POST-TRAINING EMBEDDING ALIGNMENT FOR DECOUPLING ENROLLMENT AND RUNTIME SPEAKER RECOGNITION MODELS

Chenyang Gao[⋆] *Brecht Desplanques*[†] *Chelsea J.-T. Ju*[†] *Aman Chadha*[†] *Andreas Stolcke*[†]

[⋆]Rutgers, The State University of New Jersey
[†]Amazon Alexa AI, U.S.A.

ABSTRACT

Automated speaker identification (SID) is a crucial step for the personalization of a wide range of speech-enabled services. Typical SID systems use a symmetric enrollment-verification framework with a single model to derive embeddings both offline for voice profiles extracted from enrollment utterances, and online from runtime utterances. Due to the distinct circumstances of enrollment and runtime, such as different computation and latency constraints, several applications would benefit from an asymmetric enrollment-verification framework that uses different models for enrollment and runtime embedding generation. To support this asymmetric SID where each of the two models can be updated independently, we propose using a lightweight neural network to map the embeddings from the two independent models to a shared speaker embedding space. Our results show that this approach significantly outperforms cosine scoring in a shared speaker logit space for models that were trained with a contrastive loss on large datasets with many speaker identities. This proposed Neural Embedding Speaker Space Alignment (NESSA) combined with an asymmetric update of only one of the models delivers at least 60% of the performance gain achieved by updating both models in the standard symmetric SID approach.

Index Terms— Speaker verification, embedding space alignment, asymmetric speaker recognition

1. INTRODUCTION

Speaker Identification (SID) systems are developed to recognize speakers by comparing their distinctive vocal characteristics. Current online SID systems extract speaker embeddings in real-time fashion from the incoming audio streams and perform speaker identification by comparing these embeddings against existing voice profiles [1, 2, 3]. The voice profiles are created by averaging the embeddings across the registered utterances for each speaker. These systems utilize the same speaker embedding extractor during both the enrollment and verification stage. In the remainder of this paper, we will refer to this approach as the standard *symmetric enrollment-verification framework*. However, recent research [4] has shed light on the potential of using different SID models for generating embeddings in each stage. This approach is referred to as an *asymmetric enrollment-verification framework*. It eliminates the need to use the same model during the distinct stages of enrollment and verification and it leads to many potential practical applications. The key idea is to use *embedding space alignment* to reduce the mismatch between embedding spaces originating from different SID models to enable direct embedding comparison.

This alignment opens up a range of potential applications. For example, Li *et al.* [4] proposed to use asymmetric SID involving a larger model for generating embeddings during enrollment and a smaller model for embedding extraction from the runtime audio streams. During enrollment, a computationally intensive and non-causal model can be used to extract high-quality voice profiles, while the runtime model should exhibit minimal latency and computational cost. Another pertinent application involves an industry-specific challenge. To comprehensively validate SID model performance in the real world and to compare the impact of different SID models, extensive A/B [5] or A/B/n tests become a fundamental part of the evaluation pipeline. However, each candidate model in the standard symmetric enrollment-verification framework will require an updated voice profile for a vast set of enrolled speakers. This poses a scaling issue when multiple model candidates are tested in parallel. This standard A/B/n test setup will also result in computation wasted on the creation of new voice profiles when some model candidates are eventually not being used. To overcome these inefficiencies, speaker embedding space alignment enables us to utilize the readily available voice profiles and to make them compatible with the candidate models, instead of creating new voice profiles for each candidate model. Moreover, SID model updates would potentially impact downstream applications that rely on the generated speaker embeddings to provide extra speaker identity context. Embedding alignment would provide a path to updating the SID models, without significantly impacting the downstream applications by feeding those dependent systems the embeddings that have been aligned back to the original speaker embedding space.

