Test-Time Training for Speech

Sri Harsha Dumpala [1 2] **Chandramouli Sastry** [1 2] **Sageev Oore** [1 2]

Abstract

In this paper, we study the application of Test-Time Training (TTT) as a solution to handling distribution shifts in speech applications. In particular, we introduce distribution-shifts to the test datasets of standard speech-classification tasks—for example, speaker-identification and emotion-detection—and explore how Test-Time Training (TTT) can help adjust to the distribution-shift. In our experiments that include distribution shifts due to background noise and natural variations in speech such as gender and age, we identify some key-challenges with TTT including sensitivity to optimization hyperparameters (e.g., number of optimization steps and subset of parameters chosen for TTT) and scalability (e.g., as each example gets its own set of parameters, TTT is not scalable). Finally, we propose using BitFit – a parameter-efficient fine-tuning algorithm proposed for text applications that only considers the bias parameters for fine-tuning – as a solution to the aforementioned challenges and demonstrate that it is consistently more stable than fine-tuning all the parameters of the model.

1. Introduction

Deep learning methods achieve impressive results in a variety of speech-based downstream tasks when the train and test data are in-distribution (Gulati et al., 2020; Snyder et al., 2017; Zou et al., 2022). In practice, however, the train and test distributions are usually different, i.e., there exists a distributional shift between the train and test data. In speech, such distributional shifts can be introduced due to inter-speaker variations such as speaking style, gender, age, etc., or due to background induced noises such as babble, living room, traffic, etc. These distributional shifts significantly degrade the performance of the deep learning models (Likhomanenko et al., 2021; Garcia-Romero et al., 2019; Parry et al., 2019). In real-world applications, some form of distributional shifts often occurs in the test data, making it of vital importance for deep learning models to be robust to these shifts.

One approach to handling distributional shift is with *train-time* techniques (Khurana et al., 2021; Li et al., 2020), in which we need to anticipate the type of distributional shifts that can occur during testing, and then train the model on data collected with this anticipated list of distributional shifts. In practice, the anticipated list of distributional shifts is non-exhaustive, and there is no guarantee that the trained model can generalize well to an unseen domain at test-time.

Another interesting approach which attained significant improvements in performance for imaging tasks is test-time training (TTT) (Sun et al., 2020; Liu et al., 2021; Gandelsman et al., 2022). In TTT, we update the model at inference using the test-sample. As the test sample does not have a label, a self-supervised learning task is used for this update.

The efficacy of TTT is impacted by the choice of the self-supervised learning task (Liu et al., 2021). Gandelsman et al. (2022) shows that masked auto-encoding on images is a suitable task for TTT. Motivated by the success of the transformer-based masked autoencoders (MAE) for speech (Huang et al., 2022a), we extend a test-time training approach based on MAE (Gandelsman et al., 2022) to speech in this work. To the best of our knowledge, this is the first work to adapt TTT to the speech domain. We show that TTT-MAE for speech shows significant improvements on three different downstream tasks under a variety of distributional shifts.

Two major challenges of using TTT during inference are maintaining stable performance robust to (reasonable) ranges of hyperparameters, and the potential high computational cost, which is due to both (a) increased memory requirements for updating all parameters of the model, and (b) inability to process a batch of samples if the individual samples in the batch are associated with different distributional shifts. We show that, for speech, it is possible to make significant improvements with respect to all of these issues, i.e. we improve stability, reduce memory requirements, and allow batch processing, all by using parameter-efficient training. Specifically, we show that using bias fine-tuning

[1]Vector Institute, Toronto, Canada [2]Dalhousie University, Halifax, Canada. Correspondence to: Sri Harsha Dumpala <sriharsha.d@dal.ca>, Sageev Oore <sageev@vectorinstitute.ai>.

Work presented at the ES-FoMo Workshop at ICML 2023, Copyright 2023 by the author(s).

(Zaken et al., 2022), we can process a batch of test samples, even under the condition that each test sample has a different distributional shift.

2. Method

Pre-training MAE. Our masked autoencoder (MAE) for speech, following Huang et al. (2022a); Chong et al. (2022), aims to reconstruct the masked patches of the speech Mel-spectrogram with an asymmetrical encoder-decoder architecture. Below, we provide a brief overview of the MAE.

First, we transform the input speech waveform into 128-dimensional Mel-spectrograms using a Hanning window of size 25ms for every 10ms. Next, we divide the spectrogram into a sequence of non-overlapping patches, each patch sized 16×16. These patches are then flattened and embedded with a linear projection layer. To provide positional information, fixed sinusoidal positional embeddings are added to the embedded patches. Afterward, we randomly mask 75% of the patches while preserving positional indices of all the patches. This enables the decoder to reconstruct the spectrogram. For the encoder, only unmasked patches are used to generate latent representations. The decoder then tries to reconstruct the original spectrogram, given latent representations of the encoder and masked patches as input. The latent representations and masked patches are organized in the initial order before being provided as input to the decoder. During training, the objective is to minimize mean squared error (MSE) between reconstructed and input spectrograms, averaged over the masked patches.

Train-time training. For the downstream tasks, we only use the encoder and discard the decoder. The latent representations generated by the encoder are provided as input to the task-specific classifier head. In this paper, we consider three different ways to use the pre-trained encoder for downstream tasks: 1) Linear probing: freeze the encoder to use it as a feature extractor and train only the classifier head; 2) Fine-tuning: train both encoder and classifier head, end-to-end, for the downstream task; 3) Linear-probing and fine-tuning (LP-FT): first train only the classifier head using linear probing, and then fine-tune both encoder and classifier head end-to end as explained in (Kumar et al., 2022).

Test-time training. Similar to (Sun et al., 2020; Gandelsman et al., 2022), we use a Y-shaped architecture: a shared encoder network e followed by two heads, a self-supervised head g and a task-specific classifier head h. Here, e and g are the encoder and decoder networks of the pre-trained MAE, respectively. The classifier head h, uses a linear projection from the dimension of the encoder features to the number of classes, depending on the downstream task.

When using TTT, we use linear-probing to update the weights of the classifier head h and freeze the weights of the shared encoder e during train-time training. As explained in (Gandelsman et al., 2022), we found that linear-probing is more suitable for TTT as compared to full-model fine-tuning. At test-time, the parameters of the shared encoder are updated to minimize the self-supervised loss. We explore the following approaches to update the weights of the encoder during test-time:

1) **Full fine-tuning**: In full fine-tuning, all parameters of the shared encoder are updated to minimize the self-supervised loss across various augmentations of a single test sample. However, the large size of pre-trained models, such as the MAE used in this study with 75M parameters, makes full fine-tuning computationally expensive during test time. Moreover, extensive steps of full fine-tuning during test time, as shown in Figure 1, can result in performance degradation. Since there is no validation data available at test time, early stopping is not an option.

Therefore, it is desirable to maintain stability in performance during Test-Time Training (TTT). However, even when using the SGD optimizer as suggested in (Gandelsman et al., 2022), full fine-tuning still exhibits performance degradation in speech-related tasks. Furthermore, TTT techniques entail a higher computational cost as TTT is approached as a one-sample learning problem. This means that the model can only be adapted to one test sample at a time, and batch processing is not feasible under the assumption that each test sample is subject to a different distribution shift.

