DOMAIN ADAPTATION WITH EXTERNAL OFF-POLICY ACOUSTIC CATALOGS FOR SCALABLE CONTEXTUAL END-TO-END AUTOMATED SPEECH RECOGNITION

David M. Chan[⋆†] *Shalini Ghosh*[†] *Ariya Rastrow*[†] *Björn Hoffmeister*[†]

[⋆] University of California, Berkeley [†] Amazon Alexa AI

ABSTRACT

Despite improvements to the generalization performance of automated speech recognition (ASR) models, specializing ASR models for downstream tasks remains a challenging task, primarily due to reduced data availability (necessitating increased data collection), and rapidly shifting data distributions (requiring more frequent model fine-tuning). In this work, we investigate the potential of leveraging external knowledge, particularly through off-policy generated text-to-speech key-value stores, to allow for flexible post-training adaptation to new data distributions. In our approach, audio embeddings captured from text-to-speech are used, along with semantic text embeddings, to bias ASR via an approximate k-nearest-neighbor (KNN) based attentive fusion step. Our experiments on LibiriSpeech and in-house voice assistant/search datasets show that the proposed approach can reduce domain adaptation time by up to 1K GPU-hours while providing up to 3% WER improvement compared to a fine-tuning baseline, suggesting a promising approach for adapting production ASR systems in challenging zero and few-shot scenarios.

Index Terms— speech recognition, transfer learning, fine-tuning, adaptation, context

1 Introduction & Background

One of the most challenging problems in automated speech recognition (ASR) is specializing large-scale models, particularly speech encoders, for downstream applications that often (a) have fewer labeled training examples, and (b) rapidly evolving distributions of speech data. The traditional approach to this problem is to frequently collect fresh data, which can be used to re-train and specialize models, leveraging tools such as domain-prompts [1], incremental-learning [2], knowledge distillation [3], or hand-written grammars [4] to reduce the impact of re-training the model for the downstream application. Unfortunately, for data that changes on a rapid basis, such as product listings or applications requiring per-customer specialization, such methods, while effective, are either inherently slow or remain computationally infeasible.

In this work, we propose a method that leverages external text data catalogs – large lists that can contain as much as 10 million specialized words or phrases – to improve the performance of models during both the fine-tuning process, and when specializing an already fine-tuned model to a new dataset. Here are the key highlights of our approach: first, we generate a key-value external knowledge store that maps an audio representation of each text element of the catalog (usually consisting of 1M-10M examples) to a semantic representation of the text. Next, we train a model that leverages this external store by attending over retrieved key/value pairs, which we retrieve through approximate k-nearest neighbors. Relying on an external, constant, and off-policy key-value store means that this store can be updated during specialization, requiring only an updated list of phrases for each new model instead of additional fine-tuning.

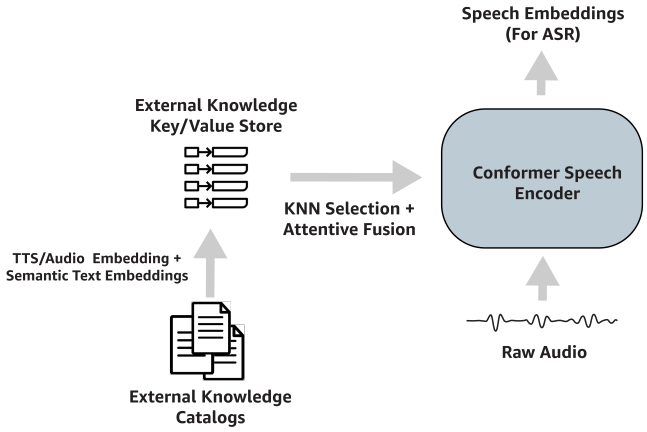

Fig. 1: An overview of our method leveraging text-to-speech mappings for contextual ASR. Using data from a text catalog, we generate audio and text representations to generate mappings from audio key to text value. To leverage these mappings for ASR, we implement a K-Nearest Neighbors attention in the speech encoder during the fine-tuning (or training) phase.

