = \frac{Z_{\text{lips}}}{\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}}}. \tag{43}$$

Combining the full and noise transfer functions. The glottis-to-lips volume velocity transfer is obtained by applying $\mathbf{K}_{\text{tot}} = \mathbf{K}_N \mathbf{K}_{N-1} \cdots \mathbf{K}_1$ as

$$\begin{bmatrix} P_{N+1} \\ U_{N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{0N} & \mathbf{B}_{0N} \\ \mathbf{C}_{0N} & \mathbf{D}_{0N} \end{bmatrix} \begin{bmatrix} P_{\text{in}} \\ U_{\text{in}} \end{bmatrix}. \tag{44}$$

Setting $P_{\text{in}} = Z_{\text{in}} U_{\text{in}}$ at the glottis and imposing $P_{N+1} = Z_{\text{lips}} U_{N+1}$ at the lips gives

$$\mathbf{A}_{0N} Z_{\text{in}} + \mathbf{B}_{0N} = Z_{\text{lips}}(\mathbf{C}_{0N} Z_{\text{in}} + \mathbf{D}_{0N})$$

therefore the input impedance of the tract seen from the glottis is

$$Z_{\text{in}} = \frac{\mathbf{D}_{0N} Z_{\text{lips}} - \mathbf{B}_{0N}}{\mathbf{A}_{0N} - \mathbf{C}_{0N} Z_{\text{lips}}}. \tag{45}$$

With Z_{in} established, substitute Eq. (45) into $H_{\text{tract}} = P_{N+1}/U_{\text{in}} = \mathbf{A}_{0N} Z_{\text{in}} + \mathbf{B}_{0N}$ gives:

$$H_{\text{tract}}(\omega) = \mathbf{A}_{0N} \left(\frac{\mathbf{D}_{0N} Z_{\text{lips}} - \mathbf{B}_{0N}}{\mathbf{A}_{0N} - \mathbf{C}_{0N} Z_{\text{lips}}} \right) + \mathbf{B}_{0N}, \tag{46}$$

$$= \frac{\mathbf{A}_{0N}(\mathbf{D}_{0N} Z_{\text{lips}} - \mathbf{B}_{0N}) + \mathbf{B}_{0N}(\mathbf{A}_{0N} - \mathbf{C}_{0N} Z_{\text{lips}})}{\mathbf{A}_{0N} - \mathbf{C}_{0N} Z_{\text{lips}}}. \tag{47}$$

The numerator expands to $(\mathbf{A}_{0N}\mathbf{D}_{0N} - \mathbf{B}_{0N}\mathbf{C}_{0N}) Z_{\text{lips}} = \det \mathbf{K}_{\text{tot}} \cdot Z_{\text{lips}} = Z_{\text{lips}}$. Therefore, the total pressure transfer function is

$$H_{\text{tract}} = \frac{Z_{\text{lips}}}{\mathbf{A}_{nN} - \mathbf{C}_{nN} Z_{\text{lips}}} \tag{48}$$

As described in Subsection 5.2, the propagation of the volume-velocity noise source $U_{\text{noise}_n}(\omega)$ to the lips is is only governed by $\mathbf{K}_{n\mapsto N+1}$, and therefore the pressure transfer function for the noise from the section n is identical to $H_{n\mapsto N+1}$. This confirms that the total combined transfer function from the glottal pulse and the noise to the lips as Eq. (7).

A.4 The Nasal Tract

Although we do not pursue nasal tract reconstruction because the nasal tract does not appear in the MRI frames, the transmission line model can be used to model it. In fact, with the TLM method, the nasal effect can be modeled as a bifurcated transmission line, and the combination of these two LTI systems can be represented as an equivalent circuit consisting of a single transmission line. In the present work as well, although the nasal cavity is not explicitly reconstructed, it can be viewed as being modeled using a lumped equivalent area function. Examples of this technique are provided in Sondhi and Schroeter [55].