Prior work in the speaker verification domain utilized a shared speaker logit score space to combine embeddings from different models to create a high-performing system ensemble [6, 7, 8]. This alignment depends on utterance-based score vectors containing the speaker similarity score against every individual training speaker in a large shared dataset that was used to train every individual system in the ensemble with a softmax-based classification loss. Even though the speaker logit score vectors can be produced by different SID systems, these score vectors can be directly compared through cosine similarity scoring, as the training speaker set is identical across the systems. In certain cases, cosine scoring in the speaker logit space can outperform cosine scoring in the speaker embedding space. It has also been shown that system fusion in this logit space outperformed the more standard score fusion [7, 8]. However, the effectiveness of this scoring method remains uncertain when the SID models are trained with training objectives other than the typical softmax-based classification loss [9]. Classification-based loss functions are typically avoided when the number of training speakers becomes unmanageable for the classification head; in those cases one typically relies instead on scalable contrastive loss variants [10, 2, 3]. In another related study [4], an auxiliary loss was introduced to align speaker embedding spaces for various models during the training process. In [11], researchers proposed to

Chenyang Gao performed this work while interning at Amazon.

use knowledge distillation to transfer the knowledge from a teacher model to a student model. While these methods alter the speaker embedding spaces to be aligned, they limit the flexibility of developing a new runtime and/or enrollment speaker embedding extractor completely independent from each other, since the alignment happens during training of the embedding extractor.

To account for the diversity of possible SID models and to allow for the models to be trained independently, we propose a flexible and lightweight Neural Embedding Speaker Space Alignment (NESSA) backend to align the speaker embeddings between frozen enrollment and runtime embedding extractors. In the context of datasets with a very large numbers of speakers, our results showed that speaker-logit-based alignment did not yield satisfactory results in the asymmetric enrollment-verification framework when the models were trained with different training objectives, speaker sets, and model structures. NESSA on the other hand performed significantly better and can in certain scenarios completely close the performance gap when compared to a more costly update of both models in the standard symmetric enrollment-verification framework.

2. EFFICIENT EMBEDDING SPACE ALIGNMENT

2.1. Problem statement

Consider two independently trained SID models denoted as model ***X*** and ***Y***, and corresponding speaker embedding spaces $\mathbb{E}_X$ and $\mathbb{E}_Y$, respectively. The goal is to conduct asymmetric speaker verification given the enrollment embeddings in $\mathbb{E}_X$ and the runtime embeddings in $\mathbb{E}_Y$. Since ***X*** and ***Y*** are trained with different configurations or model architectures, for the various reasons described in Section 1, $\mathbb{E}_X$ and $\mathbb{E}_Y$ are mismatched. The immediate task is to develop a space alignment approach that enables performant scoring in the asymmetric framework, without degrading the performance compared to the single-model symmetric system with the worst-performing model and to close the performance gap compared to the symmetric approach with the best-performing model.

2.2. Speaker-logit-based embedding space alignment

Speaker verification in the speaker logit space involves cosine scoring between score vectors that express the speaker similarity of an utterance against every individual speaker within a predefined set of speakers. These speaker similarities are typically estimated on the speakers that were used to train the embedding extractor. Thus, the embeddings of different model versions can be made compatible by calculating a lightweight mapping between the different speaker embedding spaces to the speaker logit score vectors based on a shared pool of training speakers [6, 7, 8]. When both systems are trained with a classification-based loss, the speaker logits $\mathbf{l}$ refer to the high-dimensional last layer output of the model that is used as input for the softmax-based classification loss. They are computed as $\mathbf{l} = \mathbf{W}\mathbf{r}$, where $\mathbf{r}$ is the speaker embedding, and $\mathbf{W}$ is the classification weight matrix with shape $(N \times d)$ that defines the classification head. N and d denote the number of speakers in the training set and the embedding dimension, respectively. The final classification layer does not typically include a bias term [9].

However, models that are trained with other training criteria such as contrastive losses [2] or binary cross entropy [10] do not have such a classification weight matrix. To enable speaker logit scoring in this case, we construct a classification weight matrix post-training using voice profiles: $\mathbf{W} = [\mathbf{e}_1; \mathbf{e}_2; \ldots; \mathbf{e}_N]^T$. Voice profile $\mathbf{e}_i$ indicates the length-normalized average enrollment embedding for speaker i, and N is the number of selected speakers to construct $\mathbf{W}$. We perform speaker verification using cosine similarity scoring s_c for different models in the speaker logit space as follows:

$$\begin{aligned} s_c(\mathbf{l}_e, \mathbf{l}_r) &= \frac{\mathbf{l}_e^T \mathbf{l}_r}{||\mathbf{l}_e|| \cdot ||\mathbf{l}_r||} \\ &= \frac{\mathbf{e}_X^T \mathbf{W}_X^T \mathbf{W}_Y \mathbf{r}_Y}{||\mathbf{W}_X \mathbf{e}_X|| \cdot ||\mathbf{W}_Y \mathbf{r}_Y||} \\ &= \frac{\mathbf{e}_X^T \mathbf{W}_X^T \mathbf{W}_Y \mathbf{r}_Y}{\sqrt{\mathbf{e}_X^T \mathbf{W}_X^T \mathbf{W}_X \mathbf{e}_X} \cdot \sqrt{\mathbf{r}_Y^T \mathbf{W}_Y^T \mathbf{W}_Y \mathbf{r}_Y}} \end{aligned}$$