Leveraging external text data to improve the performance of audio encoders in ASR models has been studied for a long time. Perhaps the closest work to our proposed model was presented by Chen et al. [6], who used attention over a *local* set of LSTM-based grapheme/phoneme embeddings to augment the audio encoder. They found that biasing the encoder with only 40 contextual text entities per utterance leads to improvements of up to 75% on specialized test datasets. Similarly, Sathyendra et al. [7] and Chang et al. [8] demonstrate WER reductions when small ($<$100) contexts are fused with an RNN-T in a

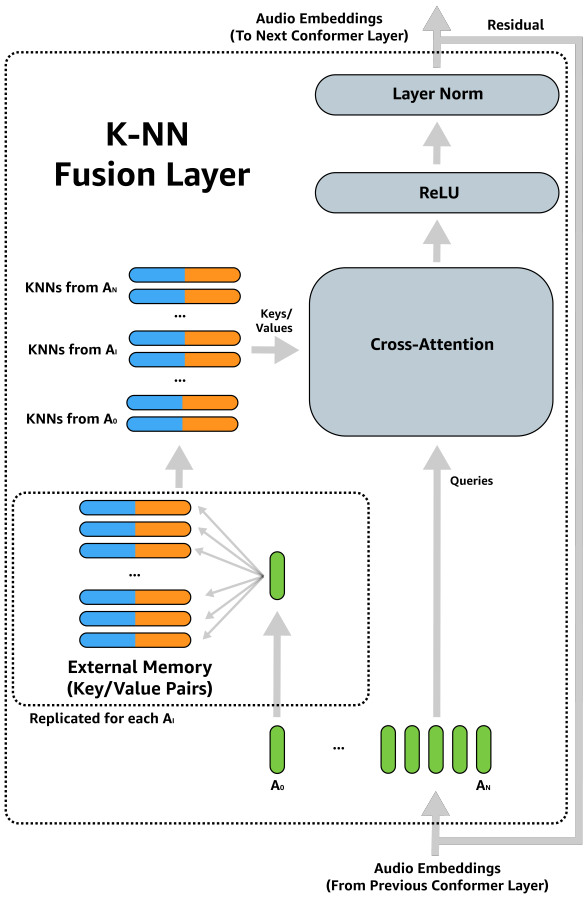

Fig. 2: Overview of the K-NN fusion layer. For each audio frame embedding, we extract approximate KNNs using audio keys from our catalog. These KNNs form a context key/value store for a standard cross-attention layer [5], where the queries are the incoming audio frame embeddings.

multi-head attention-based process. Our method differs in that it is designed primarily for *domain specialization*, whereas existing biasing methods are focused on *personalization*. This is shown foremost in the scale of the catalogs – while in prior work, each utterance may have at most 100 utterances in their context, we leverage catalogs with up to 10M samples. Thus, our approach is designed to compensate for general domain shift, rather than supplementing ASR performance through limited personalization. Further, while current biasing approaches focus on late-stage fusion, we use deep fusion in the model network, which we demonstrate (Table 2) is more effective. While biasing the *speech encoder* has been under-explored, many works [9, 10, 11, 12, 13, 14, 15, 16, 17] have shown the importance of biasing the language model in the ASR stack.

Outside of ASR, it has been shown that models augmented with external memory generated from large-scale text data have the potential to outperform similarly sized models without external knowledge. Borgeaud et al. [18] recently demonstrated that leveraging external-knowledge lookup for natural language models can lead to efficiency improvements of up to 25x, and Wu et al. [19] showed that expanding the context of

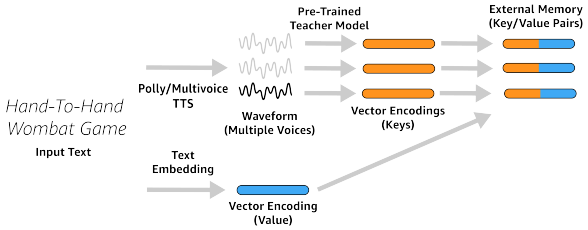

Fig. 3: Overview of our text-catalog encoding process. For each catalog entry, we generate TTS-based audio encoding that forms the "key" vector in the key-value pair. The value is a semantic text-embedding of the entry. Key/value pairs are assembled into the external memory, referenced in Figure 2.

standard text transformers through external cached key-value pairs can lead to significant perplexity improvements on the standard language modeling task. General memory augmentations have also been shown to be useful in translation [20], RL [21], and image generation [22].