Table 1. Trainable parameters in different blocks of our MAE for speech. Number of Parameters in one Encoder block are same as the number of parameters in each of the first, middle or last blocks of the encoder.

	#parameters(Proportion %)
MAE	74751488 (100.00)
Encoder	64457216 (86.23)
Decoder	10294272 (13.77)
One Encoder block	7087872 (9.48)
Bias - MAE	96000 (0.13)
Bias - Encoder	77568 (0.10)

To address these challenges, we investigate parameter-efficient fine-tuning techniques (PEFT) that have proven to be highly effective in supervised learning tasks within the field of NLP. However, their application in the context of TTT has not previously been explored.

2) **Parameter-efficient fine-tuning (PEFT)**: Here we explore different PEFT techniques to adapt the encoder to the test sample during inference. We conduct experiments using four different PEFT techniques for TTT: (i) First block: updating only the final/last block of the encoder, (ii) Last block: updating only the final/last block of the encoder, (iii) Middle block: updating only the middle block of the encoder, and (iv) Bias: updating only bias parameters of the

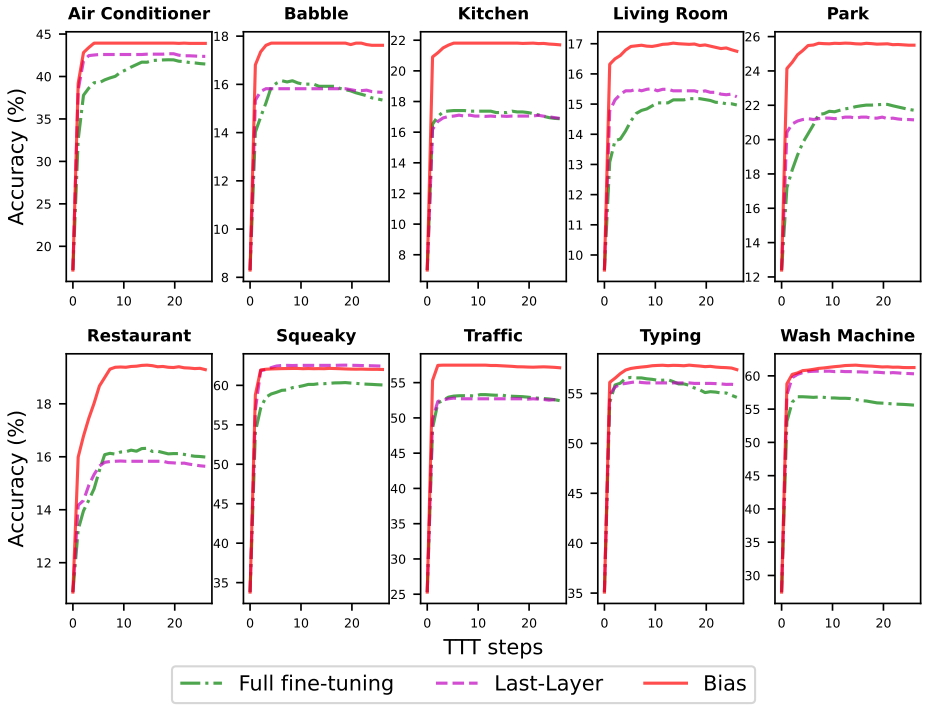

Figure 1. We compare the accuracy (speaker identification) across TTT steps between three different variants of TTT (full, last-layer and bias fine-tuning). For most distributional shifts, full fine-tuning shows degradation in performance with longer test-time training (after 20 steps) whereas bias and last layer fine-tuning show relatively stable performance even after 25 steps

encoder using the self-supervised task on the test sample. As shown in Table 1, the number of trainable parameters for updating any one block (first, middle, or last) accounts for 9.48% of the total parameters, while updating the bias parameters involves just 0.1% of the total parameters. In this study, we primarily focus on bias fine-tuning for TTT due to the following reasons:

- Bias fine-tuning is much more lightweight compared to full fine-tuning, training 830 times fewer parameters (0.78M vs. 64M).
- Zaken et al. (2022) demonstrated that, in supervised learning with limited training data, fine-tuning only bias parameters yields superior performance to full fine-tuning. Since TTT-MAE can be regarded as a one-sample unsupervised domain adaptation technique, we investigate effectiveness of updating only bias parameters during test-time training.
- TTT techniques incur higher computational costs as TTT is approached as a one-sample learning problem. This means that the model can only be adapted to one test sample at a time, and batch processing is not feasible when each test sample is drawn from a different distribution. We show that bias fine-tuning allows processing an entire batch of test samples.

Distributional shifts in speech. In this work, we identify and examine two types of distributional shifts: (1) those resulting from background noise that degrades/distorts speech quality, and (2) natural distributional shifts resulting from inter-speaker variations, including gender, age, and speaking style.

To generate degraded speech, we introduce background noise to clean speech signals at a specified signal-to-noise ratio (SNR) (see Figures 6 and 7 in Appendix). We identify two categories of background noise: (1) *Time-invariant*, where noise characteristics remain constant over time. Examples include additive white Gaussian noise (AWGN) and air conditioner (AC). (2) *Time-varying*, where noise characteristics change over time. Examples include background babble, living room, restaurant, reverberation, and traffic. Distributional shifts introduced by these types of noise are generally difficult to learn, even with adversarial training. Furthermore, some noises (e.g. babble, restaurant, and reverberation) exhibit patterns similar to speech, which can corrupt and contaminate information contained in the original signal, such as linguistic content, speaker characteristics, emotions, etc.

In this study, we explore significance of TTT in handling distributional shifts introduced by both background noise and natural variations in speech. We show that TTT-based

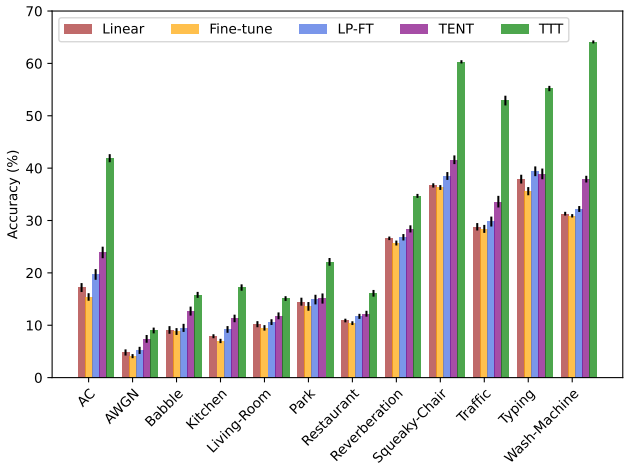

Figure 2. Compare TTT with non-TTT approaches (linear probing, fine-tuning, LP-FT, TENT) under different distributional shifts due to background noises (noises added at 0 dB). TTT significantly outperforms the non-TTT approaches across all the distributional shifts for the task of speaker identification. Results averaged over 3 runs

