Semi-supervised classification by reaching consensus among modalities

Anonymous Author(s)
Affiliation
Address
email

Abstract

We extend the Consensus Network [1] framework to Transductive Consensus Network (TCN), a semi-supervised multi-modal classification framework, and identify its two mechanisms: consensus and classification. By putting forward three variants as ablation studies, we show both mechanisms should be functioning together. Overall, TCNs outperform or align with the best benchmark algorithms when only 20 to 200 labeled data points are available.

1 Introduction

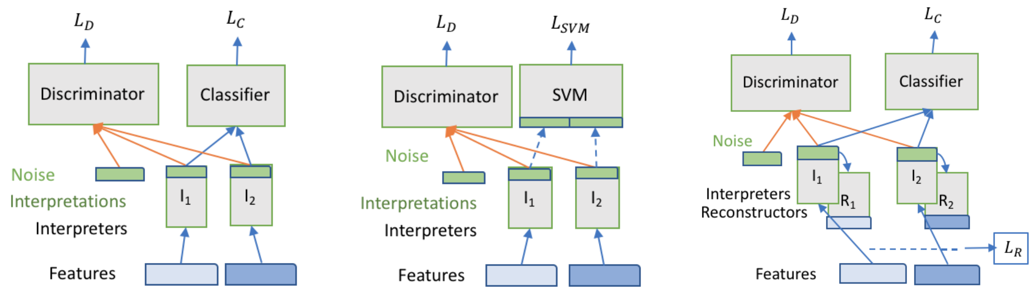
Traditionally, semi-supervised learning and multi-view learning are applied to increase data usage efficiency. On one hand, semi-supervised learning have shown good applications. TSVM [2] regularizes the decision boundaries using unlabeled data, Ladder [3] utilizes cascading autoencoder structures, and Categorical GAN [4] incorporates information theoretic optimization goals. On the other hand, multi-view learning distills information contained in multiple modalities. Co-training [5] and tri-training [6] directly sets up classifiers to supervise each other. PVAE [7] and SemiMVAE [8] set up variational autoencoder losses between modalities. Specifically, Consensus Networks [1] use adversarial training [9] that learns modality-invariant representations, and outperformed traditional algorithms on detecting cognitive impairments.

However, Consensus Networks are supervised learning algorithms, hence are limited by the availability of labeled data. This motivates us to push it forward to semi-supervised regime, resulting in Transductive Consensus Networks (TCN). TCNs function in two mechanisms, which we call the consensus mechanism and the classification mechanism. We put forward several variants in ablation study manner to study the roles of these two mechanisms, and show that the existence of both mechanisms are crucial to good performance of TCNs. Overall, TCN accuracies are better than or align with those of benchmark algorithms (semi-supervised or supervised, multi-modal or uni-modal) on Bank Marketing and DementiaBank datasets, when 20-200 labeled data points are available.

2 Models

2.1 Consensus Networks

We first briefly review the CN framework [1] for supervised, multi-view classification. Consider a dataset, $\{\mathbf{x^{(i)}}, y^{(i)}\}$ ($\mathbf{x^{(i)}} \in \mathcal{X}$), where each data point $\mathbf{x}$ is composed of feature values from multiple modalities (i.e., 'views'). If M is the total number of modalities, then $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M}]$. For $m = 1, ..., M$, $\mathbf{x_m}$ could have different dimensions, but the dimension of $\mathbf{x_m^{(i)}}$ is consistent throughout the dataset. E.g., there may be 200 acoustic features and 100 semantic features for a data point, but *all* data points are constrained to those dimensions.



(a) TCN-vanilla and TCN-embed (b) TCN-svm (c) TCN-AE

Figure 1: Network structures for TCN and TCN-embed (left), TCN-svm (middle) and TCN-AE (right), taking the example when there are two modalities in data (M=2).