Inspired by Borgeaud et al. [18] and Wu et al. [19], we apply a context embedding approach with a focus on ASR, leveraging TTS-generated audio data and semantic text embeddings to bias the speech encoder of a conformer model. To the best of our knowledge, using TTS to encode textual context has not been explored in prior work. Our key contributions are three-fold. (1) We outline the first method (to our knowledge) to leverage large-scale text data for contextual biasing of the speech encoder. (2) We show that our approach combined with an approximate K-NN lookup yields improved WER on ASR models, particularly when encoded catalogs match the target domain. (3) We show that our approach provides accurate solutions under the constraint of quick reactions to distribution changes (e.g., fast catalog updates for sporting events, changes in personal catalogs), without model retraining.

2 Methods

An overview of our method is given in Figure 1. Our approach consists of two key components: (1) A method for generating key-value mappings between the audible speech and a text representation of the catalog, which we call an "external memory" and (2) An attention-based module for fusing the "external memory" with the existing speech encoder. The external memory must be capable of offline and off-policy updates to enable memory alteration without incurring re-training costs.

2.1 Generating the External Memory

An overview of the external-memory generation process is shown in Figure 3. Our approach generates the external memory consisting of audio-embedding key/text-embedding value pairs from a text-only catalog. To generate the audio-embedding key, we use text-to-speech (TTS) to generate waveform representations of the audio data, and then embed these waveform representations using the pre-trained speech encoder model. To generate the text-embedding values, we leverage

off-the-shelf semantic text embedding methods, including 1-hot, GLoVE [23] and BERT-style embedding [24] approaches.

TTS: We explore two TTS modules to generate audio for the audio-embeddings: the Amazon Polly TTS service[1], and an Alexa-AI Internal text to speech (TTS) library optimized for synthetic ASR data, Multivoice-TTS [25]. For both TTS methods we use ten voices drawn from en-US and en-GB locales. Silence (0.1s) is inserted before and after each utterance.

Audio Embedding: While audio embeddings for the external catalog could be constructed in several ways, similar to Wu et al. [19], we aim to make audio-embeddings as close to on-policy self-attention embeddings as possible. Thus, we use the mean of the self-attention representations of the baseline model (no fine-tuning) at an intermediate layer, as audio embeddings. In this work, we always assume the existence of a suitable seed model from which the catalog audio embeddings, a_i, can be generated, and they are generated offline prior to training the model described in subsection 2.2. While training the model and catalog jointly would eliminate off-policy speech embeddings, such a method presents technical challenges.

Text Embedding: We explore several methods of generating the text embeddings (the "value" in the catalog K-V pairs). Initially, for small catalogs, we explored learned one-hot embeddings, however, while one-hot embeddings can lead to better performance (Table 1), they are not scalable – as they cannot be computed offline (and thus, cannot be inserted during test time). To generate scalable text embeddings, we explore two semantic text-embedding approaches: GLoVE embeddings [23], which are built using word co-occurrence probabilities, and BERT-style embeddings [24], which are learned from large statistical models. GLoVE embeddings are 300 dimensional, and computed using the publicly available vectors, and our BERT-style embeddings are computed using the `all-MiniLM-L6-v2` model in the `sentence-transformers` package [26].