where $\mathbf{l}_e$ and $\mathbf{l}_r$ are the speaker logit score vectors for the enrollment profile and runtime embedding, respectively. Matrices $\mathbf{W}_X$, $\mathbf{W}_Y$ represent the classification weights using a shared set of speakers for models ***X*** and ***Y***. Embedding $\mathbf{e}_X$ is the enrollment voice profile generated by model ***X*** and $\mathbf{r}_Y$ is the runtime speaker embedding extracted by model ***Y***.

To make this scoring approach more efficient, we used the Cholesky decomposition and the fusion approach as described in [6]:

$$\begin{aligned} s_c(\mathbf{l}_e, \mathbf{l}_r) &= \frac{\tilde{\mathbf{e}}_X^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \tilde{\mathbf{r}}_Y}{\sqrt{\tilde{\mathbf{e}}_X^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \tilde{\mathbf{e}}_X} \cdot \sqrt{\tilde{\mathbf{r}}_Y^T \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \tilde{\mathbf{r}}_Y}} \\ &= \frac{\tilde{\mathbf{e}}_X^T \tilde{\mathbf{M}}^T \tilde{\mathbf{M}} \tilde{\mathbf{r}}_Y}{\sqrt{\tilde{\mathbf{e}}_X^T \tilde{\mathbf{M}}^T \tilde{\mathbf{M}} \tilde{\mathbf{e}}_X} \cdot \sqrt{\tilde{\mathbf{r}}_Y^T \tilde{\mathbf{M}}^T \tilde{\mathbf{M}} \tilde{\mathbf{r}}_Y}} \\ &= s_c(\tilde{\mathbf{M}} \tilde{\mathbf{e}}_X, \tilde{\mathbf{M}} \tilde{\mathbf{r}}_Y) \end{aligned}$$

where $\tilde{\mathbf{W}} = [\mathbf{W}_X; \mathbf{W}_Y]$ with shape $N \times 2d$, $\tilde{\mathbf{e}}_X^T = [\mathbf{e}_X^T; \mathbf{0}]$, and $\tilde{\mathbf{r}}_Y^T = [\mathbf{0}; \mathbf{r}_Y^T]$. $\tilde{\mathbf{M}}$ is an upper triangular matrix with dimensions $2d \times 2d$ such that $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}} = \tilde{\mathbf{W}}^T \tilde{\mathbf{W}}$. This approach allows for efficient speaker logit scoring that does not negatively scale with the number of training speakers N in $\mathbf{W}$ during inference.

2.3. Neural Embedding Speaker Space Alignment

Instead of performing the speaker similarity scoring in a shared speaker logit space, we propose to use a Neural Embedding Speaker Space Alignment (NESSA) that employs a lightweight DNN $\mathscr{F}$ to enable accurate cosine similarity scoring in the asymmetric enrollment-verification framework. In contrast to the approach of jointly training both models in the asymmetric framework as proposed in [4], our proposal involves computing this space alignment after the training of each individual model. As before, we assume that the length-normalized enrollment voice profile $\mathbf{e}_X^i$ from speaker i is produced by model ***X***, and the length normalized runtime embedding $\mathbf{r}_Y^j$ from speaker j is generated by model ***Y***. We explore three different approaches to train NESSA:

Scoring in embedding space *X* ($\mathscr{M}_1$): In this approach, we use the embedding space of model ***X*** as a reference space, and train a space aligner $\mathscr{F}$ to map runtime embeddings $\mathbf{r}_Y$ to model space ***X***, so as to perform the verification in that embedding space. The training objective is:

$$\mathscr{L} = \frac{1}{N} \sum_i^N \text{MSE}(\mathscr{F}(\mathbf{r}_Y^i), \mathbf{r}_X^i) \tag{1}$$

where we use the mean squared error (MSE) as the training objective and N is the number of embeddings in each minibatch. During evaluation, we will perform cosine scoring between voice profile $\mathbf{e}_X$ and runtime embedding $\mathscr{F}(\mathbf{r}_Y)$ in embedding space ***X***.