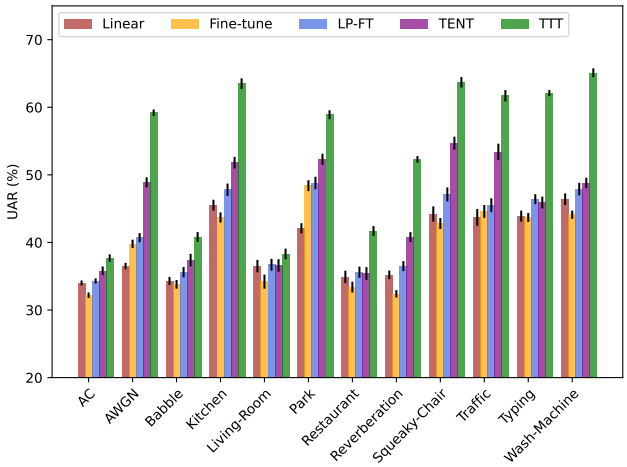

Figure 3. Compare TTT with non-TTT approaches (linear probing, fine-tuning, LP-FT, TENT) under different distributional shifts due to background noises (noises added at 0 dB). TTT significantly outperforms the non-TTT approaches across all the distributional shifts for the task of emotion classification. Results averaged over 3 runs. Emotion classification reported in terms of unweighted average recall (UAR (%))

methods consistently outperform non-TTT techniques with significant margins. Use of PEFT techniques, particularly BitFit, further improve the performance and stability of TTT under different distributional shifts. To show the effectiveness of TTT under different distributional shifts, we conduct the following experiments: 1) Training with clean speech and testing with speech corrupted by various background noises. 2) Training models on one speaking style and testing with another speaking style. 3) Training with speech from one gender and testing with the other gender. 4) Training with either younger or older speakers and testing with the other.

3. Experiments and Results

Implementation Details In all experiments, we use 9-layer ViT by default as the MAE encoder. For the decoder, we use a 3-layer Transformer. We use Voxceleb2 dataset (Chung et al., 2018) for pre-training the MAE. We pre-train the MAE for 120K steps with a batch size of 392, which takes about 7 days using 8 Nvidia RTX6000 24GB GPUs. The AdamW optimizer with an initial learning rate of 0.001 and a weight decay of 0.05 is applied. The learning rate has a cosine decay schedule (Chen et al., 2020) with 20K warmup steps. We transform raw waveform (mono-channel sampled at 16 KHz) into 128 Mel-frequency bands extracted with 20 ms Hanning window and 10 ms stride. We use the original speech spectrograms and apply no augmentations. We only use random masking of the spectrograms as an augmentation, with a masking ratio of 0.75 for pre-training. During TTT, for each test sample, we train only the encoder (freezing the decoder weights)for 20 steps using SGD optimizer with a fixed learning rate of 2.5e-3, batch size of 128, momentum of 0.9 and weight decay of 0.2. We also show performance plots for 25 steps of TTT (see Figure 1). During TTT, we follow the same procedure as pre-training: mask 75% of the input patches and provide the unmasked patches as input to the encoder whereas all the patches are provided to the decoder. Then update the encoder weights using reconstruction loss (MSE) on the masked patches as the objective function. We follow the same procedure for full, bias, first-layer, middle-layer and last layer fine-tuning.s Moreover, we do not use any augmentation on top of random masking for TTT. We performed most of these experiments using a single Nvidia A40 48GB GPU. Unless and otherwise specified, we report results for TTT after 20 TTT steps.

Dataset details. Table 3 (see Appendix) provide details of the datasets used in this work. We perform speaker identification using VCTK (Yamagishi et al., 2019); emotion recognition using CREMA-D (Cao et al., 2014), IEMOCAP (Busso et al., 2008), RAVDESS (Livingstone & Russo, 2018) and TESS (Dupuis & Pichora-Fuller, 2011) datasets; low-vocabulary speech recognition using Speech commands (Warden, 2018) dataset. We use original speech samples from each of the datasets in the pre-training and train-time training phases. To evaluate models' performance under different distributional shifts, we introduce diverse

Table 2. Performance of different variants of TTT under distributional shifts due to background noises added at 0dB SNR. Different variants of TTT: Full refers to Full fine-tuning; First, Middle, Last and Bias refer to fine-tuning only the first layer, middle layer, last layer and bias parameters, respectively. AWGN → additive white Gaussian noise. Bias fine-tuning performs better than other variants of TTT across most of the distribution shifts due to background noise

	Speaker Identification					Emotion Recognition				
	Full	First	Middle	Last	Bias	Full	First	Middle	Last	Bias
Air Conditioner	41.8	42.6	40.8	42.5	**43.9**	37.7	37.6	36.1	38.0	**42.8**
Gaussian (AWGN)	8.8	9.1	8.2	9.3	**10.4**	59.2	59.0	58.6	59.7	**62.8**
Babble	15.7	17.0	16.0	15.7	**17.3**	40.8	42.3	41.6	41.2	**42.8**
Kitchen	17.2	17.8	16.9	17.1	**21.8**	63.5	64.1	63.7	64.6	**65.9**
Living Room	15.1	16.3	16.0	15.4	**16.9**	38.3	39.4	38.5	**40.5**	40.3
Park	22.1	22.4	23.3	21.2	**25.6**	58.9	59.6	58.2	60.1	**62.1**
Restaurant	16.1	18.5	16.3	17.2	**19.4**	41.7	43.1	41.8	42.5	**43.9**
Reverberation	34.7	35.4	35.2	35.8	**37.9**	52.3	53.4	52.2	52.5	**54.7**
Squeaky-Chair	60.3	60.6	60.8	**62.5**	62.1	63.7	64.3	64.4	63.6	**65.5**
Traffic	52.9	55.9	54.3	52.7	**57.2**	61.7	65.3	63.2	65.7	**66.8**
Typing	55.2	57.4	57.4	55.9	**57.7**	62.1	63.2	59.7	63.3	**64.6**
Washing-Machine	56.0	60.8	60.5	60.4	**61.4**	64.1	65.1	65.0	65.4	**66.2**

background noises sourced from Microsoft's Scalable Noisy Speech Dataset (MS-SNSD) (Reddy et al., 2019). These noises are exclusively added during the testing phase and are not used in pre-training or train-time training of the models.

Test-time Training vs No Test-time Training. In Figure 2 and 3, we compare TTT for speech with different non-TTT techniques (linear probing, fine-tuning, LP-FT (Kumar et al., 2022) and TENT (Wang et al., 2021)) for different background noises unseen during training. TTT outperforms the non-TTT techniques for every unseen background noise condition. When comparing between non-TTT techniques, for most of the background noises, simple linear-probing performs better than fine-tuning. Similar to images in (Kumar et al., 2022), LP-FT performs better than both linear-probing and fine-tuning. TENT, a test-time adaptation technique, performs better or comparable to LP-FT but performs inferior to TTT. This can be attributed to the fact that TENT requires a large set of test samples to learn the distribution of the test sample but performs poorly under one-sample testing condition, as illustrated in (Khurana et al., 2022).