There are M interpreter networks I_m $(m = 1, .., M)$, each compressing one modality of features into a representation, which we call **consensus interpretation vector**. A discriminator D tries to distinguish the origin of each latent representation. A classifier C makes predictions based on all representations.

$$\mathbf{v_m} = I_m(\mathbf{x_m}) \qquad P(\hat{m}) = D(\mathbf{v_m}) \qquad P(\hat{y}) = C([\mathbf{v_1}, .., \mathbf{v_m}])$$

The training is done by iteratively optimizing two targets:

$$\min_{C,I} \mathcal{L}_C \text{ and } \min_{D} \max_{I} \mathcal{L}_{\mathcal{D}}, \text{ where} \tag{1}$$

$$\mathcal{L}_C = \mathbb{E}_{\mathbf{x}}[\text{-log}P(y|\mathbf{x})] \qquad \mathcal{L}_D = \mathbb{E}_{\mathbf{x}}\mathbb{E}_m[\text{-log}P(\hat{m} = m|\mathbf{v_m})]$$

Note that empirically, an additional noise modality $\mathbf{v_0} \sim \mathcal{N}(\mu_{\mathbf{1..M}}, \sigma^2_{1..M})$ is injected to enhance the ability of the discriminator.

2.2 Transductive Consensus Networks

In this paper we extend CN to TCN. Formally, the input data include those labeled, $\{\mathbf{x^{(i)}}, y^{(i)}\}$ $(\mathbf{x^{(i)}} \in \mathcal{X}_{\mathcal{L}})$, and unlabeled, $\{\mathbf{x^{(i)}}\}$ $(\mathbf{x^{(i)}} \in \mathcal{X}_{\mathcal{U}})$. In the semi-supervised learning setting, there can be a lot more unlabeled data points than labeled: $|\mathcal{X}_{\mathcal{U}}| \gg |\mathcal{X}_{\mathcal{L}}|$, where the whole dataset is $\mathcal{X} = \mathcal{X}_{\mathcal{L}} \cup \mathcal{X}_{\mathcal{U}}$.

Here each data point $\mathbf{x}$ contains feature values from multiple modalities (i.e., 'views'), and the interpreter networks $I_m(m = 1, .., M)$, discriminator D and classifier C are set up identical to CN as well. Different from CN, the classification loss is defined on only those labeled data, while the discriminator loss is defined across both labeled and unlabeled data:

$$\mathcal{L}_{\mathcal{C}} = \mathbb{E}_{\mathbf{x} \in \mathcal{X}_{\mathcal{L}}} \text{ and } \mathcal{L}_{\mathcal{D}} = \mathbb{E}_{\mathbf{x} \in \mathcal{X}}\mathbb{E}_m[\text{-log}P(\hat{m} = m|\mathbf{v_m})] \tag{2}$$

2.3 TCN variants

TCNs function in two mechanisms: The consensus mechanism compresses each data sample into "consensus interpretations", and the classifier mechanism tries to make these interpretations meaningful. To perform ablation studies on these mechanisms, we test the following three variants.

TCN-embed consists of the same networks as TCN but $N_p = 30$ iterations of $\min_{D} \max_{I} \mathcal{L}_{\mathcal{D}}$ are carried out before the iterative optimizations (1). TCN-embed enhances consensus mechanism, yet allowing both mechanisms to cooperate. TCN-embed results align with TCN.

TCN-svm removes the classifier network from TCN-embed. After the pretraining phase across the whole dataset, we extract the consensus interpretations of those labeled data samples to train an SVM. TCN-svm lets the consensus mechanism to function alone, resulting in almost trivial classifiers.

TCN-AE contains an additional reconstructor network per modality $R_{1..M}(.)$, each recovering the input modality from latent interpretations: $\hat{\mathbf{x}}_\mathbf{m} = R_m(\mathbf{v_m} + \epsilon)$ Defining reconstruction loss as $\mathcal{L}_\mathcal{R} = \mathbb{E}_{x \in \mathcal{X}} \mathbb{E}_m |\hat{\mathbf{x}}_\mathbf{m} - \mathbf{x_m}|^2$, the optimization target in TCN-AE can be expressed as:

$$\min_{C, I_{1..M}, R_{1..M}} \mathcal{L}_\mathcal{C}, \text{ and } \max_{I_{1..M}} \min_{D} \mathcal{L}_\mathcal{D}, \text{ and } \min_{I_{1..M}, R_{1..M}} \mathcal{L}_\mathcal{R} \tag{3}$$

As shown in Figure 3, TCN-AE has inferior performances than TCN. Reconstruction in an autoencoder style counteracts the consensus mechanisms, and should *not* be used with CN models.