Scoring in embedding space *Y* ($\mathscr{M}_2$): Alternatively, we can perform this mapping the other way around by mapping the enrollment embeddings from space ***X*** to space ***Y***, using loss

$$\mathscr{L} = \frac{1}{N} \sum_i^N \text{MSE}(\mathscr{F}(\mathbf{e}_X^i), \mathbf{e}_Y^i). \tag{2}$$

We then perform verification between $\mathscr{F}(\mathbf{e}_X)$ and $\mathbf{r}_Y$ in embedding space $\boldsymbol{Y}$. An advantage of this approach is that the mapping of enrollment embeddings can be performed offline, which would completely eliminate the impact of NESSA on runtime latency.

Boosting NESSA with contrastive learning ($\mathscr{M}_3$): The original embedding spaces $\boldsymbol{X}$ or $\boldsymbol{Y}$ might not be the most suited spaces to compare the embeddings between two very different SID models. To further increase the impact of NESSA, we propose to adapt both the enrollment and runtime embeddings simultaneously to a new embedding space with two DNNs $\mathscr{F}_1$ and $\mathscr{F}_2$ respectively. We introduce an additional contrastive loss term as in [4] to make this new embedding space suitable for speaker verification purposes. As the NESSA backend will typically be trained on smaller datasets compared to the training dataset of the embedding extractors, we will still anchor the new embedding space to the embedding space of the best-performing model (here assumed to be model $\boldsymbol{Y}$) using MSE loss terms for both enrollment and runtime embeddings similar to Eq. (2). The final loss function is defined as follows:

$$\begin{aligned}\mathscr{L} = &- \alpha \frac{1}{N} \sum_{i}^{N} \log \frac{e^{w \cdot s_c(\mathscr{F}_1(\mathbf{e}_X^i), \mathscr{F}_2(\mathbf{r}_Y^i))}}{\sum_{j}^{N+M} e^{w \cdot s_c(\mathscr{F}_1(\mathbf{e}_X^j), \mathscr{F}_2(\mathbf{r}_Y^i))}} \\ &+ \beta \frac{1}{N} \sum_{i}^{N} \text{MSE}(\mathscr{F}_1(\mathbf{e}_X^i), \mathbf{e}_Y^i) \\ &+ \gamma \frac{1}{N} \sum_{i}^{N} \text{MSE}(\mathscr{F}_2(\mathbf{r}_Y^i), \mathbf{r}_Y^i) \end{aligned} \tag{3}$$

where α, β, γ are scalars to control the importance of the loss function terms and w is a trainable parameter to rescale the range of cosine similarity s_c. We will set a relatively low value for γ as the corresponding loss term acts as a regularization penalty and does not help with learning a proper alignment between embedding spaces. Previous studies [12] showed that increasing the number of negative samples in contrastive learning leads to more discriminative representations. We increase the original number of negative samples in the contrastive loss term (limited by the batch size N) by adding M additional distinct voice profiles.

3. EXPERIMENTAL SETUP

3.1. Enabling quick A/B tests without voice profile updates

We will use A/B testing as a case study. We will assume that candidate model $\boldsymbol{Y}$ outperforms the reference model $\boldsymbol{X}$ during offline evaluation. We have the existing voice profiles generated by model $\boldsymbol{X}$ and we want to enable cosine scoring with runtime embeddings extracted by the better model $\boldsymbol{Y}$ against the existing voice profiles through embedding alignment, instead of updating the voice profiles.

3.2. Datasets for embedding alignment and SID evaluation

Training and evaluation is conducted on de-identified voice assistant speech data with consent of the speakers. To construct the training dataset for embedding space alignment, we apply an existing speaker recognition model to the data and build positive speaker/utterance pairs based on high speaker similarity scores. This process results in a dataset with 200K speakers including both enrollment and runtime utterances. Within the training dataset, instances from 10% of the speakers serve as validation data for model selection and hyperparameter tuning. The dataset for evaluating the SID systems is constructed by first randomly sampling de-identified utterances. The sampled utterances, together with the enrollment data of speakers associated with the same group of speakers, are compared by multiple annotators to create the speaker labels. We only keep utterances with consistent annotation labels. To evaluate the generalization capability of the trained models there is no group overlap between the training datasets and the evaluation data, but the alignment training dataset and evaluation datasets are sampled from the same in-domain distribution.