B Finite Difference Techniques

B.1 Time Domain Governing Equation

Solving Eq. (18) with the wall-friction momentum equation Eq. (26) requires a convolution with a half-order integro-differential operator for frequency-dependent resistance $R(\omega) \propto \sqrt{\omega}$. In practice, the resistance is implemented using Hagen–Poiseuille DC resistance [39]

$$R_n^{\mathrm{DC}} = \frac{4\mu l_n \pi}{A_n^2}$$

which is the Stokes flow resistance for a circular duct in the zero-frequency limit. With this substitution, all terms in the circuit equations for the N-section vocal tract model reduce to the following coupled ODEs.

$$\dot{p}_n = \frac{1}{C_n}(u_n - u_{n+1} - u_{w,n}) \qquad \text{mass} \qquad (49)$$

$$(L_{n-1} + L_n)\dot{u}_n + (R_{n-1} + R_n)u_n = p_{n-1} - p_n \qquad \text{momentum} \qquad (50)$$

$$L_{w,n}\ddot{u}_{w,n} + R_{w,n}\dot{u}_{w,n} + \frac{u_{w,n}}{C_{w,n}} = \dot{p}_n \qquad \text{wall vibration} \qquad (51)$$

The expression for the wall vibration can be deduced from Eq. (29) by taking time derivative and denoting $u_{w,n} = S_n l_n \dot{\xi}_n$. The last section $i = N$ terminates into the radiation impedance $Z_{\mathrm{lips}} = R_{\mathrm{lips}} \parallel j\omega L_{\mathrm{lips}}$. In the time domain this parallel R–L load introduces two additional flow unknowns u_{N+1} (total lip flow) and u_{N+2} (inductive branch flow):

$$\dot{u}_{N+1}L_N + u_{N+1}(R_N + R_{\mathrm{lips}}) - u_{N+2}R_{\mathrm{lips}} = p_N, \qquad (52)$$

$$\dot{u}_{N+2}L_{\mathrm{lips}} + u_{N+2}R_{\mathrm{lips}} - u_{N+1}R_{\mathrm{lips}} = 0. \qquad (53)$$

The radiated pressure (voltage across R_{lips}) is

$$p_{\mathrm{out}}(t) = R_{\mathrm{lips}}\big(u_{N+1}(t) - u_{N+2}(t)\big) = L_{\mathrm{lips}}\,\dot{u}_{N+2}(t). \qquad (54)$$

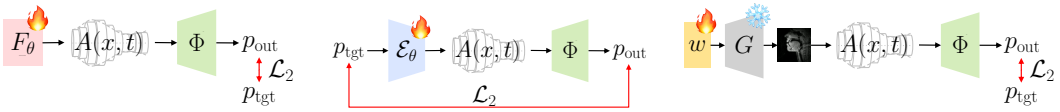
These equations describe the governing equations in continuous time-domain. The only departure from the frequency-domain expressions is the substitution of $R_n(\omega)$ by R_n^{DC}.

B.2 Numerical Scheme

In order to solve the time-domain governing equation on a discrete stencil, the semi-implicit scheme has been employed:

$$\dot{f} = \frac{f - f'}{\Delta t\,\vartheta} - \frac{\bar{\vartheta}}{\vartheta}\,\dot{f}', \qquad \bar{\vartheta} = 1 - \vartheta. \qquad (55)$$

While various schemes may result depending on the value of ϑ, in practice, we set $\vartheta = 0.5$ (trapezoidal) to perform time-domain simulations. The process of solving Subsection B.1 with this scheme reduces to a tridiagonal linear system, which can be resolved through the time-domain recursion and `jax.linalg.tridiagonal_solve`. Please note that in this study, this time-domain simulation serves only as a baseline for validating the performance of the employed frequency-domain simulator. For further details regarding the time-domain implementation, please refer to Birkholz and Jackel [5].



(a) Neural field training (b) Self-supervised autoencoding (c) GAN latent optimization

Figure 8: A schematic diagram illustrating three application cases of the differentiable Φ, as described in Section 6. The problem of estimating $A(x,t)$ from speech can thus be solved through various gradient-based optimization techniques, and *none* of these methods presuppose a ground truth A, but rather resolve this inverse problem solely by utilizing the real-world speech signal p_{tgt}.

