

WhisTLE: Deeply Supervised, Text-Only Domain Adaptation for Small Pretrained Speech Recognition Transformers

Akshat Pandey
Comcast
Speech AI
Washington, D.C., USA
akshat_pandey@comcast.com

Karun Kumar
Comcast
Speech AI
Washington, D.C., USA
karun_kumar@comcast

Raphael Tang
University College London
Department of Computer Science
London, UK
r.tang@cs.ucl.ac.uk

Abstract

Pretrained automatic speech recognition (ASR) models such as Whisper perform well but still need domain adaptation to handle unseen parlance. In many real-world settings, collecting speech data is impractical, necessitating text-only adaptation. We propose WhisTLE, a deeply supervised, text-only adaptation method for pretrained encoder-decoder ASR models. WhisTLE trains a variational autoencoder (VAE) to model encoder outputs from text and fine-tunes the decoder using the learned text-to-latent encoder, optionally combined with text-to-speech (TTS) adaptation. At inference, the original encoder is restored, incurring no extra runtime cost. Across four datasets and four ASR models, WhisTLE alone helps in 28 of 32 (88%) cases, with an average relative WER drop of 22%. Using WhisTLE alone or in combination with other adaptation approaches helps in 116 of 144 (81%) cases, with an average additional WER drop of 15%. Given all treatments that improve on baseline scores, WhisTLE helps in 45 of 55 (82%) cases, with an average additional relative WER drop of 16%. WhisTLE with TTS reduces word error rate (WER) by a *relative* 49% and outperforms all non-WhisTLE baselines in 100 of 112 scenarios. We also find that WhisTLE additively complements any combination of other domain adaptation approaches; we thus recommend the inclusion of WhisTLE during standard processes for adapting encoder-decoder ASR models.¹

1 Introduction

Although state-of-the-art automatic speech recognition (ASR) models such as Whisper (Radford et al., 2022) are trained on hundreds of thousands of hours of speech-text pairs, they still benefit from domain adaptation, especially if the target domain

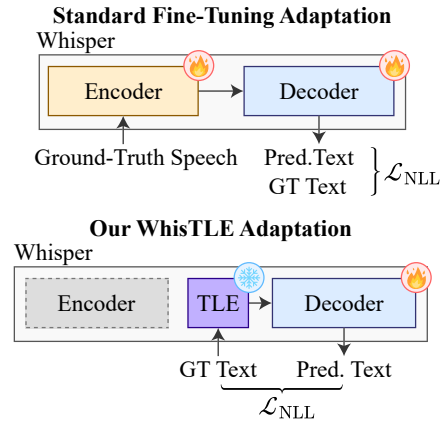


Figure 1: Our WhisTLE adaptation process. Speech-text pairs are treated as usual (top subfigure), while text data with no paired audio is passed through our frozen text-to-latent encoder (TLE), a trained variational autoencoder (VAE) instead of the Whisper encoder, before being fed to the decoder (see bottom). The TLE module is frozen at training time; everything else is optimized using the usual negative log-likelihood loss ($\mathcal{L}_{NLL}$).

contains words unseen in the source domain. Unfortunately, three constraints arise in practice: First, gathering speech in the target domain may be infeasible, either financially or otherwise, limiting us to text-only adaptation. Second, the ASR system must be small (e.g., less than 1–2 billion parameters) for efficient deployment, preventing the use of speech large language models (LLMs; Bai et al., 2024) and thus LLM text-tuning approaches (Fang et al., 2025). Lastly, pretrained ASR models preclude us from applying other model architectures that afford text-only training (Vuong et al., 2023; Ling et al., 2023; Deng and Woodland, 2023b; Deng et al., 2021), as those require training from scratch. This problem is not merely an academic artifact: in real-world deployment, users continuously evolve and speak new words and parlance, reducing ASR effectiveness, and collecting the required amount of speech for full training can be prohibitively expensive and time consuming.

¹Code: <https://github.com/akshat0123/WhisTLE>.

A standard text-only adaptation paradigm is to train an auxiliary language model (LM) in the target domain and fuse its outputs with the ASR model. Two popular approaches here are shallow fusion and deep fusion (Gulcehre et al., 2015), with the main difference being that shallow fusion linearly combines the log probabilities of the ASR model and the LM during decoding, whereas deep fusion integrates the ASR model and the LM at the hidden feature level. Another auxiliary model-based approach is internal language model estimation (Meng et al., 2020). Similar to shallow fusion, this approach also linearly combines the log probabilities of the ASR model and the LM during decoding, but in addition it also subtracts log probabilities produced by the ASR model’s internal language model, represented by the log probabilities produced by the model after zeroing out the ASR model’s acoustic representation. However, external language modeling approaches can still fail for unseen words since end-to-end ASR models do not generalize well to novel grapheme sequences.

More promisingly, recent approaches synthesize speech using text-to-speech (TTS) models, then fine-tune the ASR model (Perrin and Boulianne, 2025). Nevertheless, TTS provides only input–output supervision and does not explicitly guide how an end-to-end model’s *internal state* should adapt to a target domain. Prior work has shown deep supervision to benefit knowledge distillation (Sun et al., 2019) and image classification (Lee et al., 2015), to name a few.

In this work, we propose WhisTLE (adapting Whisper with Text-to-Latent Encodings), a deeply supervised adaptation approach for encoder–decoder ASR models, such as OpenAI’s Whisper (Radford et al., 2022) and Canary (Puvvada et al., 2024). WhisTLE runs complementary to TTS adaptation, which supervises only the input speech and the output text, by also deeply supervising the ASR model’s latent state. Specifically, we train a variational autoencoder (VAE; Kingma and Welling, 2022) to directly model ASR encoder output using text rather than audio. Then, we fine-tune the ASR decoder using our text-only encoder as a drop-in replacement for the ASR encoder, optionally mixing in TTS adaptation.

Our main contributions are as follows: First, to our knowledge, we propose the first deeply supervised, text-only domain adaptation method for pretrained encoder–decoder ASR transformers. Unlike prior work on text-only training with novel and

LLM architectures, our approach also applies in the pretrained, non-LLM setting. Second, we show that WhisTLE improves ASR quality for four state-of-the-art models on out-of-distribution domains, particularly when combined with TTS adaptation, likely due to its deep latent supervision in addition to TTS’s input–output supervision. Across four out-of-domain datasets, two in-domain datasets, and four encoder–decoder models, WhisTLE with TTS yields a 49.0% average relative WER reduction over no adaptation and outperforms all non-WhisTLE baselines in 100 out of 112 experimental scenarios. We thus recommend the inclusion of WhisTLE in standard practices for adapting ASR models to new domains.