2.2 External Memory Fusion

An overview of the external memory fusion process is given in Figure 2. The speech encoder in our proposed work is based on the Conformer encoder [27], augmented with additional K-Nearest-Neighbor (KNN) fusion layers. In each KNN fusion layer, for each audio frame embedding a_i of the utterance A, we query the external memory $E = (k_i, v_i), 1 \leq i \leq |E|$ for a set of m nearest neighbors:

$$\mathcal{N}_{a_i} = \underset{N \subset E, |N|=m}{\arg\min} \sum_{(k_j, v_j) \in N} ||k_j - a_i||_2^2 \quad (1)$$

We then construct the context for the layer as $\mathcal{C} = \cup_{a_i \in A} \mathcal{N}_{a_i}$. From $\mathcal{C}$ we can construct two matrices, $K_c \in \mathbb{R}^{m|A|, d_{\text{key}}}$ and $V_c \in \mathbb{R}^{m|A|, d_{\text{value}}}$, consisting of the keys and values respectively. The output of our K-NN fusion layer is then:

$$F(A, E) = A + \text{LN}\left(\text{ReLU}\left(\text{softmax}\left(\frac{(AW_q)K_c'}{\sqrt{d}}\right)(V_c W_v)\right)\right) \quad (2)$$

where LN is LayerNorm. Unfortunately, because we are working with large catalogs, the computation of Equation 1 can be very expensive. Thus, instead of computing the exact nearest neighbors, we rely on approximate nearest neighbors, which can be computed much more efficiently. To efficiently extract approximate nearest neighbors from our large-scale catalogs, we leverage the FAISS [28] library to generate Optimized Product-Quantization-transformed keys (64 dimensions) [29], which are searched using a Hierarchial Naviagable Small Worlds (HNSW) index with 2048 centroids encoded with product-quantized fast-scan [30]. Such an approach leads to only a 15% increase in forward-pass latency, even when running with catalogs with over 7M key/value pairs.

2.3 Experimental Design

Catalog Data Sources: In our work we explore several different catalog data sources. For Librispeech, we build a simulated catalog using the 2500 rarest tokens present in either the training or test datasets. Our internal Alexa catalog focuses on assistant queries in a media domain, and consists of 15K movie titles. In both cases, we build a unique catalog for training and testing, allowing us to explore how well the model performs under distribution shift of the catalog at test time.

ASR Base Model: Although in practice our method could be applied to many different speech encoders, we use the Conformer encoder [27]. For the decoder, we use a 1-layer LSTM decoder with 320 hidden dimensions. While we explore several encoder sizes, we primarily follow Gulati et al. [27] for Librispeech and use a 16-layer encoder with a hidden dimension of 144 (10.3M Params). For Alexa data, we use a conformer model with 208.37M parameters.

3 Results & Discussion

Librispeech: Our key results are shown for Librispeech in Table 1 for several choices of TTS, Text Embeddings, and NNs/Frame (K). We can see that overall, augmenting models with additional data leads to stronger performance than models without external data. For Librispeech, when training with the train catalog and testing with the test catalog, we get strong transfer performance, exceeding that of when we use the training catalog for both training and testing, suggesting additional zero-shot specialization. While 1-hot vectors outperform BERT vectors, we must train these vectors for each catalog, leading to an inability to do test-time specialization. BERT outperforms GLoVE in all cases (with GLoVE causing regressions on test-time specialization). Figure 4 demonstrates that our method can capture and apply domain data from the catalogs. In this experiment, the model is trained with a cata-

[1] https://aws.amazon.com/polly/

Table 1: Word Error Rate on Librispeech data with a small (10.3M param) model. MV-TTS refers to Multivoice-TTS.

Catalog	TTS	Text	K	test-clean	test-other
Baseline				5.77	13.34
Train	Polly	1-Hot	4	5.75 (0.34%)	13.30 (0.29%)
	Polly	1-Hot	8	5.72 (0.86%)	13.19 (1.10%)
	Polly	1-Hot	16	5.71 (1.03%)	13.15 (1.42%)
	Polly	BERT	8	5.74 (0.52%)	13.26 (0.60%)
	MV-TTS	1-Hot	8	5.52 (4.33%)	12.96 (2.84%)
	MV-TTS	BERT	8	5.68 (1.63%)	13.05 (2.18%)
Test	Polly	GLoVE	8	6.33 (-8.84%)	14.56 (-9.15%)
	Polly	BERT	8	5.71 (1.03%)	13.24 (0.75%)
	MV-TTS	GLoVE	8	6.15 (-6.17%)	14.32 (-6.84%)
	MV-TTS	BERT	8	**5.34** (8.05%)	**12.84** (3.86%)

Table 2: Librispeech test-set Relative WER *improvement* for models augmented with catalog data in different layers.