C Experimental Details

C.1 Simulation Details

The simulation is modeled in centimeters. For all experiments, we use $N = 32$ spatial segments. For the benchmarks in the paper, and in general, we run our simulation at 16 kHz. Higher fidelity can however be achieved with higher sample rates. For the singing examples in the supplementary material, we use sample rates of 32 kHz - 44.1 kHz.

Glottal Pulse. To estimate the glottal pulse u_{in} of each signal, we first extract fundamental frequencies from the speech sample using Swift-F0 [44]. The fundamental frequencies are then used to construct a glottal pulse using the LF model [18]. Its time-varying amplitude is determined by the RMS energy of the signal, measured in Librosa. The pulse is normalized so it has a maximum value of 1 cm^3/s. Generally, we find that gradient-based optimization is insensitive to the glottal pulse used. The spectrogram loss focuses gradients on formants, which are a property of the vocal tract's resonances and not the glottal pulse. To show robustness to glottal pulse, we include a vocal tract reconstruction of a song played on a piano in the supplementary material. The model is able to create an acapella reconstruction of the song, Ode to Joy, while maintaining a human-like timbre.

C.2 Loss Function

As described in Section 4, we use a multiscale log Mel spectrogram loss to fit the speech samples. We use $n_{\text{mel}} = 128$ Mel bins, FFT sizes of $(512, 1024, 2048)$, window lengths of $(160, 400, 800)$, and hop lengths of $(40, 80, 160)$. This builds three configurations for the spectrogram resolution: $\mathcal{C} = \{(512, 160, 40), (1024, 400, 80), (2048, 800, 160)\}$. For a target pressure p_{tgt} and an output pressure $p_{\text{out}} \triangleq \Phi(u_{\text{in}}, A_\theta)$, the loss is computed using the L_2 norm of the log Mel spectrogram distances under three configurations $(n_{\text{fft}}, n_{\text{win}}, n_{\text{hop}}) \in \mathcal{C}$ averaged over the choices

$$\mathcal{L}_2(p_{\text{out}}, p_{\text{tgt}}) = \frac{1}{|\mathcal{C}|} \sum_{\mathcal{C}} \left\| \log \text{mel}(p_{\text{out}}) - \log \text{mel}(p_{\text{tgt}}) \right\|_2 . \tag{56}$$

Here, log mel is the log Mel spectrogram transformation using the Mel filterbank $\mathbf{M} \in \mathbb{R}^{n_{\text{mel}} \times n_{\text{freq}}}$ and the short-time Fourier transformation (STFT) that outputs the spectrogram $\mathbf{p} \in \mathbb{R}^{n_{\text{freq}} \times K}$, where $n_{\text{freq}} = \lfloor n_{\text{fft}}/2 \rfloor + 1$ is the number of STFT frequency bins and $K = \lceil (n_{\text{sample}} - n_{\text{window}})/n_{\text{hop}} \rceil + 1$ is the number of its window chunks. The m-th Mel bin of the k-th window of the log Mel spectrogram is computed as $\log \text{mel}(p)[m,k] = \log(\sum_j \mathbf{M}[m,j] \cdot |\mathbf{p}[j,k]|)$. It is well-known that this trick of employing a multi-resolution spectrogram can distribute the resolution dependency of the loss calculation, thereby facilitating smoother optimization.

C.3 Evaluation Setup for Perceptual Speech Quality Metrics

We evaluate how well each method can fit 25 random speech samples from LibriTTS-R [29]. For the results in Table 1, we parametrize each area function with a multiplicative filter network [19] (this choice is explained in Subsection 6.1). Gradient descent is done with the Adam optimizer [28] with a learning rate of 10^{-2} for 200 steps.

Using TorchAudio-Squim [32], we report SI-SDR (scale-invariant signal-to-distortion ratio), STOI (short-time objective intelligibility), and PESQ (wideband perceptual evaluation of speech quality).