2 Methodology

Recently, transformer-based models (Vaswani et al., 2023) have been applied to ASR with great success (Baevski et al., 2020a; Hsu et al., 2021; Gulati et al., 2020). Whisper (Radford et al., 2022) and Canary (Puvvada et al., 2024) are such state-of-the-art encoder–decoder transformers; we thus focus on them in this paper.

In the context of speech recognition, text-only domain adaptation refers to training a speech recognition model without any ground-truth audio, only text. Current approaches can be broadly split into two categories: those that modify and extend existing ASR models to handle new vocabulary (Joshi and Singh, 2022; Bataev et al., 2023; Zheng et al., 2021; Gulcehre et al., 2015; Meng et al., 2022; Li et al., 2023), and those that detail novel ASR architectures capable of text-only training (Vuong et al., 2023; Ling et al., 2023; Deng and Woodland, 2023b; Deng et al., 2021). We are solely concerned with the former since the latter additionally requires copious amounts of speech recognition training from scratch; we wish to work exclusively with existing ASR models, as that enables us to leverage large-scale pretrained models and greatly reduce resource consumption. Literature regarding text-only domain adaptation for existing ASR models can be further split into approaches that generatively model ASR input when audio is not available (Zheng et al., 2021; Bataev et al., 2023; Joshi and Singh, 2022), and approaches that integrate the use of an external language model at inference time (Gulcehre et al., 2015; Meng et al., 2022; Li et al., 2023).

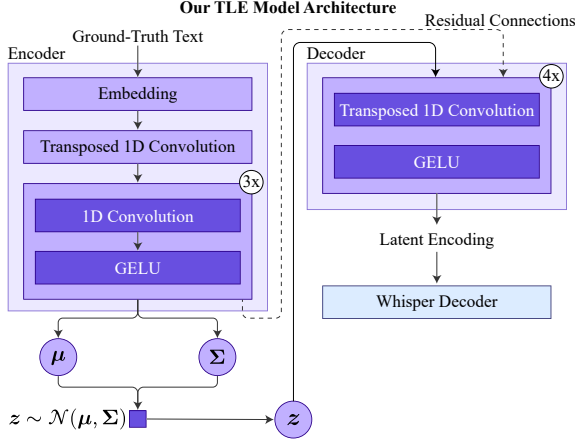


Figure 2: Model architecture of our text-to-latent encoding VAE (purple), which also follows an encoder-decoder paradigm.

2.1 Our WhisTLE Approach

The goals in our approach to text-only adaptation are as follows: (1) maximize our leveraging of the current state of the art in automatic speech recognition, (2) minimize training time and resource efforts, and (3) require no additional resources at inference time. Our WhisTLE approach differs from prior approaches making use of an external language model because it requires no additional resources at inference time. It also differs from prior generative approaches in that we train a model to synthesize transformer encoder representations given text, rather than modeling audio or Mel spectrograms directly.

Our approach rests on two hypotheses: that deep supervision of latent states helps for domain adaptation, and that modeling a transformer model’s encoder output is “simpler” (i.e., more training efficient, as we show in Section 5) than modeling speech in general. In line with the information bottleneck hypothesis (Tishby et al., 2000), this is likely due to encoder representations capturing less information than speech, as they focus on what is necessary for speech recognition.

Concretely, we train a variational autoencoder (VAE; Kingma and Welling, 2022) to precisely model Whisper (Radford et al., 2022) encoder outputs given text input. After the text adaptation VAE is trained, the Whisper decoder can be fine-tuned on text without corresponding audio by using the VAE output rather than the Whisper encoder output during training. At inference time, Whisper decoding proceeds the same as it was prior to text-only adaptation, without the use of any additional run-

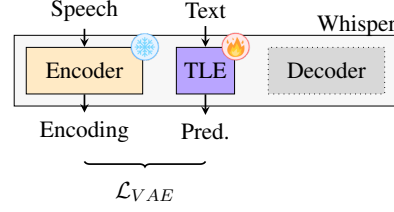


Figure 3: Our WhisTLE adapter training process. Speech-text pairs are fed into the Whisper encoder and the TLE module, the former receiving audio and the latter receiving text. Given the ground-truth Whisper latent encoding and the predicted TLE encoding, we optimize the VAE loss.

time or memory: we *discard* the training-only TLE module. Figure 1 displays the pipeline for training Whisper using the text-only encoder.

Formally, let $x \in \mathbb{R}^n$ be speech audio waveform in the target domain and the string $y \in \Sigma^\ell$ be the tokenized transcript (of ℓ length) of x , where Σ is the vocabulary. During full Whisper training, audio input is encoded using the Whisper encoder $f_\theta : \mathbb{R}^n \mapsto \mathbb{R}^{[n/k] \times h}$, where k is the downsampling factor and h is the number of hidden dimensions. The encodings are then fed to the decoder $g_\theta : \mathbb{R}^{[n/k] \times h} \mapsto \mathbb{P}(\Sigma^\ell)$, which outputs a differentiable probability distribution over strings of length ℓ . We optimize the network’s parameters θ end-to-end using gradient descent on backpropagating the negative log-likelihood loss, e.g.,

$$\theta^* := \underset{\theta}{\operatorname{argmin}} \mathcal{L}_{\text{NLL}},$$

$$\mathcal{L}_{\text{NLL}} := - \sum_{i=1}^{\ell} \log g_\theta(f_\theta(x))[y_i],$$

where $[y_i]$ extracts the output probability associated with token y_i at the i^{th} timestep.