Even though TTT achieves significant improvements in performance under different distributional shifts, there are a few shortcomings of TTT when applied to speech such as high memory requirements during TTT, degradation in performance across TTT steps and inability to process batch of test samples, as shown in Figure 1 and explained in Section 2.

We overcome these issues by incorporating PEFT techniques into TTT. PEFT techniques are light weight as we need to fine-tune fewer parameters compared to full fine-tuning, thus requiring lesser memory (see Table 1). We also find that, for speech, PEFT techniques achieve better consistency in performance compared to full fine-tuning across the TTT steps (see Figure 1). Here we compare performance of different PEFT techniques used during TTT under different distributional shifts in Table 2. PEFT techniques of fine-tuning a specific block performs comparable to full-fine-tuning. Similar to (Lee et al., 2022), we find certain layers are more suitable for a specific set of background noises and there is no single block which is optimal for every background noise. Selecting a layer to fine-tune during TTT is not feasible as we only have a single test sample. To overcome this issue, we use Bitfit (Zaken et al., 2022) for TTT, where we fine-tune only the bias parameters of the encoder during TTT. Bitfit, a light-weighted fine-tuning approach, consistently performs better (comparable) to full-fine-tuning across all the distributional shifts.

4. Conclusion

In this work, we extend test-time training to speech related tasks such as speaker identification and emotion recognition under a variety of distributional shifts. In our application of TTT-MAE to speech, we observed that TTT is sensitive to hyperparameters such as training steps and subset of parameters considered for optimization. To overcome these issues, we applied PEFT techniques to make TTT more stable and scalable. We find that PEFT techniques, being light-weight, achieve better or comparable performance to full fine-tuning. Specifically, we find that bias fine-tuning, motivated from BitFit, improves both performance and stability. Further, we propose a new approach to process batches of test samples using bias fine-tuning for TTT.

References

Azimi, F., Palacio, S., Raue, F., Hees, J., Bertinetto, L., and Dengel, A. Self-supervised test-time adaptation on

video data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3439–3448, 2022.

Baade, A., Peng, P., and Harwath, D. Mae-ast: Masked autoencoding audio spectrogram transformer. *arXiv preprint arXiv:2203.16691*, 2022.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 34, 2020.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Banerjee, P., Gokhale, T., and Baral, C. Self-supervised test-time learning for reading comprehension. *arXiv preprint arXiv:2103.11263*, 2021.

Boudiaf, M., Denton, T., van Merriënboer, B., Dumoulin, V., and Triantafillou, E. In search for a generalizable method for source free domain adaptation. *arXiv preprint arXiv:2302.06658*, 2023.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4): 335–359, 2008.

Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., and Verma, R. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chong, D., Wang, H., Zhou, P., and Zeng, Q. Masked spectrogram prediction for self-supervised audio pre-training. *arXiv preprint arXiv:2204.12768*, 2022.

Chung, J. S., Nagrani, A., and Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

Darestani, M. Z., Liu, J., and Heckel, R. Test-time training can close the natural distribution shift performance gap in deep learning based compressed sensing. In *International Conference on Machine Learning*, pp. 4754–4776. PMLR, 2022.

Dou, Q., Coelho de Castro, D., Kamnitsas, K., and Glocker, B. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

Dupuis, K. and Pichora-Fuller, M. K. Recognition of emotional speech for younger and older talkers: Behavioural findings from the toronto emotional speech set. *Canadian Acoustics*, 39(3):182–183, 2011.

Gandelsman, Y., Sun, Y., Chen, X., and Efros, A. A. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, 2022.

Garcia-Romero, D., Snyder, D., Watanabe, S., Sell, G., McCree, A., Povey, D., and Khudanpur, S. Speaker recognition benchmark using the chime-5 corpus. In *Interspeech*, pp. 1506–1510, 2019.

Gong, Y., Lai, C.-I., Chung, Y.-A., and Glass, J. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10699–10709, 2022.

Goyal, S., Sun, M., Raghunathan, A., and Kolter, Z. Test-time adaptation via conjugate pseudo-labels. *arXiv preprint arXiv:2207.09640*, 2022.

Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pp. 5036–5040, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Hu, H., Yang, X., Raeesy, Z., Guo, J., Keskin, G., Arsikere, H., Rastrow, A., Stolcke, A., and Maas, R. Redat:

Accent-invariant representation for end-to-end asr by domain adversarial training with relabeling. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6408–6412. IEEE, 2021.

Huang, P.-Y., Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., and Feichtenhofer, C. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022a.

Huang, W., Zhang, Z., Yeung, Y. T., Jiang, X., and Liu, Q. SPIRAL: self-supervised perturbation-invariant representation learning for speech pre-training. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b.

Huo, Z., Sim, K. C., Li, B., Hwang, D., Sainath, T. N., and Strohman, T. Resource-efficient transfer learning from speech foundation model using hierarchical feature fusion. *arXiv preprint arXiv:2211.02712*, 2022.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pp. 709–727. Springer, 2022.

Kenton, J. D. M.-W. C. and Toutanova, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Khurana, A., Paul, S., Rai, P., Biswas, S., and Aggarwal, G. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2022.

Khurana, S., Moritz, N., Hori, T., and Le Roux, J. Unsupervised domain adaptation for speech recognition via uncertainty driven self-training. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6553–6557. IEEE, 2021.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Li, L., Mak, M.-W., and Chien, J.-T. Contrastive adversarial domain adaptation networks for speaker recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Li, X., Wang, C., Tang, Y., Tran, C., Tang, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 827–838, 2021.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Likhomanenko, T., Xu, Q., Pratap, V., Tomasello, P., Kahn, J., Avidov, G., Collobert, R., and Synnaeve, G. Rethinking evaluation in ASR: are our models robust enough? In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association*, pp. 311–315. ISCA, 2021.

Lin, G.-T., Li, S.-W., and Lee, H.-y. Listen, adapt, better wer: Source-free single-utterance test-time adaptation for automatic speech recognition. *arXiv preprint arXiv:2203.14222*, 2022.

Liu, A. T., Yang, S.-w., Chi, P.-H., Hsu, P.-c., and Lee, H.-y. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6419–6423. IEEE, 2020.

Liu, Y., Kothari, P., Van Delft, B., Bellot-Gurlet, B., Mordan, T., and Alahi, A. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

Livingstone, S. R. and Russo, F. A. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5), 2018.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Lu, X., Matsuda, S., Hori, C., and Kashioka, H. Speech restoration based on deep learning autoencoder with layer-wised pretraining. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

Mohamed, A., Lee, H.-y., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., et al. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 2022.

Mummadi, C. K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., and Metzen, J. H. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., and Kashino, K. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. *arXiv preprint arXiv:2204.12260*, 2022.

Osowiechi, D., Hakim, G. A. V., Noori, M., Cheraghalikhani, M., Ben Ayed, I., and Desrosiers, C. Tttflow: Unsupervised test-time training with normalizing flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2126–2134, 2023.

Parry, J., Palaz, D., Clarke, G., Lecomte, P., Mead, R., Berger, M., and Hofer, G. Analysis of deep learning architectures for cross-corpus speech emotion recognition. In *Interspeech*, pp. 1656–1660, 2019.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673, 2020.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Reddy, C. K., Beyrami, E., Pool, J., Cutler, R., Srinivasan, S., and Gehrke, J. A scalable noisy speech dataset and online subjective test framework. *arXiv preprint arXiv:1909.08050*, 2019.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33: 11539–11551, 2020.

Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, volume 2017, pp. 999–1003, 2017.

Song, P., Ou, S., Du, Z., Guo, Y., Ma, W., Liu, J., and Zheng, W. Learning corpus-invariant discriminant feature representations for speech emotion recognition. *IEICE TRANSACTIONS on Information and Systems*, 100(5): 1136–1139, 2017.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Sung, Y.-L., Cho, J., and Bansal, M. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022a.

Sung, Y.-L., Cho, J., and Bansal, M. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022b.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Wang, R., Sun, Y., Gandelsman, Y., Chen, X., Efros, A. A., and Wang, X. Test-time training on video streams, 2023.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Yamagishi, J., Veaux, C., MacDonald, K., et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *University of Edinburgh.*

The Centre for Speech Technology Research (CSTR), 2019.

Yang, F.-E., Cheng, Y.-C., Shiau, Z.-Y., and Wang, Y.-C. F. Adversarial teacher-student representation learning for domain generalization. *Advances in Neural Information Processing Systems*, 34:19448–19460, 2021.

Yang, Z., Ding, M., Guo, Y., Lv, Q., and Tang, J. Parameter-efficient tuning makes a good classification head. *arXiv preprint arXiv:2210.16771*, 2022.

Zaken, E. B., Goldberg, Y., and Ravfogel, S. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 1–9, 2022.

Zou, H., Si, Y., Chen, C., Rajan, D., and Chng, E. S. Speech emotion recognition with co-attention based multi-level acoustic information. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7367–7371. IEEE, 2022.

Table 3. Details of the datasets used for pre-training and downstream tasks. Total Duration is in hours and Average Length is in seconds. The tasks Speaker ID, Emotion Recog., Ltd Vocab. ASR refer to Speaker identification, emotion recognition and limited vocabulary ASR, respectively.

Dataset	Task	#Classes	#Speakers	#Total Samples	Total Duration	Average Length
VoxCeleb2	Pre-training	–	5994	770000	2300	8.6
VCTK	Speaker ID	109	109	42075	44	3.8
CREMA-D	Emotion Recog.	4	91	4397	3.04	2.5
IEMOCAP	Emotion Recog.	4	10	5531	7.00	4.6
RAVDESS	Emotion Recog.	4	24	672	0.70	3.7
TESS	Emotion Recog.	4	2	1600	0.92	2.1
Speech Commands	Ltd. Vocab. ASR	12	2618	105829	29.5	1.0

A. Related work

Domain adaptation and generalization. These methods are based on the assumption that models will have access to labelled data from the train distribution and unlabelled (labelled) data from the test distribution. A common strategy in domain adaptation is to learn domain invariant features between train and test data distributions (Sun & Saenko, 2016; Tzeng et al., 2017; Long et al., 2018). Another approach is to perform self-training on the test distribution by generating pseudo-labels for the unlabelled data (Xie et al., 2020). Domain generalization techniques mainly resort to adversarial training, meta learning or adversarial data augmentation (Yang et al., 2021; Balaji et al., 2018; Dou et al., 2019; Volpi et al., 2018). In speech, domain adaptation and generalization techniques have been applied to tasks such as automatic speech recognition, emotion recognition and speaker classification (Khurana et al., 2021; Hu et al., 2021; Song et al., 2017; Li et al., 2020) All these methods assume information about the test domain is available during training.

Learning at inference. In the above techniques, the model is trained to generalize to all possible distributional shifts. But anticipating every possible distributional shift at train time is not feasible, particularly in real world applications. Most models trained using domain generalization techniques are fixed during inference even when the test distribution changes. To alleviate this problem, another line of work is to adapt the model to test samples at inference. Methods to update the model at inference can be classified into two types: test-time adaptation and test-time training.

Test-time Adaptation (TTA): These methods allow using off-the-shelf models without any additional training. In general terms, test-time adaptation focuses on adapting models that were not trained with a special configuration prior to being used at inference (Goyal et al., 2022; Boudiaf et al., 2023). One of the first approaches of this category, called TENT (Wang et al., 2021), requires the model and target data. It then updates the model layers by minimizing the Shannon entropy of predictions. Mummadi et al. (2021) improves TENT by using a log-likelihood ratio instead of entropy, and by estimating target batch statistics. Another approach is to update batch normalization (BN) statistics using large number of test samples (Nado et al., 2020; Schneider et al., 2020). SITA (Khurana et al., 2022) is one such approach which can be used on a single test data example. SITA generates a pseudo-batch by randomly augmenting this example and then computes statistics on this pseudo-batch. TTA in computer vision is heavily targeted on the BN layer's adaptation by re-estimating batch statistics on target data. In this work we use transformer models, which have achieved state-of-the-art performance on many speech-based downstream tasks. Since transformer models are not equipped with BN layers, as the length of batched input sequences are different, TTA techniques cannot be applied directly to speech. Only one work has applied TTA to speech (Lin et al., 2022); this approach appears limited to ASR models trained with CTC loss.

Test-time training (TTT): The basic paradigm in TTT (Sun et al., 2020) is to use a test-time task (usually a self-supervised learning task) besides the main task during training, and update the pre-trained model using test data with the (self-supervised) test-time objective before the final prediction. Sun et al. (2020) uses rotation prediction as the self-supervised task. Later, TTT++ (Liu et al., 2021) considers contrastive loss as the self-supervised task in addition to aligning the features by comparing the statistics of the source data with those of the current test batch. Recently, Osowiechi et al. (2023) minimizes the distribution shift, between the train and test distributions, estimated using normalizing flows. Gandelsman et al. (2022) shows that using masked autoencoding (MAE) (He et al., 2022) as the self-supervised task for TTT achieves substantial improvements in image recognition under various distributional shifts. TTT-based techniques are applied to other domains such as videos (Azimi et al., 2022; Wang et al., 2023), natural language processing (NLP) (Banerjee et al., 2021) and compressed sensing of medical images (Darestani et al., 2022), where the self-supervised task varies across domains. In

this work, we extend TTT framework to speech domain for the first time. Our work extends the TTT-MAE framework (Gandelsman et al., 2022) to speech by using work on audio MAE trained with spectrograms (Huang et al., 2022a; Chong et al., 2022).