3.3. Asymmetric SID systems

To assess the effectiveness of the space alignment methods for varying performance progress, we select four SID models to construct two main asymmetric SID systems. The SID systems employ a multi-layer LSTM architecture [13, 2] with projection layers. Each LSTM layer has 1200 nodes, and 400 nodes in the projection layer. The output speaker embedding size is 400. The acoustic input features are 40-dimensional log Mel-filter bank energies with a Hamming window of 25 ms and a step size of 10 ms for all models. These features are passed through an energy-based voice activity detection module to remove the non-speech frames. The four models are:

- **GE2E**: A 3-layer LSTM architecture trained using the generalized-end-to-end (GE2E) loss in the default configuration from [2]. It was trained on a large internal voice assistant dataset, that is significantly larger than the embedding space alignment training datasets.
- **BCE**: A 4-layer LSTM architecture trained with the binary cross-entropy (BCE) loss [10] on a second large internal dataset of the same scale as used for the **GE2E** model.
- **SA*early***: A model that uses a 3-layer LSTM architecture trained with the GE2E loss on the space alignment (SA) training dataset with early stopping.
- **SA*full***: Similar to **SA*early*** but trained until full convergence and initialized with a different random seed.

We define two asymmetric SID systems, each uniquely defined by their enrollment-verification model versions:

- **GE2E/BCE** enrollment-verification: The voice profiles are extracted by the **GE2E** model, while the runtime embeddings are generated by the **BCE** model. The goal is to evaluate embedding space alignment when both models have similar performance.
- **SA*early*/SA*full*** enrollment-verification: The voice profiles are extracted by the **SA*early*** model, while the runtime embeddings are generated by the **SA*full*** model. The main goal is to evaluate embedding space alignment when there is a large performance gap between the two models.

3.4. Embedding space alignment configuration

The speaker logit alignment weight matrix $\mathbf{W}$ is constructed from voice profiles generated by the enrollment embedding extractor for a varying number of speakers in the alignment training dataset. For example, $\tilde{\mathbf{M}}_{1\text{K}}$ indicates we are using 1000 voice profiles to construct $\tilde{\mathbf{W}}$ before executing the Cholesky decomposition.

The lightweight model architecture of NESSA is a 3-layer multilayer perceptron (MLP) with ReLU activations [14]; the hidden size of the MLP is set to 800. The output embeddings are 400-dimensional. Each model is trained for 50 epochs with 2000 training steps per epoch; the batch size is set to 1024. We used the Adam [15] optimizer with an initial learning rate of 10^{-3}, with an exponential learning rate decay with a ratio of 0.96 after every epoch. The weights in the loss function for NESSA with contrastive learning ($\mathscr{M}_3$) are set to $\alpha = 1.0, \beta = 0.5, \gamma = 0.1$, w is initialized to 5.

4. RESULTS AND ANALYSIS

4.1. Baseline results for symmetric enrollment-verification

Baseline experiments involving a symmetric enrollment-verification framework are shown in the top rows of Table 1. For all experiments

Table 1. Relative False Reject Rate (FRR) impact in % of symmetric and asymmetric speaker verification at different fixed False Accept Rate (FAR) target values on an in-house evaluation dataset following the evaluation protocol described in [16]. Higher relative FRR impact is better and 0% impact indicates the baseline single-model symmetric systems.