During text-only adaptation with WhisTLE, we instead replace f_θ with a frozen *text-based* encoder $f_\phi^{\text{TLE}} : \Sigma^\ell \mapsto \mathbb{R}^{[n/k] \times h}$. The Whisper encoder is set aside, and we pass only the tokens y to f_ϕ^{TLE} to produce an approximation of the Whisper encoder output $f_\theta(x)$. We can then feed this approximation to the decoder for training and proceed with log-likelihood optimization as usual:

$$\theta_{\text{TLE}}^* := \underset{\theta}{\operatorname{argmin}} \tilde{\mathcal{L}}_{\text{NLL}},$$

$$\tilde{\mathcal{L}}_{\text{NLL}} := - \sum_{i=1}^{\ell} \log g_\theta(f_\phi^{\text{TLE}}(y))[y_i].$$

Note the lack of dependence on x . After text-only training, we abandon f_ϕ^{TLE} and use Whisper in the standard fashion.

Figure 2 displays the architecture of our proposed text-to-latent encoder (TLE). The model is a convolutional VAE with three convolutional encoder layers and four convolutional decoder layers, with residual connections between the respective encoder and decoder layers. Text is embedded and upsampled using a transposed convolutional layer prior to entering convolutional encoder layers.

Figure 3 displays the training process of the TLE module. We train our VAE on speech and text in the *source* domain, $\mathbf{x}_s$ and $\mathbf{y}_s$. We use Whisper encoder outputs as ground truth and optimize the usual VAE loss function with beta regularization:

$$\begin{aligned} \mathcal{L}_{\text{VAE}} := & \mathbb{E} \|f_{\theta}(\mathbf{x}) - f_{\phi}^{\text{TLE}}(\mathbf{y})\|_2^2 \\ & + \lambda [\mathbb{E} \|f_{\theta}(\mathbf{x}_{\text{unmasked}}) - f_{\phi}^{\text{TLE}}(\mathbf{y})\|_2^2] \\ & + \beta \text{KL}(\mathbb{P}_{\phi}(\mathbf{z}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})), \end{aligned}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the first term is the mean-squared reconstruction loss on masked audio, the second term is the mean-squared reconstruction loss on unmasked audio, scaled by λ , and the third term controls how far the posterior deviates from the isotropic unit Gaussian prior, scaling with β . Our TLE VAE f_{ϕ}^{TLE} is trained to minimize this objective, e.g., $\text{argmin}_{\phi} \mathcal{L}_{\text{VAE}}$.

We split the reconstruction error function of the standard variational autoencoder loss function into masked and unmasked mean-squared errors to account for the fact that the Whisper encoder does not use masking during training. The model must be able to learn on its own where in the original Whisper encoder output reflects input padding, and simultaneously produce accurate reconstructions of the portion of encoder output that reflects speech.

3 Experimental Setup

Datasets. We use six total datasets in our testing, two of which we consider “in domain” and four of which we consider “out of domain.” For the in-domain datasets, we use CommonVoice (Ardila et al., 2020) (~ 1.1 million audio clips) and LibriSpeech (Panayotov et al., 2015) ($\sim 100\text{k}$ clips). For the out-of-domain datasets, we use EMNS (Noriy et al., 2023) ($\sim 1.1\text{k}$ clips), EmoV-DB (Adigwe et al., 2018) ($\sim 17\text{k}$ clips), the Free ST American English corpus (ST-AEDS) (Ltd., 2018) ($\sim 6.8\text{k}$ clips), and the Open-source Multi-speaker Corpora of the English Accents in the British Isles (EABI; Demirsahin et al., 2020) (3.8k).

TLE	loe bush whipped him in the face and left no sting
TTS	Lowe Bush whipped him in the face and left no sting
TLE+TTS	Low bush whipped him in the face and left no sting
TLE	he was defeated by derban in the november election
TTS	he was defeated by derbyn in the november election
TLE+TTS	he was defeated by Durbin in the november election
TLE	anglican chant as a method of singing prose versions of the sums
TTS	Anglican chant is a method of singing prose versions of the sums
TLE+TTS	Anglican chant is a method of singing prose versions of the Psalms
TLE	phillips stood undecided his ears strained to catch the lightest sound
TTS	philip stood undecided his ears strained to catch the lightest sound
TLE+TTS	philip stood undecided his ears strained to catch the slightest sound

Table 1: Some example differences between the top two single treatments (TLE, TTS) and their combination (TLE+TTS); improved handling of homonyms and named entity recognition is evident.

EMNS (Noriy et al., 2023) is a dataset of strictly British-English acted speech focused on providing training data for storytelling, with a focus on emotional intensity. Similarly, EmoV-DB (Adigwe et al., 2018) also focuses on emotional speech. ST-AEDS (Adigwe et al., 2018) is an American English based dataset of audio transcriptions recorded on a cellphone. EABI (Demirsahin et al., 2020) is a dataset focused on providing transcriptions from a wide breadth of English accents from the British Isles. These choices reflect the practical setting where an end-to-end ASR model pretrained in a domain with large-scale data (e.g., Common-Voice) is adapted to a target domain with smaller resources. In the case the target “out of domain” datasets diverge from the “in domain” datasets with regard to emotional intensity, accent, and recording environment, depending on the dataset.

We assess our WhisTLE as follows: (1) fine-tune Whisper on an in-domain dataset using the standard approach; (2) train our VAE on the same in-domain dataset; (3) train Whisper using text-only training on an out-of-domain dataset; then (4) test Whisper on both the audio and text pairs of the out-of-domain dataset. As baselines, we compare WhisTLE against TTS-generated data, shallow fusion, deep fusion, and internal language model estimation at inference time. We also experiment with combining multiple adaptation approaches, as they are not mutually exclusive with WhisTLE.

Hyperparameters. For in-domain fine-tuning, we train Whisper for 100K steps with a batch size of 8. For text-only training on out-of-domain datasets (e.g., WhisTLE and TTS adaptation), we train for 50K steps with a batch size of 8. For every step of text-only training, we also train on the in-domain dataset for two steps using the audio and text, in order to prevent the model from catastrophically forgetting in-domain data.