Parameter efficient fine-tuning (PEFT). Fine-tuning entire pre-trained models achieves state-of-the-art performance for various downstream tasks (Kenton & Toutanova, 2019; Raffel et al., 2020; Mohamed et al., 2022). However, as the size of these models increases rapidly, updating the models in parameter-efficient ways becomes crucial (Yang et al., 2022; Chen et al., 2022). In the NLP domain, PEFT techniques typically refer to either: a) insertion of new learnable modules with fewer parameters as compared to the whole model; or b) modifying carefully selected parameters of the model. In (Houlsby et al., 2019; Pfeiffer et al., 2020; Sung et al., 2022b; Li & Liang, 2021), only the additional parameters added to the pre-trained models are fine-tuned for the downstream tasks while (Yang et al., 2022; Zaken et al., 2022) fine-tune a subset of parameters for downstream tasks without inserting any new modules. Zaken et al. (2022) show that just fine-tuning the bias parameters, which constitute about 0.1% of the overall parameters, can outperform full fine-tuning, especially for small datasets. Motivated from NLP, PEFT techniques are also applied to pre-trained models trained using images (Jia et al., 2022; Chen et al., 2022; Sung et al., 2022a; Lee et al., 2022), and speech (Li et al., 2021; Huo et al., 2022). In this work, we study PEFT techniques applied to speech in the context of TTT, focusing on the following questions: 1) Can PEFT techniques achieve comparable or improved performance compared to full fine-tuning in TTT? 2) Can PEFT be more stable than full fine-tuning in TTT?

Self-Supervision using masking in speech. Denoising autoencoders (Vincent et al., 2008) is one of the earliest forms of self-supervision in speech (Lu et al., 2012). Subsequent advancements in self-supervision have predominantly focused on masked language modeling (MLM) (Liu et al., 2020; Baevski et al., 2020; Hsu et al., 2021; Gong et al., 2022; Huang et al., 2022b). Many of these works employ transformer networks trained with MLM frameworks to achieve state-of-the-art performance on various speech-related downstream tasks. More recently, the concept of a masked autoencoder (MAE) based on the Vision Transformer (ViT) has been extended to the audio and speech domain (Huang et al., 2022a; Baade et al., 2022; Chong et al., 2022; Niizumi et al., 2022). In MAE, only the non-masked spectrogram patches are encoded, distinguishing it from other approaches that encode both masked and non-masked input (wave/spectrogram) segments for self-supervised pre-training. This distinction makes MAE-based models computationally appealing. In this work, we analyze the effect of distributional shift on the performance of MAE for speech and use TTT-based approach to enhance the robustness of MAE in the presence of such distributional shifts.

Table 4. Emotion recognition under natural distributional shifts caused by (a) **speaking style variations**: Train model using CREMA-D (CRE) dataset and test with other datasets i.,e IEMOCAP (IEM), RAVDESS (RAV) and TESS. Column CRE: matched condition – train and test on CREMA-D dataset, (b) **Gender variations**: Train model using speech data from speakers of one gender and test with speakers from other gender. F-M: train on female and test on male; M-F: train on male and test on female; F-F and M-M: train and test speakers from the same gender. We use CREMA-D dataset for these experiments. TTT variants (Full and Bias fine-tuning) outperform non-TTT methods (linear probing and fine-tuning) across different natural distribution shifts.

Table 5. *
(a) evaluate across datasets - variation in speaking style

	CRE	IEM	RAV	TESS
Linear	52.2	35.4	34.7	33.1
Fine-tune	61.5	32.5	33.8	34.3
Full	66.8	45.5	**44.3**	39.4
Bias	**67.3**	**45.9**	44.2	**40.6**

Table 6. *
(b) Across genders - variation in gender

	F-F	F-M	M-M	M-F
Linear	54.2	31.6	52.1	30.9
Fine-tune	59.4	29.8	58.6	30.2
Full	66.0	54.1	61.5	56.8
Bias	**68.2**	**56.5**	**64.0**	**60.5**

Evaluation under natural distributional shifts. We evaluate performance of TTT techniques across different natural distributional shifts caused by inter-speaker variations, e.g., speaking style (Table 5), gender (Table 6), and age (Table 7).

In Table 5, we compare performance of non-TTT (Linear probing and fine-tuning) with TTT (full fine-tuning (Full) and Bias) techniques for distributional shift due to speaking style variation. Here we train on CREMA-D (emotively acted utterances in American English) and test with samples from IEMOCAP (emotive conversations enacted in American English), RAVDESS (emotively acted utterances in North American English) and TESS (emotively acted utterances in Canadian English). TTT-based techniques achieve significantly better performance compared to non-TTT methods for speaking style variations.

To evaluate the performance of TTT in addressing distributional shifts caused by gender variations for the task of emotion classification (CREMA-D dataset), we conducted training using speech data from one gender (either female or male) and carried out testing using speech data from the opposite gender (male or female). Table 6 provides a comparison between non-TTT and TTT-based techniques. For gender variations, non-TTT approaches exhibit a steep decline in performance between matched (F-F and M-M) and mismatched (F-M and M-F) conditions. In contrast, TTT approaches (full and Bias) show very small degradation in performance between matched and mismatched conditions. For both matched and mismatched conditions, TTT with Bias fine-tuning performs better than full fine-tuning.

We evaluate performance of TTT under age variations using TESS dataset for the task of emotion recognition (refer to Table 7). The TESS dataset was collected from two female speakers: a younger (Y) speaker aged 26 years and an older speaker aged 64 years. In the O-Y scenario, we trained the model using younger (Y) speaker's speech and tested it with older (O) speaker's speech. Similarly, in the Y-O scenario, we trained with the older speaker (O) and tested with the younger (Y) speaker. For unmatched train-test conditions (Y-O and O-Y), TTT techniques outperformed non-TTT techniques, with bias fine-tuning performing better than full fine-tuning. Interestingly, even in matched conditions (Y-Y and O-O), TTT-based techniques performed comparably or better than non-TTT techniques for all the natural distributional shifts.

Table 7. Emotion classification under age variation. We use TESS dataset consisting of two female speakers: one Younger (Y) and one older (O). We train on one speaker and test using other speaker's speech. Y-O $\rightarrow$ train on Y and test with O; O-Y $\rightarrow$ train on O and test with Y

	Y-Y	Y-O	O-O	O-Y
Linear	49.3	29.8	50.2	31.7
Fine-tune	51.8	28.2	52.3	30.4
Full	52.2	40.8	52.5	41.6
Bias	**52.4**	**41.9**	**52.9**	**42.8**

B. Results on Speech Commands Dataset

In Table 8, we compare the performance of non-TTT techniques (Linear, Fine-tune) with TTT-based techniques (TTT-Full, TTT-Bias) using speech commands dataset for the task of low-vocabulary speech recognition. Results show that TTT techniques outperform non-TTT techniques under different distributional shifts. Also in TTT-based techniques, the Bias fine-tuning performs better (comparable) to full fine-tuning across all the distributional shifts.

Table 8. Results on Speech Commands dataset for the task of low-vocabulary speech recognition. performance in terms of Accuracy (in % averaged over 3 random seeds) when tested with samples corrupted with different noises. All noises added at 0dB SNR. Results show that the TTT-based techniques significantly improves the performance over non-TTT techniques under different distributional shifts. Bias fine-tuning performs better than full-fine-tuning across all distributional shifts.