Embedding Alignment Approach	Enrollment/Verification Model	Relative FRR impact @ target FAR (%) (↑) @12.5%FAR	@5.0%FAR	@2.0%FAR	Enrollment/Verification Model	Relative FRR impact @ target FAR (%) (↑) @12.5%FAR	@5.0%FAR	@2.0%FAR
×	GE2E/GE2E	0	0	0	SA_{early}/SA_{early}	0	0	0
×	BCE/BCE	11.08	8.55	5.35	SA_{full}/SA_{full}	63.62	62.67	58.75
speaker logits $\tilde{M}_{200K}$	GE2E/GE2E	−198.92	−190.56	−198.15	SA_{early}/SA_{early}	−36.75	−20.44	−13.65
speaker logits $\tilde{M}_{200K}$	BCE/BCE	−163.24	−194.99	−198.15	SA_{full}/SA_{full}	21	26.19	25.05
speaker logits $\tilde{M}_{1K}$	GE2E/BCE	−514.59	−497.2	−464.79	SA_{early}/SA_{full}	−69.31	−70.51	−73.15
speaker logits $\tilde{M}_{10K}$	GE2E/BCE	−517.03	−496.61	−458.56	SA_{early}/SA_{full}	−62.88	−65.79	−71.22
speaker logits $\tilde{M}_{200K}$	GE2E/BCE	−504.86	−488.94	−461.19	SA_{early}/SA_{full}	−64.19	−67.03	−72.38
NESSA $\mathscr{M}_1$	GE2E/BCE	−1.35	−2.36	−3.5	SA_{early}/SA_{full}	6.94	4.26	4.62
NESSA $\mathscr{M}_2$	GE2E/BCE	5.95	4.13	1.46	SA_{early}/SA_{full}	37.56	36.12	32.2
NESSA $\mathscr{M}_3$ ($M = 50K$)	GE2E/BCE	11.35	11.5	7.3	SA_{early}/SA_{full}	43.62	40.88	35.48

we will report the relative false reject rate (FRR) changes at fixed target values of the false accept rate (FAR) [16] against **GE2E** and **SA*early*** baselines. As expected asymmetric enrollment-verification without embedding space alignment did not perform significantly better than random scoring due to the mismatch of the embedding spaces, hence these results are not included.

4.2. Speaker-logit-based embedding space alignment

We present the speaker-logit-based embedding space alignment in the middle section of Table 1. Speaker logit alignment enhances the results of the asymmetric framework compared to having no alignment at all. However, a six-fold FRR increase (around -500%) is observed against a strong **GE2E** baseline. Additionally, increasing the number of alignment speakers only improves the performance marginally. Table 1 also includes symmetric **GE2E/GE2E** and **SA*early*/SA*early*** speaker logit scoring. We observe that symmetric speaker-logit scoring triples the FRR (around -200%) when compared to the **GE2E** baseline that uses standard speaker embedding scoring which somewhat contradicts previous studies [7, 6]. The degradation for symmetric speaker logit scoring with **SA*early*** is less pronounced (-10% to -35%), indicating it is important that the embedding extractors are trained on the same set of speakers as those used for speaker logit scoring, which significantly limits the flexibility of the alignment method. Most likely this performance gap can be further decreased by using classification-based losses to train the embedding extractors as proposed in [7, 6], however these types of losses cannot be directly applied to datasets with a large number of speakers, due to scaling issues.

4.3. Neural Embedding Speaker Space Alignment

The results with NESSA are presented in the bottom part of Table 1. We observe the following. First, all NESSA approaches perform significantly better than the speaker logit scoring method, demonstrating the effectiveness of training a post-training space embedding aligner using neural network techniques. Second, $\mathscr{M}_2$ enrollment embedding alignment to the model candidate embedding space leads to significantly better results than $\mathscr{M}_1$ alignment to the original runtime embedding space. This is somewhat expected as the candidate model ***Y*** has better speaker verification performance, which should correspond to a higher-quality speaker embedding space; it should be the preferred target space for alignment. The performance of $\mathscr{M}_2$ is in between the performance of symmetric **GE2E** and **BCE**, showing that asymmetric framework with space alignment can benefit from updating a model on only a single side of the speaker verification trial. Third, alignment $\mathscr{M}_3$ outperforms all other alignment methods. When the baseline and candidate model performance are comparable, as is the case for the **GE2E** and **BCE** models, $\mathscr{M}_3$ alignment can even slightly outperform the **BCE** candidate model in the symmetric framework. We argue this is because the alignment training data and the evaluation data are sampled from the same specific domain, and thus embedding alignment can perform (partial) fine-tuning. When the performance difference between the baseline and candidate models is large and the embedding extractors are trained on the same in-domain data (**SA*early*** vs. **SA*full***), $\mathscr{M}_3$ can achieve up to 60% of the performance improvement achieved by the candidate model in the symmetric framework.