		Method	Out-of-Domain Dataset			
			EMNS	EmoV-DB	ST-AEDS	EABI
In-Domain Dataset	CommonVoice	None	14.6	11.6	6.4	9.5
		TLE	9.5	11.8	3.8	8.5
		TTS	11.8	7.9	2.8	7.1
		SF	10.1	10.5	6.1	11.3
		DF	17.0	17.4	14.2	22.6
		ILME	9.6	10.1	5.1	4.2
		TLE+SF	12.4	8.4	5.3	6.9
		TTS+SF	8.1	9.0	3.2	4.7
		TLE+DF	16.8	17.2	11.9	21.2
		TTS+DF	18.3	12.7	8.2	17.1
		TLE+ILME	5.7	5.8	4.2	6.6
		TTS+ILME	5.2	6.3	2.9	2.5
		TLE+TTS	7.2	6.9	2.5	6.1
		TLE+TTS+SF	6.6	8.2	3.4	3.3
		TLE+TTS+DF	14.0	10.6	8.1	16.6
		TLE+TTS+ILME	4.9	4.0	2.8	2.4
	LibriSpeech	None	11.8	26.4	7.1	9.4
		TLE	11.3	20.2	6.3	8.6
		TTS	10.8	17.5	4.1	7.8
		SF	84.5	69.8	52.1	32.9
		DF	72.2	58.6	49.2	40.0
		ILME	34.3	28.1	15.3	19.4
		TLE+SF	27.3	54.5	30.9	13.1
		TTS+SF	41.6	61.4	52.1	53.8
		TLE+DF	32.1	65.1	41.7	36.0
		TTS+DF	31.6	44.6	27.2	25.4
		TLE+ILME	25.6	28.0	16.1	11.8
		TTS+ILME	23.1	26.9	15.4	8.2
		TLE+TTS	8.7	10.1	4.2	7.1
		TLE+TTS+SF	26.3	84.7	72.7	22.0
		TLE+TTS+DF	24.3	58.6	27.9	27.7
		TLE+TTS+ILME	17.0	39.7	16.1	8.2
	Standard fine-tuning	1.0	1.2	1.0	2.1	

Table 2: Main results for Whisper-Large.

		Method	Out-of-Domain Dataset			
			EMNS	EmoV-DB	ST-AEDS	EABI
In-Domain Dataset	Common Voice	None	16.7	23.1	13.7	12.5
		TLE	9.3	13.7	6.2	7.5
		TTS	5.6	13.5	4.3	4.8
		SF	15.2	23.7	17.3	17.9
		DF	19.7	33.4	22.2	39.9
		ILME	12.1	12.7	9.9	10.1
		TLE +SF	11.7	17.8	10.0	11.5
		TTS+SF	8.8	11.8	10.9	5.9
		TLE +DF	16.1	20.0	18.0	22.8
		TTS+DF	14.5	25.2	13.4	15.6
		TLE +ILME	7.7	7.4	8.9	11.5
		TTS+ILME	5.8	10.8	8.7	3.4
		TLE +TTS	6.1	8.4	2.3	4.0
		TLE +TTS+SF	6.1	6.4	6.2	3.9
		TLE +TTS+DF	14.3	16.8	12.0	13.3
		TLE +TTS+ILME	5.5	4.6	5.2	3.5
	LibriSpeech	None	8.7	9.6	3.1	6.4
		TLE	8.3	7.5	3.0	5.9
		TTS	6.0	5.5	2.1	4.2
		SF	17.6	18.4	9.6	4.9
		DF	21.8	24.1	10.4	6.9
		ILME	17.2	19.0	9.6	3.4
		TLE +SF	16.9	14.0	11.3	3.2
		TTS+SF	7.4	9.7	3.6	2.5
		TLE +DF	21.4	22.8	12.1	6.9
		TTS+DF	19.0	20.4	10.2	5.3
		TLE +ILME	16.6	17.6	8.5	3.2
		TTS+ILME	6.8	8.6	3.6	2.0
		TLE +TTS	5.3	4.8	2.0	3.9
		TLE +TTS+SF	6.6	8.5	5.5	2.2
		TLE +TTS+DF	18.9	21.3	10.4	5.5
		TLE +TTS+ILME	6.3	7.6	4.2	2.2
	Standard fine-tuning	1.1	1.2	1.0	2.1	

Table 3: Main results for Whisper-Medium.

For shallow fusion (SF; [Gulcehre et al., 2015](#)), we train a trigram language model on the out-of-domain datasets and linearly combine it (scaled by weight γ) with the in-domain ASR model. For all SF experiments, we search over γ values of 0.10, 0.25, 0.50, and 0.75, and report the best result of the set. For deep fusion ([Gulcehre et al., 2015](#)) (DF), we use the same trigram language model, but train a bottleneck neural network that projects the output of the last hidden Whisper decoder layer into the same space as the language model output. The trained output of this neural network is then used as a per-token scaling factor for the output of the language model during inference. For experiments where CommonVoice is the “in domain” dataset, CommonVoice data is used to train the bottleneck network for 10k steps. For all LibriSpeech “in domain” experiments, LibriSpeech data is used to train the bottleneck network.

For internal language model estimation (ILME; [Meng et al., 2020](#)), we again use the same trigram language model, but during decoding we add output from the language model (scaled by γ), and subtract the output from the internal language model of the ASR model (scaled by α). Internal language

model output is calculated by submitting silence to the Whisper feature extractor, encoding it, and submitting this to the decoder to get the “internal language model” probabilities. The γ value is taken from the best performing shallow fusion model for each “in domain”, “out of domain” pair, and we search over α values of 0.10, 0.25, 0.50, and 0.75, and report the best result of the set.

We use the FastSpeech2 ([Ren et al., 2022](#)) and SpeechT5 ([Ao et al., 2022](#)) models trained on CommonVoice and LibriSpeech, respectively, for the TTS adaptation experiments. For all TTS experiments, we generate audio for out-of-domain datasets using randomly selected audio from the in-domain dataset for speaker simulation during generation. We then fine-tune the base model on the generated audio after fine-tuning on the in-domain audio. When combining TTS and TLE, TLE training is also performed on the same data as TTS training. This setup aims to evaluate whether synthetic speech provides complementary supervision for improving robustness under domain shift across diverse out-of-domain evaluation conditions. Further details for our training settings can be found in Section A of the appendix.