Dataset	1	3	12	16	3,12	all
clean	1.02%	3.65%	6.65%	2.63%	7.79%	8.05%
other	0.71%	2.88%	2.97%	1.08%	3.41%	3.86%

log containing 300K training-set unique bigrams, and we show the performance of this model using ten test catalogs, each consisting of 30K bigrams, taken either from the test set or dev set. As the fraction of bigrams in the test data that are available in the test catalog increases, the performance of the model improves – showing our approach can use the information in test catalogs effectively in a zero-shot learning setup.

We also run several ablations with Multivoice-TTS, BERT Embeddings and 8 NNs/Frame. Table 2 explores the performance of our approach when placing the external knowledge augmentation at different layers of the network. While using external knowledge in all layers is the most effective approach, we find that such an approach is latency-prohibitive, as it increases the latency of a forward pass of the model by ≈85%. Using a single layer increases latency by only ≈15%, while two layers increase latency by ≈23%. Table 3 explores the performance of the method on Librispeech for differing model sizes. As the number of parameters increases, the gains provided by external memory decrease.

Alexa: To further validate our method, we additionally explore a real-world simulation of our model's ability to generalize to test data. We started with a seed model B, and trained two derived models: B_{FT}, fine-tuned on both the TTS Catalog for Alexa ($\mathcal{C}$, section 2) and an additional 120K hours of de-

Table 3: Librispeech test-set Relative WER *improvement* over baseline fine-tuning using differing model sizes (M: Millions of params).

Dataset	5M	10M	50M	100M	300M
clean	28.9%	8.05%	4.28%	1.66%	0.08%
other	19.3%	3.86%	2.65%	-0.07%	0.01%

Fig. 4: Librispeech test-clean WER vs. test catalog/data overlap.

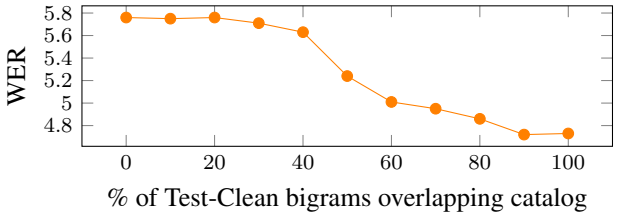

Table 4: Performance on Alexa data. T-C: Time for Catalog Generation. T-FT: Time for fine-tuning. WER-I: Relative word-error rate *improvement*. Multivoice-TTS, BERT, and 8 NNs/Frame.

Model/Test Data	T-C (min)	T-FT (GPU-Hours)	rel. WER-I
B_{FT}/Test-TTS	0	2048	7.1%
B_{cat}/Test-TTS	33	1600	6.8%
B_{FT}/Alexa Test	-	-	0.52%
B_{cat}/Alexa Test	-	-	4.12%
$B_{\text{FT}+T}$/Alexa Test	0	1024	19.66%
$B_{\text{cat}+T}$/Alexa Test	28	0	21.27%

identified Alexa data, $\mathcal{D}$, and B_{cat}, which trains the proposed method on $\mathcal{D}$, with catalog $\mathcal{C}$. In both cases, the full model (the speech encoder, and if applicable, the fusion model) are fine-tuned. The results (Table 4, rows 1/2) demonstrate that even with significantly fewer GPU hours, our approach achieves similar WER. We then transfer both models to the test dataset (consisting of real speech) without additional tuning. We see in Table 4 (rows 3/4) that our trained model achieves better performance, suggesting that the model has learned to generalize better than the model trained with fine-tuning alone.

Finally, we update our fine-tuned and catalog models to include the test data, T. Test data is incorporated into the fine-tuned model through GPU-based training, while the test data is incorporated into the catalog model through catalog generation/concatenation. Table 4 (rows 5/6) demonstrates that even with *no additional GPU training* our approach ($B_{\text{cat}+T}$) achieves similar performance to fine-tuning ($B_{\text{FT}+T}$).