C.4 Neural Fields

We study two neural field parameterizations F_θ in the paper, a 4-layer fully connected network with random Fourier feature (RFF) positional encoding [59], and a 3-layer multiplicative filter network (MFN) [19]. The RFF network has a 64-dimensional encoding, and a 256-dimensional hidden dimension. The MFN network had an input scale of 256, and 256-dimenaional hidden dimension. To constrain the area functions to be positive, we apply a softmax to the output, and clip the minimum values at 10^{-2}. All models were trained with the Adam optimizer [28] for 200 steps and a learning rate of 10^{-2}. To fit a 5 second speech sample, training takes approximately 1 minute.

Neural Fields Improve Acoustic Reconstruction. We also study how each field performs at reconstructing real-world speech in Table 5. Using the experimental setup in Appendix C.3, we reconstruct the area functions for 25 speech samples from [29]. We find that both neural fields significantly outperform the discrete case across all speech metrics.

Table 5: **Neural Field Reconstruction**

	Discrete	**RFF** [59]	**MFN** [19]
SI-SDR	10.78 ±2.22	14.51 ±3.78	**16.39** ±2.42
STOI	0.81 ±0.02	0.84 ±0.05	**0.93** ±0.02
PESQ	1.57 ±0.12	1.76 ±0.26	**1.98** ±0.26

C.5 Autoencoding

We use the Wav2Vec 2.0 [1] architecture for the autoencoder. Raw waveforms are input into a CNN with the same dimension as the original Wav2Vec2 model. The CNN embeddings are then passed into a 12 layer transformer with 12 heads, and a hidden dimension of 768. The model is trained with the Adam optimizer [28] at a learning rate of 10^{-4}. The model is trained on 3 second segments of English and a batch size of 64 for 15,000 iterations, equating to 800 hours of training audio. It is then fine-tuned for each other language for 10 epochs.

We train our encoder on LibriTTS-R [29] for English, Multilingual LibriSpeech [46] for other European languages, KSS [48] for Korean, JVS [58] for Japanese, and AISHELL-3 [53] for Chinese.

C.5.1 UMAP Projection

UMAP [40] projections are performed on area functions encoded from the VoxAngeles dataset [10]. To generate the UMAP projections, we used 12 neighbors and a minimum distance of 0.2. However, we found that the UMAP projections are generally robust to these hyperparameters. In Figure 9, we show an UMAP projection on an even larger set of IPA phonemes. The area functions remain disentangled by phoneme, and still form a continuous spectrum.

C.6 MRI Generation

A StyleGAN2 [26] network is trained on the MRI data of [36]. Using the identity labels from the dataset, the network is conditioned on these labels to preserve identity during reconstruction. to The model is trained using the default configurations for 64x64 images. During reconstruction, we use $K = 30$ latent vectors to reconstruct 3 second clips at a time.

The audio recorded from the MRI dataset [36] is extremely noisy due to the spinning magnets in the MRI machine. To remove noise from the recordings, we use the Adobe Express speech enhancement tool. We then use the enhanced speech samples as the target utterances when reconstructing the MRI video.

C.6.1 MRI Reconstruction with Diffusion Models

We have also tried using Diffusion Posterior Sampling (DPS) [11] for MRI reconstruction, but we have found that its performance was worse than using the GAN. The diffusion model was not able to make large geometric changes such as tongue movement. DPS and its follow-ups mainly focus on low-level inverse problems such as deblurring. Compared to a diffusion model, a GAN has a semantically meaningful latent space which allows better optimization of the vocal tract shape during speech. Solving blind inverse problems with diffusion models is still an open problem, and we believe that combining diffusion with acoustic simulation is an exciting extension of our work.