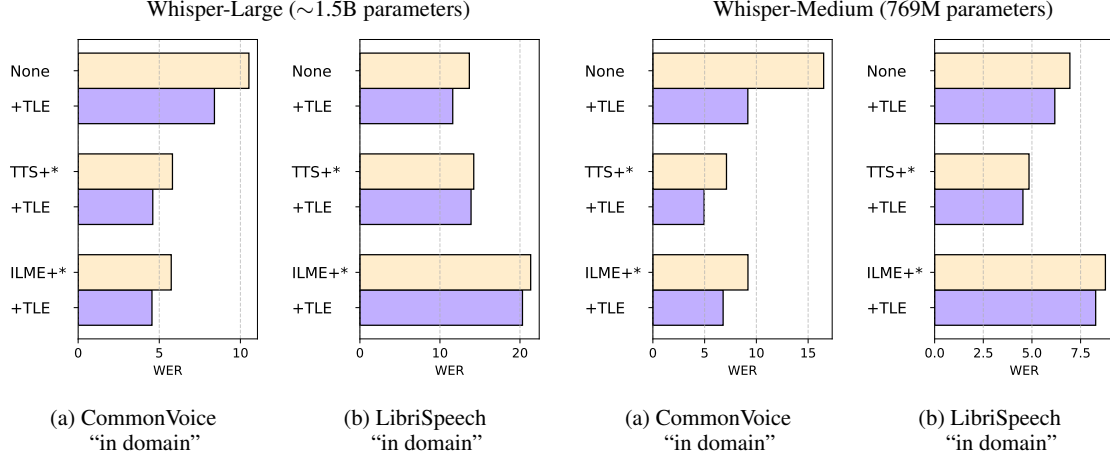


Figure 4: The results of adding TLE as a treatment in combination with other treatments; lower WERs are better. The designation “TTS+*” implies all treatment combinations excluding TLE, SF, and DF (i.e., TTS, TTS+ILME). The “+TLE” bars designate the average change in word error rate across all “out of domain” datasets when adding TLE to an existing treatment (i.e., TTS+TLE, TTS+ILME+TLE). Results are statistically significant at the 95% level according to the paired bootstrap.

4 Results and Discussion

4.1 Main Whisper Results

Tables 2 and 3 display the changes in word error rates (WER) on out-of-domain datasets for our WhisTLE approach (TLE for short), training using TTS-generated data (TTS) and deep fusion (DF), inference using shallow fusion (SF) and internal language model estimation (ILME), as well as combining multiple approaches.

Overall, combining TLE and TTS attains the best average WER of 5.6 across the treatments—supporting our hypothesis that the model benefits from both input–output and latent supervision—followed by TTS at 7.2, TLE + TTS + ILME 8.4, TTS + ILME and TLE respectively at 8.8, no adaptation at 11.9, and the rest higher than 15. Although TTS outperforms TLE, combining TLE and TTS yields an average WER improvement (1.64 points) roughly equal to the original performance gap between TTS and TLE alone (1.60), showing a strong compounding effect. TLE outperforms shallow fusion in 14 out of 16 occurrences, deep fusion in 16 out of 16 occurrences, and internal language model estimation in 12 out of 16 occurrences. Adding TLE in any combination (e.g., TLE+TTS vs. TTS) helps in 69 out of 80 (86%) scenarios, resulting in an average relative WER drop of 22%. These differences are statistically significant at the 95% level according to the paired bootstrap.

Dataset-wise, the largest absolute gains from TLE+TTS appear on ST-AEDS and EMNS, with

average drops of 3–4 WER compared to either TTS or TLE alone. Improvements are somewhat smaller on EABI (2 points) and EmoV-DB (1.5 points) but remain consistent in direction. Model-wise, Whisper-medium consistently benefits more from TLE+TTS, with average WERs below 6.0 on all out-of-domain sets and showing relative drops exceeding 25% compared to no adaptation. Whisper-large shows strong gains from TLE+TTS on CommonVoice (50% relative drop vs. None), but its LibriSpeech results are noisier, partly due to SF catastrophically raising WER. Across both models, the direction of improvement is robust, confirming that the compounding benefit of TLE and TTS is not model specific.

We also confirm that TLE additively improves any existing combination of adaptation approaches. As shown in Figure 4, including TLE improves the WER on average for all in-domain dataset and method configurations, averaged across the out-of-domain datasets, with SF and DF excluded due to their quality degradation. Adding TLE to no domain adaptation (first row) yields the largest relative improvement of 23%, followed by any combination with TTS (15%) and any with ILME (14%).

In total, TLE (both alone and as part of a combination of approaches) reduces WER in 116 of 144 (81%) cases. Out of the 28 cases where it does not, 15 (54%) performed worse than the baseline prior to the addition of TLE. Of these 15 cases, all include the use of an external language modeling approach. Compared to TLE’s reduction of WER

in 81% of cases, SF only reduces WER in 27% of cases, DF none of the cases, and ILME 44% of cases. On average, SF worsens WER by a relative 76%, DF by 114%, and ILME by 0%. Given this (along with the results of our linear-mixed effects analysis), we argue that the true underlying cause of the degradation in these 15 cases are the external language modeling methods and not TLE. We discuss the problems with external language modeling approaches in the context of our experiments further in Section 4.2.

4.2 External Language Model-Based Treatments

Internal language model estimation (ILME) reduces average WER on out-of-domain datasets by 4.3 when the base ASR model is trained on CommonVoice. When the base model is trained on LibriSpeech however, ILME increases WER by an average of 8.0 points. The LibriSpeech (Panayotov et al., 2015) corpus is a curated dataset of derived of audiobook narrations with relative stability in audio recording quality compared to CommonVoice (Ardila et al., 2020), which is a crowd-sourced dataset of recordings of Wikipedia text with a large variety of accents as well as variability in audio quality and formatting. ILME aims to alleviate domain mismatch by removing the bias induced by an ASR model’s internal language model. The difference in ILME’s performance on the CommonVoice and LibriSpeech trained base models suggests that a base model trained on a dataset with more robust diversity of audio sources and quality may benefit more from an adjustment strictly limited to the ASR model’s internal language model representation.

TLE- and TTS-based approaches, while less performant on LibriSpeech (TLE has an average WER drop of 1.4, and TTS 3.1), than CommonVoice (TLE has an average WER drop of 4.7, and TTS 6.3), do not suffer from the same problems as ILME in our experiments. The TTS approach directly models audio, and TLE models the ASR model’s internal audio representation, and in our experiments both are more robust to changes in audio quality in out-of-domain testing data than any external modeling based approaches we tested.