Table 2. Ablation study for $\mathscr{M}_3$, on α, β and γ and number of additional speakers M

Alignment Approach	Enrollment/Verification	12.5%	5.0%	2.0%
×	GE2E/GE2E	0	0	0
$\mathscr{M}_3$ ($M = 50K$)	GE2E/BCE	11.35	11.5	7.3
$\mathscr{M}_3$ ($\alpha = 0$)	GE2E/BCE	7.57	4.57	−2.82
$\mathscr{M}_3$ ($\beta = 0, \gamma = 0$)	GE2E/BCE	−65.14	−41	−26.46
$\mathscr{M}_3$ ($M = 0$)	GE2E/BCE	12.16	6.93	4.18
$\mathscr{M}_3$ ($M = 10K$)	GE2E/BCE	13.24	10.77	8.56
×	SA_{early}/SA_{early}	0	0	0
$\mathscr{M}_3$ ($\beta = 0, \gamma = 0$)	SA_{early}/SA_{full}	14.06	26.12	28.12

Finally, we present an ablation study of $\mathscr{M}_3$ in Table 2 with the following findings. First, when excluding the effect of the contrastive loss by setting $\alpha = 0$, the performance can already slightly improve over $\mathscr{M}_2$. This illustrates the benefit of realigning both spaces. Second, training an entirely new space by setting $\beta = 0$ and $\gamma = 0$ resulted in significantly worse performance. This highlights the importance of selecting a strong reference embedding space. We hypothesize that this caused by the fact that **GE2E** and **BCE** were already trained on a larger-scale dataset with only the SID task in mind. The construction of the new shared space is based on a smaller alignment dataset, which is detrimental for final SID performance. However, when there are significant performance differences between models due to a weaker **SA*early*** model as in the last row of Table 2, the construction of a new space can perform better than the symmetric baseline. But the performance is still worse compared to utilizing a reference space in $\mathscr{M}_3$ or $\mathscr{M}_2$ alignment.

5. CONCLUSION

We have investigated post-training speaker embedding space alignment for SID systems within an asymmetric enrollment-verification framework, where different models are used to generate voice profiles and runtime speaker embeddings. A case study in enabling A/B tests within this asymmetric framework, so as to avoid extensive voice profiles rebuilding for each new candidate model, showed a need for embedding alignment. Our proposed NESSA method effectively bridges the mismatch between different embedding spaces, so that between 60% and 100% of the potential gain from the candidate model is achievable without explicit voice profile updates.

6. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-Vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[2] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[3] Jason Pelecanos, Quan Wang, and Ignacio Lopez Moreno, "Dr-Vectors: Decision Residual Networks and an Improved Loss for Speaker Recognition," in *Proc. Interspeech*, 2021, pp. 4603–4607.

[4] Qingjian Li, Lin Yang, Xuyang Wang, Xiaoyi Qin, Junjie Wang, and Ming Li, "Towards lightweight applications: Asymmetric enroll-verify structure for speaker verification," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7067–7071.

[5] Dan Siroker and Pete Koomen, *A/B testing: The most powerful way to turn clicks into customers*, John Wiley & Sons, 2015.

[6] Anna Silnova, Themos Stafylakis, Ladislav Mošner, Oldřich Plchot, Johan Rohdin, Pavel Matějka, Lukáš Burget, Ondřej Glembek, and Niko Brummer, "Analyzing Speaker Verification Embedding Extractors and Back-Ends Under Language and Channel Mismatch," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 9–16.

[7] Galina Lavrentyeva, Sergey Novoselov, Vladimir Volokhov, Anastasia Avdeeva, Aleksei Gusev, Alisa Vinogradova, Igor Korsunov, Alexander Kozlov, Timur Pekhovsky, Andrey Shulipa, Evgeny Smirnov, and Vasily Galyuk, "STC Speaker Recognition System for the NIST SRE 2021," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 354–361.

[8] Sergey Novoselov, Vladimir Volokhov, and Galina Lavrentyeva, "Universal speaker recognition encoders for different speech segments duration," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[10] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh, "SphereFace2: Binary classification is all you need for deep face recognition," in *International Conference on Learning Representations*, 2022.

[11] Shuai Wang, Yexin Yang, Tianzhe Wang, Yanmin Qian, and Kai Yu, "Knowledge distillation for small foot-print deep speaker embedding," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6021–6025.

[12] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency, "Self-supervised learning from a multi-view perspective," in *International Conference on Learning Representations*, 2021.

[13] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Abien Fred Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[15] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[16] Ruirui Li, Chelsea J.-T. Ju, Zeya Chen, Hongda Mao, Oguz Elibol, and Andreas Stolcke, "Fusion of Embeddings Networks for Robust Combination of Text Dependent and Independent Speaker Recognition," in *Proc. Interspeech 2021*, 2021, pp. 4593–4597.