4 Conclusion

This paper introduces the first approach for large-scale contextualization of speech-encoder representations using text-only catalog data. We strongly believe that our method represents a promising step forward for ensuring the recognition of rare words and efficient transfer novel test-time distributions. While this paper is a first step towards contextualized speech encoders, many directions for future work remain including investigating embeddings for the catalogs (such as grapheme/phoneme embeddings), exploring other languages and word pronunciations, and understanding the performance in larger-scale rapidly changing real-world distributions.

5 References

[1] S. Dingliwa, A. Shenoy, S. Bodapati, A. Gandhe, R. T. Gadde, and K. Kirchhoff, "Domain prompts: Towards memory and compute efficient domain adaptation of asr systems," in *Interspeech 2022*, 2022.

[2] D. Baby, P. D'Alterio, and V. Mendelev, "Incremental learning for rnn-transducer based speech recognition models," in *Interspeech 2022*, 2022.

[3] K. Zhao, H. D. Nguyen, A. Jain, N. Susanj, A. Mouchtaris, L. Gupta, and M. Zhao, "Knowledge distillation via module replacing for automatic speech recognition with recurrent neural network transducer," in *Interspeech 2022*, 2022.

[4] A. Gandhe, A. Rastrow, and B. Hoffmeister, "Scalable language model adaptation for spoken dialogue systems," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 907–912.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *NeruIPS*, vol. 30, 2017.

[6] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, "Joint grapheme and phoneme embeddings for contextual end-to-end asr." in *Interspeech*, 2019, pp. 3490–3494.

[7] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8537–8541.

[8] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware transformer transducer for speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 503–510.

[9] S. Novotney, S. Mukherjee, Z. Ahmed, and A. Stolcke, "Cue vectors: Modular training of language models conditioned on diverse contextual signals," *arXiv:2203.08774*, 2022.

[10] A. Shenoy, S. Bodapati, and K. Kirchhoff, "Contextual biasing of language models for speech recognition in goal-oriented conversational agents," *arXiv preprint arXiv:2103.10325*, 2021.

[11] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-fusion end-to-end contextual biasing." in *Interspeech*, 2019, pp. 1418–1422.

[12] B. Liu and I. Lane, "Dialog context language modeling with recurrent neural networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5715–5719.

[13] A. Jaech and M. Ostendorf, "Personalized language model for query auto-completion," *arXiv:1804.09661*, 2018.

[14] S. Kim and F. Metze, "Dialog-context aware end-to-end speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 434–440.

[15] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li, "Hierarchical recurrent neural network for document modeling," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 899–907.

[16] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search." in *Interspeech*, 2018, pp. 2227–2231.

[17] T. Munkhdalai, K. C. Sim, A. Chandorkar, F. Gao, M. Chua, T. Strohman, and F. Beaufays, "Fast contextual adaptation with neural associative memory for on-device personalized speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6632–6636.

[18] S. Borgeaud, A. Mensch *et al.*, "Improving language models by retrieving from trillions of tokens," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2206–2240.

[19] Y. Wu, M. N. Rabe, D. Hutchins, and C. Szegedy, "Memorizing transformers," *arXiv preprint arXiv:2203.08913*, 2022.

[20] U. Khandelwal, A. Fan, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Nearest neighbor machine translation," *arXiv preprint arXiv:2010.00710*, 2020.

[21] A. Goyal, A. Friesen, A. Banino, T. Weber, N. R. Ke, A. P. Badia, A. Guez, M. Mirza, P. C. Humphreys, K. Konyushova *et al.*, "Retrieval-augmented reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022.

[22] Z. Tang, S. Gu, J. Bao, D. Chen, and F. Wen, "Improved vector quantized diffusion models," *arXiv:2205.16007*, 2022.

[23] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] I. Vallés-Pérez, J. Roth, G. Beringer, R. Barra-Chicote, and J. Droppo, "Improving multi-speaker tts prosody variance with a residual encoder and normalizing flow," in *Interspeech 2021*, 2021.

[26] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," in *NAACL 2021*, Jun. 2021, pp. 296–310.

[27] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[28] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[29] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 4, pp. 744–755, 2013.

[30] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 824–836, 2018.