Shallow fusion and deep fusion performed the poorest in our experiments, with an increase in WER rates of 13.2 and 17.4 respectively when compared to the base trained model. The main cause of shallow fusion’s poor performance in our exper-

Whisper-Large			
Parameter	Coef.	Std. Err.	<i>p</i>
TLE	-0.057	0.027	0.036
ILME	0.072	0.023	0.002
TTS	-0.059	0.019	0.002
TLE×ILME	-0.007	0.033	0.822
TLE×TTS	0.032	0.027	0.229
Whisper-Medium			
Parameter	Coef.	Std. Err.	<i>p</i>
TLE	-0.034	0.011	0.002
ILME	0.004	0.009	0.636
TTS	-0.057	0.008	0.001
TLE×ILME	0.022	0.013	0.096
TLE×TTS	0.005	0.012	0.659

Table 4: Linear mixed-effects model estimation, reported using treatment coding.

iments was the dramatic increase of hallucinations produced by linearly combining the outputs of the external language model without accounting for the bias of the base model’s internal language model representation (as ILME does). Deep fusion failed largely due to the mismatch of training audio used to train the bottleneck neural network fusing the ASR and LM outputs. Because we are modeling a scenario where we have no access to the out-of-domain audio, the fusion network was trained on in-domain audio, which proved to be insufficient training for successfully integrating the external LM when met with out of domain audio.

4.3 Linear Mixed-Effects Analysis

To evaluate the effectiveness of TLE across in-domain and out-of-domain datasets and to determine whether the improvement is statistically significant regardless of the effects of dataset or compounding text-only adaptation approaches, we use a linear mixed-effects model as follows:

$$\begin{aligned} \text{WER}_{ijk} = & \beta_0 + \beta_1(\text{TLE}) + \beta_2(\text{ILME}) + \beta_3(\text{TTS}) \\ & + \beta_4(\text{TLE} \times \text{ILME}) + \beta_5(\text{TLE} \times \text{TTS}) \\ & + \beta_6(\text{Test Dataset}) + u_i + v_j + \epsilon_{ij}, \end{aligned}$$

where β_0 is the baseline WER, β_1 to β_3 are the fixed effects for TLE, TTS and ILME, β_4 and β_5 are the interactions between TLE and ILME and TLE and TTS, u_i a random intercept for the audio sample ID, v_j a random intercept for the in-domain datasets, and ϵ_{ij} the remaining residual error; all random effects are Gaussian. Since SF and DF are prone to frequent hallucination, we omit them from these tests. This model totals 10 free parameters, estimated on 500 speech–text examples from each out-of-domain dataset.

As reported in Table 4, the results of the linear mixed-effects model test show that in the case of

		Model	Out-of-Domain Dataset			
			EMNS	EmoV-DB	ST-AEDS	EABI
In-Domain	CV	Standard	7.8	18.4	3.7	8.6
		TLE	7.4	16.8	4.5	8.1
	LS	Standard	12.2	31.0	5.6	7.8
		TLE	11.0	24.5	5.5	6.4

Table 5: Results for Canary-1B.

		Model	Out-of-Domain Dataset			
			EMNS	EmoV-DB	ST-AEDS	EABI
In-Domain	CV	Standard	10.4	10.3	4.9	9.4
		TLE	10.6	10.6	4.7	7.0
	LS	Standard	20.5	35.9	23.5	30.4
		TLE	9.4	9.3	3.5	9.3

Table 6: Results for Canary-180M-flash.

both Whisper-large and Whisper-medium, TLE reduces the WER (-5.7 and -3.4 points respectively) to a significant degree ($p < 0.05$). None of the interaction terms for either model are statistically significant, demonstrating the effectiveness of TLE regardless of the presence of other text adaptation approaches.

4.4 Auxiliary Canary Experiments

To assess generalization to other encoder-decoder speech recognition models, we run experiments for Canary-1B (Puvvada et al., 2024) and Canary-180M-flash (Želasko et al., 2025). Canary-1B uses a Conformer-based encoder (Gulati et al., 2020) and a standard transformer decoder, while Canary-180M-flash is a smaller variant of Canary-1B with faster training and inference speeds. We choose the Canary model family for our experiments due to their comparable use of an encoder-decoder architecture, allowing for direct application of our text-only training approach. The only adjustment made for WhisTLE is adding an additional linear layer at the end of our VAE, as Canary, unlike Whisper, also requires encoding lengths to be input into its decoder, so our VAE must also produce encoding lengths. This is not required for Whisper (Radford et al., 2022), which inherently forces encodings to account for a fixed length of thirty seconds.

Tables 5 and 6 report WhisTLE results for Canary-1B and Canary-180M-flash on four out-of-domain datasets. With LibriSpeech as the in-domain corpus, Canary-1B achieves a 16.3% relative WER reduction and Canary-180M-flash a substantial 71%, improving on all datasets. These trends align with our Whisper results, where TLE consistently lowers WER. Using CommonVoice

yields smaller gains, with 4.4% for Canary-1B and 6.0% for Canary-180M-flash, like Whisper’s more modest improvements on datasets such as EmoV-DB. Overall, these findings suggest that text-only latent supervision generalizes beyond Whisper to other encoder-decoder ASR models.

5 Related and Future Work

Past generative approaches include using synthetic text-to-speech (TTS) systems to generate out-of-distribution audio (Zheng et al., 2021) and generative adversarial networks to synthesize Mel spectrograms (Bataev et al., 2023). Zheng et al. (2021) rely on a closed-source paid TTS system, so model details are unavailable. Bataev et al. (2023) uses 130M parameters for spectrogram generation and audio synthesis. Our FastSpeech2 (Ren et al., 2022) implementation has 46M parameters and SpeechT5 (Ao et al., 2022) has 144M, both using the HiFi-GAN vocoder. Our TLE approach uses 91M parameters for Whisper-medium and 104M for Whisper-large, which is similar in size to these TTS systems but much faster to train. FastSpeech2 trains on batches of 48 sentences for 160K steps, while SpeechT5 jointly pretrains on text and audio with 12K-token batches before TTS fine-tuning. Both TTS papers above also do not include the additional training required for HiFi-GAN. In contrast, our TLE trains for only 100K steps on batches of four sentences.