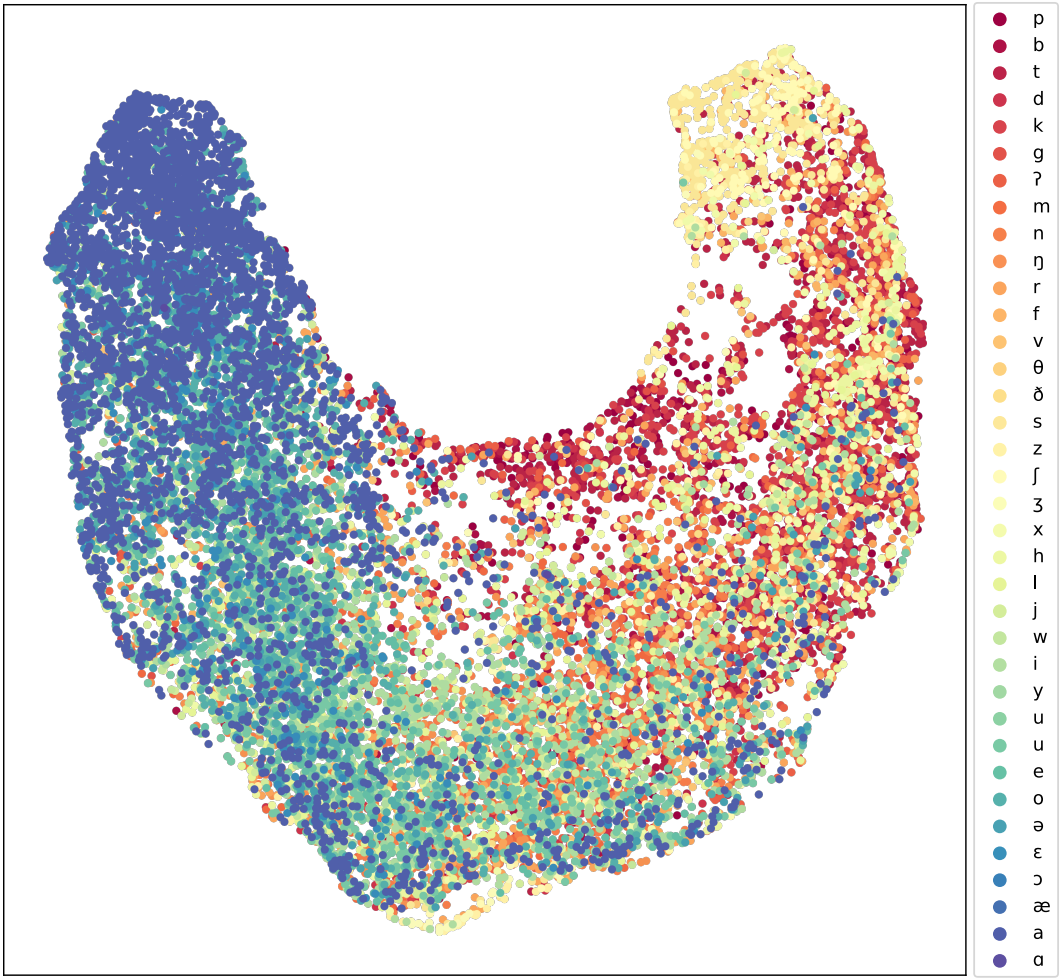


Figure 9: **A Larger UMAP Projection of Encoded Vocal Tract Area Functions.** We provide a UMAP projection of an even larger set of phonemes. The embedded area functions form a continuous space spanning the set of IPA symbols.

D Ethical Discussion

While the simulator presented in this paper is intended for realistic speech reconstruction and not speech generation, we acknowledge that generative speech models—and the tasks that they enable, such as voice cloning—have become serious ethical issues. As with other speech models, responsible use is essential. All samples produced with our simulator should be clearly labeled as such.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper provides evidence for all claims in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the paper. Assumptions are clearly stated.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our paper does not provide theoretical results. However, derivations of the equations presented in the paper are provided in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides details on how to reproduce experimental results. Additional details are added in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer:[Yes]

Justification: We will release code and data used upon publication.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters and optimizers used are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Confidence intervals are reported for numerical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We explicitly detail what GPUs we use to run experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our work relies solely on public datasets, and are properly attributed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts of our work, and have an ethical discussion in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: All models are trained on publicly available datasets. Because our models are trained specifically for vocal tract reconstruction, they are unlikely to pose a risk for applications such as deepfake generation or voice cloning.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit assets used. The IPA chart for example is shared under a CC BY-SA 4.0 license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not share new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not do research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not do research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development does not involve LLMs as a component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.