	Clean	AC	Babble	Kitchen	Living	Park	Restaurant	Reverb	Traffic	Typing
Linear	92.3	51.4	30.9	54.2	35.8	50.2	37.8	40.7	56.1	53.8
Fine-tune	95.2	50.9	29.3	53.6	34.2	49.3	38.3	39.2	55.4	54.5
TTT-Full	**95.4**	62.2	37.4	63.3	45.5	58.3	45.5	55.1	63.7	64.1
TTT-Bias	94.8	**63.7**	**39.2**	**64.4**	**46.4**	**60.4**	**47.8**	**57.2**	**66.4**	**67.3**

C. Improved utilization of computational resources.

We discuss an approach to process a batch of test samples using TTT for the real world scenario of each test sample associated with a different distributional shift. Since each test sample gets its own copy of parameters, TTT can be applied with only one test sample at a time. However, bias fine-tuning gives us an unique opportunity to apply TTT to a batch of examples while still ensuring that each sample gets its own copy of parameters. For example, consider a simple linear layer containing weight matrix W and a bias-vector b; given a batch $\mathbf{x}$ containing B samples, the output of this layer can be written as $\mathbf{y} = W\mathbf{x} + b$. In TTT-bias, each example in this batch gets its own b while sharing W; now, suppose that the

linear layer gets an additional input Δb (initialized to zeros) that contains B learnable vectors and is used to compute the output $\mathbf{y} = W\mathbf{x} + b + \Delta b$: since Δb contains B learnable parameters, each parameter can be optimized independently while taking advantage of the GPU batch-processing. We illustrate this in Figure 4: in practice, this can be implemented as forward hooks in PyTorch without changing the model code.

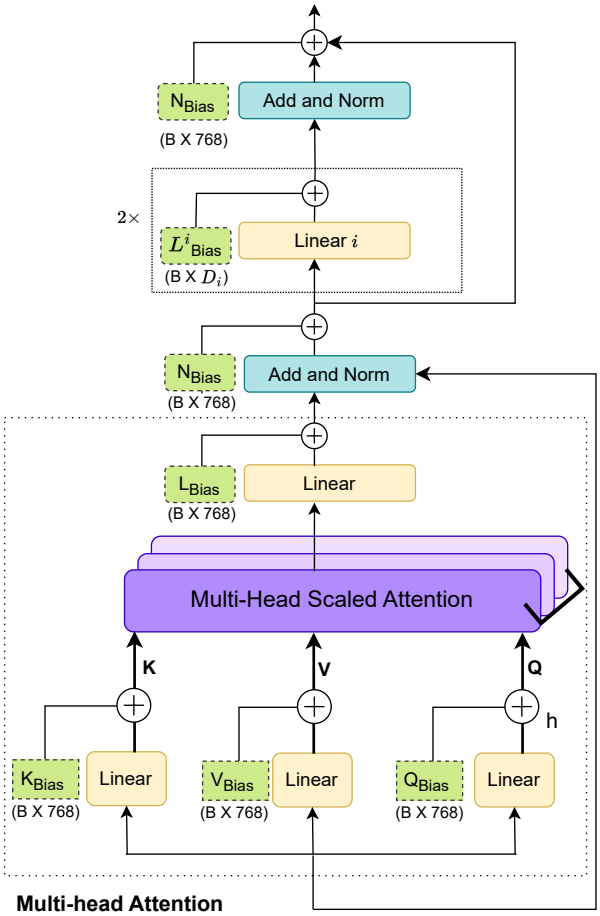

Figure 4. Illustration of TTT-Bias Fine-tuning on Batch of Test Samples: for each example in a batch containing B samples, we create a learnable parameter corresponding to each trainable bias as shown in green boxes and the outputs of the modules are adjusted accordingly as shown. Only the parameters in green boxes need to be fine-tuned and since these parameters are not shared across examples in a batch, TTT-bias can take advantage of GPU batch processing to improve throughput.

Figure 5 shows the timing analysis for processing a batch of test samples using (1) TTT (process single sample at a time), (2) TTT-Batch: Process a batch of test samples using the approach illustrated in Figure 4. As shown in Figure 5, as the batch size increases, the throughput of TTT-Batch improves.

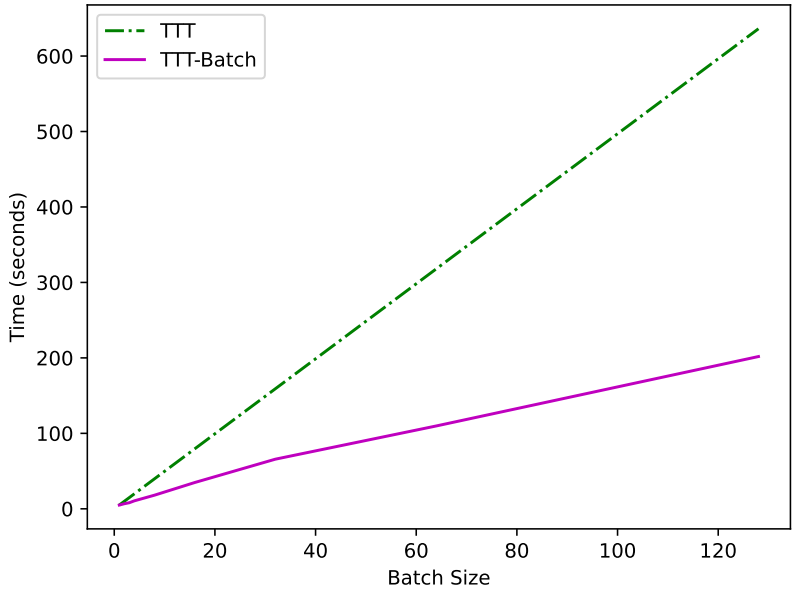

Figure 5. Timing analysis of TTT-Bias Fine-tuning on Batch of Test Samples. Here we provide the timing analysis of the approach shown in Figure 4 to process a batch of test samples. We vary the batch size of the test samples and compare the time taken by TTT (process only one sample at a time) and TTT-Batch (process batch of test samples simultaneously. We show that the difference in time taken by TTT-Batch and TTT increases as the batch size increases). For the purpose of this illustration, we use 32 augmentations for each TTT step