A future research direction is to characterize the Pareto curve of efficiency vs. effectiveness for domain adaptation methods. Another is to quantify which adaptation methods may be best either alone or in combination prior to running the methods to evaluate. Table 1 displays examples of changes in output when combining TLE and TTS treatments. Anecdotally, we find a combination of these approaches to lead to better handling of homonyms and named entities, but a more detailed study is needed for the exact effects of each treatment and when to use one over another. We also feel it would be a worthwhile study to examine the changes in latent space encodings when using audio and the original model encoder versus the predicted encoding generated by the text-only adaptation module.

Using separate language models in ASR dates back to hidden Markov models and remains widely used. In end-to-end ASR, shallow fusion (Gulcehre et al., 2015) combines model output scores with an external LM during inference. Meng et al.

(2022) approximates internal LM scores in end-to-end ASR for better external LM integration; Li et al. (2023) use reranking or deep fusion with LLaMA to handle out-of-domain vocabulary. Recent speech LLM approaches directly integrate an LLM with a speech encoder (Bai et al., 2024), which allows text-only domain adaptation (Fang et al., 2025). Unfortunately, the multibillion-scale nature of speech LLMs hinders our large-scale deployment to millions of end users and hence motivates the need for the present WhisTLE work. We also did not consider the use of large-language model-based approaches in our external language-modeling based experiments primarily because of the overhead they introduce at inference time (both in memory and runtime). In contrast, both TLE and TTS-based approaches introduce no additional inference-time overhead, and while n -gram models do introduce some overhead, it is minimal when compared to the prospect of loading an LLM for use as an external language model during inference.

Finally, architecture-based approaches introduce new models to jointly learn from audio and text. Deng and Woodland (2023a) also modify Wav2vec2 (Baevski et al., 2020a) and use an LLM-based decoder to reduce reliance on acoustic representations. Hybrid Autoregressive Transducer (Meng et al., 2023) proposes a modular model with an internal LM trainable on text only. Sato et al. (2022) add an adaptation branch embedding acoustic and linguistic features in the same space, enabling unpaired text training. In the future, we plan to generalize our approach beyond speech recognition to any encoder-decoder task.

6 Conclusions

We introduced WhisTLE, a deeply supervised, text-only adaptation method for pretrained encoder-decoder ASR. By modeling encoder outputs from text and fine-tuning the decoder, WhisTLE especially complements the input-output supervision of TTS adaptation. Across our experiments, WhisTLE with TTS reduced WER by a *relative* 49.0% on average and outperformed all non-WhisTLE baselines in most scenarios. We find that WhisTLE improves WER both on its own and in combination with other adaptation approaches. These results demonstrate the efficacy of deep latent supervision for text-only adaptation; we recommend WhisTLE as a complementary approach for leveraging unpaired text in ASR domain adaptation.

Limitations

While we have shown that the TLE approach is a significant advancement in text-only adaptation of speech recognition models, several limitations exist. Firstly, because the method relies upon decoder training, the use of this approach requires a sufficiently large decoder to work effectively. Models that do not have a traditional decoder, such as those that use CTC decoding (Graves et al., 2006) like wav2vec 2.0 (Baevski et al., 2020b) may not be good candidates for this approach. Additionally, while we test our approach on multiple out-of-domain datasets, all of them are based in English and are focused on English speakers in the US and UK, as are our in-domain datasets. Further testing would be required to establish the effectiveness of this approach on accents that diverge further from US- and UK-based English.

In direct comparison with TTS-based approaches, our method also requires training a TLE adapter specifically for a particular ASR system, whereas many trained TTS models are already available for use publicly. Also since TLE training is performed jointly with in-domain audio-text pairs and out-of-domain text data (as having in-domain data is a standard premise in domain adaptation), use of this approach does require access to some *in-domain* audio-text data at training time. Finally, due to the constraint of minimizing inference speed and cost, we limited our external language model testing to the use of an n -gram language model. It is possible that using a billion-scale contemporary language model would bolster the results of using deep fusion, shallow fusion, as well as internal language model estimation in study.

References

- A. Adigwe, N. Tits, K. El Haddad, and 1 others. 2018. [The emotional voices database: Towards controlling the emotion dimension in voice generation systems](#). *Preprint*, arXiv:1806.09514.
- J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, and 1 others. 2022. [SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing](#). *Preprint*, arXiv:2110.07205.
- R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, and 1 others. 2020. [Common Voice: A massively-multilingual speech corpus](#). *Preprint*, arXiv:1912.06670.
- A. Baevski, H. Zhou, A. Mohamed, and M. Auli. 2020a. [wav2vec 2.0: A framework for self-supervised](#)

- learning of speech representations. *Preprint*, arXiv:2006.11477.
- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. [wav2vec 2.0: A framework for self-supervised learning of speech representations](#). *Preprint*, arXiv:2006.11477.
- Ye Bai, Jingping Chen, Jitong Chen, Wei Chen, and 1 others. 2024. Seed-ASR: Understanding diverse speech and contexts with LLM-based speech recognition.
- V. Bataev, R. Korostik, E. Shabalin, and 1 others. 2023. [Text-only domain adaptation for end-to-end ASR using integrated text-to-Mel-spectrogram generator](#). *Preprint*, arXiv:2302.14036.
- I. Demirsahin, O. Kjartansson, A. Gutkin, and C. Rivera. 2020. [Open-source Multi-speaker Corpora of the English Accents in the British Isles](#). In *LREC*.
- K. Deng, S. Cao, Y. Zhang, and L. Ma. 2021. [Improving hybrid CTC/attention end-to-end speech recognition with pretrained acoustic and language model](#). *Preprint*, arXiv:2112.07254.
- K. Deng and P.C. Woodland. 2023a. [Adaptable end-to-end ASR models using replaceable internal LMs and residual softmax](#). *Preprint*, arXiv:2302.08579.
- K. Deng and P.C. Woodland. 2023b. [Decoupled structure for improved adaptability of end-to-end models](#). *Preprint*, arXiv:2308.13345.
- Yangui Fang, Jing Peng, Xu Li, Yu Xi, and 1 others. 2025. Low-resource domain adaptation for speech LLMs via text-only fine-tuning.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. [Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks](#). volume 2006, pages 369–376.
- A. Gulati, J. Qin, C. Chiu, N. Parmar, Y., J. Yu, W. Han, and 1 others. 2020. [Conformer: Convolution-augmented transformer for speech recognition](#). *Preprint*, arXiv:2005.08100.
- C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, and 1 others. 2015. [On using monolingual corpora in neural machine translation](#). *Preprint*, arXiv:1503.03535.
- W. Hsu, B. Bolte, Y.H. Tsai, K. Lakhotia, and 1 others. 2021. [HuBERT: Self-supervised speech representation learning by masked prediction of hidden units](#). *Preprint*, arXiv:2106.07447.
- R. Joshi and A. Singh. 2022. [A simple baseline for domain adaptation in end to end ASR systems using synthetic data](#). In *ECNLP*.
- D.P. Kingma and M. Welling. 2022. [Auto-encoding variational bayes](#). *Preprint*, arXiv:1312.6114.
- C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. 2015. Deeply-supervised nets. In *AISTATS*.
- Y. Li, Y. Wu, J. Li, and S. Liu. 2023. [Prompting large language models for zero-shot domain adaptation in speech recognition](#). *Preprint*, arXiv:2306.16007.
- S. Ling, G. Ye, R. Zhao, and Y. Gong. 2023. [Hybrid attention-based encoder-decoder model for efficient language model adaptation](#). *Preprint*, arXiv:2309.07369.
- Surfing Technology Ltd. 2018. St-aeds-20180100 1, free st american english corpus.
- Z. Meng, T. Chen, R. Prabhavalkar, Y. Zhang, G. Wang, K. Audhkhasi, J. Emond, T. Strohman, B. Ramabhadran, W.R. Huang, E. Variani, Y. Huang, , and P.J. Moreno. 2023. [Modular hybrid autoregressive transducer](#). *Preprint*, arXiv:2210.17049.
- Z. Meng, Y. Gaur, N. Kanda, J. Li, X. Chen, Y. Wu, and Y. Gong. 2022. [Internal language model adaptation with text-only data for end-to-end speech recognition](#). *Preprint*, arXiv:2110.05354.
- Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong. 2020. [Internal language model estimation for domain-adaptive end-to-end speech recognition](#). *Preprint*, arXiv:2011.01991.
- K.A. Noriy, X. Yang, and J.J. Zhang. 2023. [EMNS /imz/ corpus: An emotive single-speaker dataset for narrative storytelling in games, television and graphic novels](#). *Preprint*, arXiv:2305.13137.
- V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. [Librispeech: An ASR corpus based on public domain audio books](#). In *ICASSP*.
- Y. Perrin and G. Boulianne. 2025. [Towards improved speech recognition through optimized synthetic data generation](#). *Preprint*, arXiv:2508.21631.
- K.C. Puvvada, P. Żelasko, H., O. Hrinchuk, N. Rao Koluguri, and 1 others. 2024. [Less is more: Accurate speech recognition and translation without web-scale data](#). *Preprint*, arXiv:2406.19674.
- A. Radford, J.W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. 2022. [Robust speech recognition via large-scale weak supervision](#). *Preprint*, arXiv:2212.04356.
- Y. Ren, C. Hu, X., T. Qin, S. Zhao, Z. Zhao, and T. Liu. 2022. [FastSpeech 2: Fast and high-quality end-to-end text to speech](#). *Preprint*, arXiv:2006.04558.
- H. Sato, T. Komori, T. Mishima, Y. Kawai, T. Mochizuki, S. Sato, and T. Ogawa. 2022. [Text-Only Domain Adaptation Based on Intermediate CTC](#). In *Interspeech 2022*.
- S. Sun, Y. Cheng, Z. Gan, and J. Liu. 2019. Patient knowledge distillation for BERT model compression.

- N. Tishby, F.C. Pereira, and W. Bialek. 2000. [The information bottleneck method](#). *Preprint*, arXiv:physics/0004057.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, and 1 others. 2023. [Attention is all you need](#). *Preprint*, arXiv:1706.03762.
- T. Vuong, K. Mundnich, D. Bekal, V. Elluru, and 1 others. 2023. AdaBERT-CTC: Leveraging BERT-CTC for text-only domain adaptation in ASR. In *EMNLP*.
- X. Zheng, Y. Liu, D. Gunceler, , and D. Willett. 2021. [Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems](#). *Preprint*, arXiv:2011.11564.
- P. Żelasko, K. Dhawan, D. Galvez, K.C. Puvvada, A. Pasad, and 1 others. 2025. [Training and inference efficiency of encoder-decoder speech models](#). *Preprint*, arXiv:2503.05931.

A Training Details

For all whisper-large experiments, we used a batch size of 2 with 4 gradient accumulation steps (for an equivalent batch size of 8), with a learning rate of $6.25e-8$ for a total of 100k steps. For all whisper-medium experiments, we used a batch size of 4 with 2 gradient accumulation steps (for an equivalent batch size of 8), with a learning rate of $1.25e-7$ for a total of 100k steps. For the whisper-large TLE adapter, we trained it with batch size 4 with a learning rate of $1.25e-5$ for 100k steps. For the whisper-medium TLE adapter, we trained it with batch size 4 with a learning rate of $2.5e-5$ for 100k steps. Training the TLE module for Whisper-large took approximately 12 hours, and training the TLE module for Whisper-medium took approximately 7 hours. All training was performed on a Nvidia L40S gpu with seed 33.

For all Canary-1B experiments, we used a batch size of 4 with 2 gradient accumulation steps (for an equivalent batch size of 8) with a learning rate of $1e-6$. For all Canary-180M-flash experiments, we used a batch size of 8 with a learning rate of $1e-5$. We did not perform a full range of canary experiments to the computation requirements. Our primary purpose in including Canary model results is to demonstrate that TLE can work with models other than Whisper, rather than an in-depth adaptation approach comparison.

Each result in the main results tables (tables 2, 3, 5, and 6) is the result of a single run, with seed 33. We were unable to run multiple trials for all the experiments due to the computational requirements.

The Whisper analysis alone involved performing 48 training runs of Whisper ([TLE, TTS, TLE+TTS] * [8 ID/OOD datasets] * [Whisper-large/medium]). This number (48) does not include deep fusion training, TTS audio generation and TLE module training.

Generating the audio for the out-of-domain data using FastSpeech2 (Ren et al., 2022) took approximately 6 hours, and generating audio for the out-of-domain data using SpeechT5 (Ao et al., 2022) took approximately 5.5 hours.