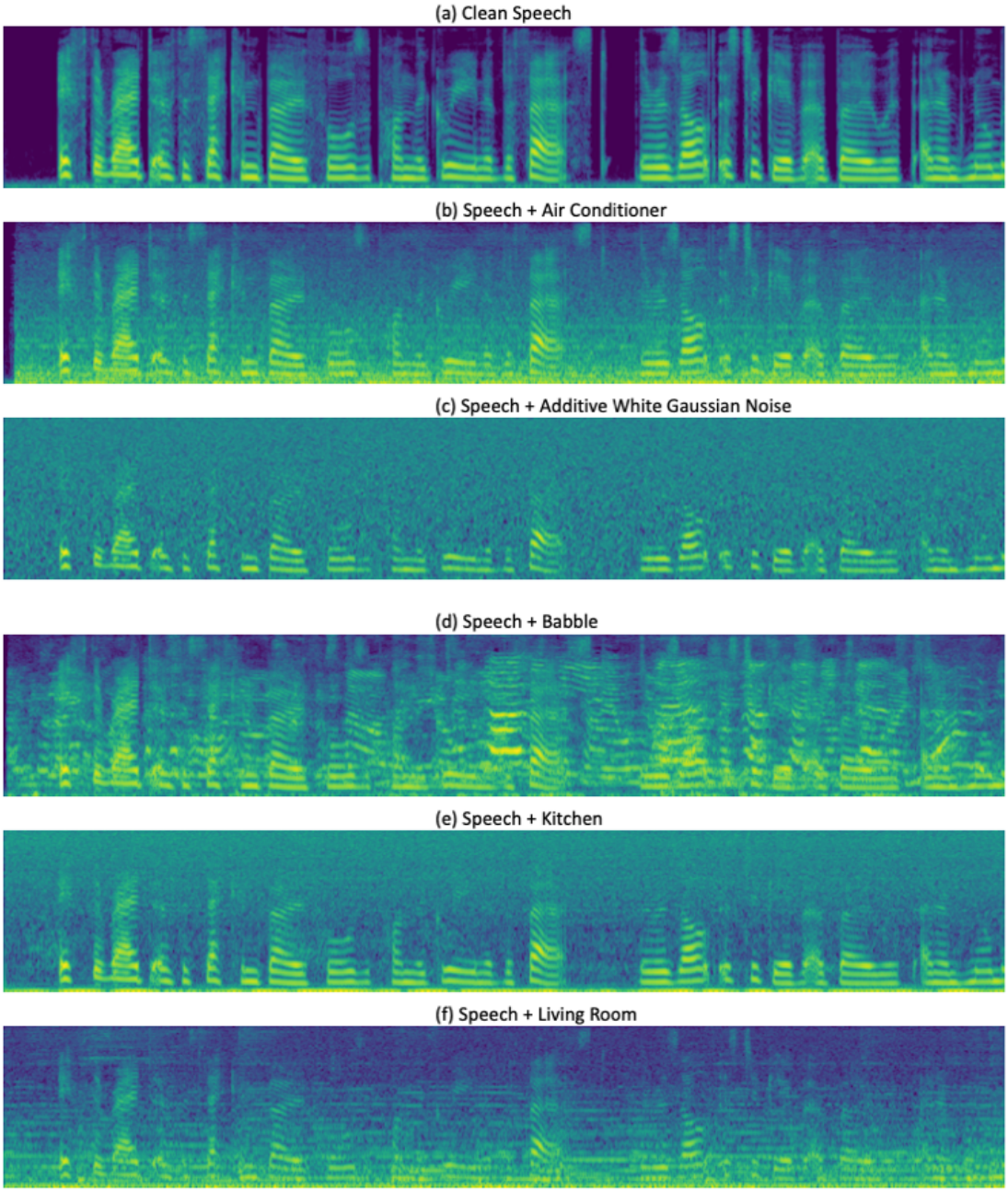

Figure 6. Mel-spectrograms of speech with different distributional shifts due to background noises added at 0 dB SNR. Remaining spectrograms (g-m) are continued in Figure 7. (a) spectrogram of clean speech. (b-m) shows the Mel-spectrograms of speech when added with different background noises. Characteristics of background noises (e)-(i) and (k) vary with time thus distorting the characteristics of speech critical for speech-based applications. For instance, panel (d) shows how babble noise, which has similar characteristics to speech, when added to speech, introduces patterns similar to clean speech along both time and frequency dimensions, and thus distorts the patterns in clean speech. Similarly, living room (see (f) and restaurant (see (h)) noises distort speech patterns in time and frequency dimensions.

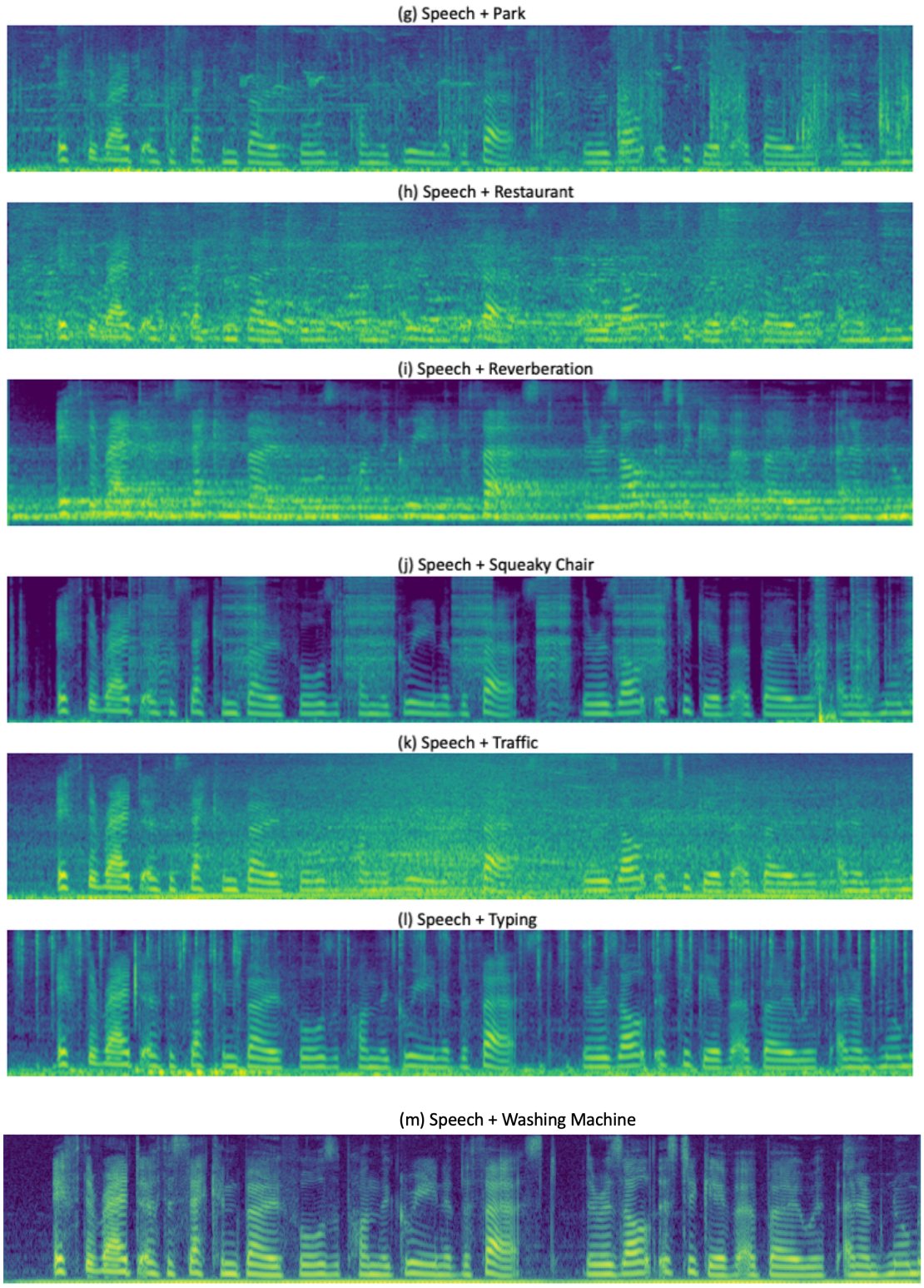

Figure 7. Mel-spectrograms of speech with distributional shifts caused due to different background noises added at 0 dB SNR