3 Experiments and Results

We run experiments on two classification datasets, DementiaBank [10] and Bank Marketing [11].

Dataset	N. of samples	%pos / %neg	N. features per modality
Bank Marketing ('BM')	9640	48.13 / 51.87	10 / 22 / 12
DementiaBank ('DB')	473	50.76 / 49.26	185 / 117 / 110

Table 1: In BM, the three modalities correspond to basic information, statistical data, and employment. In DB, the three modalities correspond to acoustic, syntactic-semantic, and lexical.

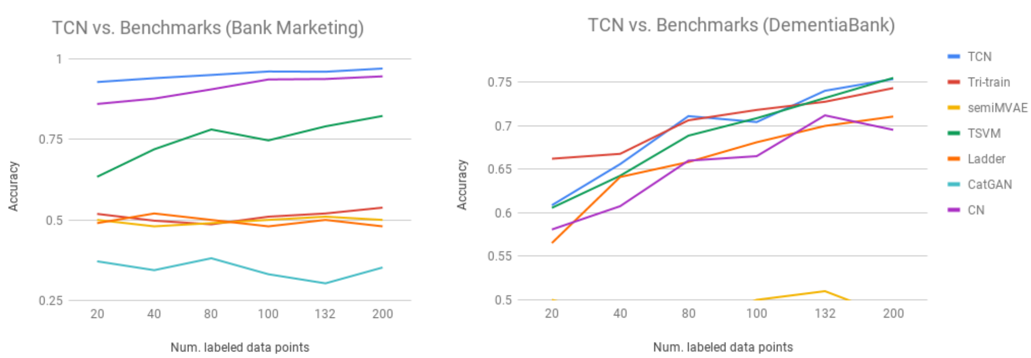


Figure 2: TCN (top blue line) outperforms or aligns with benchmark algorithms, including multi-modal semi-supervised (tri-train [6]), uni-modal semi-supervised (TSVM [2], Ladder [3], CatGAN [4]), and multi-modal supervised (CN [1]).

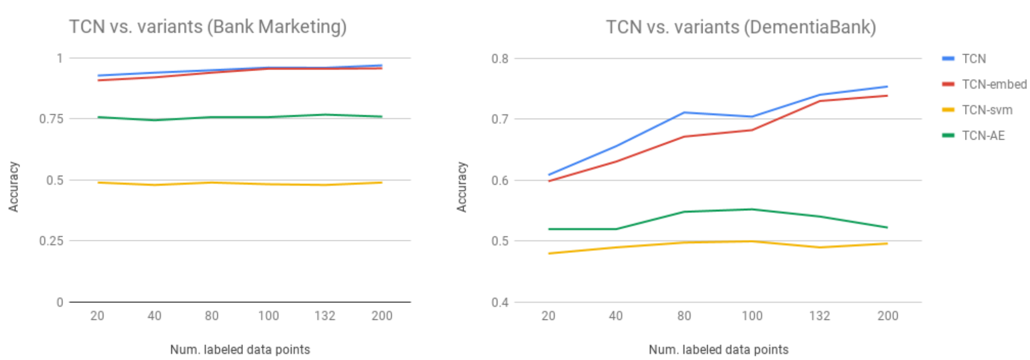


Figure 3: Accuracy plots for TCN vs its variants. Best viewed in colors. TCN-embed accuracies aligns with TCN, both significantly outperforming TCN-AE, which is better than TCN-svm. The consensus and classification mechanisms should both be present.

References

[1] Zining Zhu, Jekaterina Novikova, and Frank Rudzicz. Detecting cognitive impairments by agreeing on interpretations on linguistic features. *arxiv 1808.06570*, 2018.

[2] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference of Machine Learning (ICML-99)*, volume 99, pages 200–209, 1999.

[3] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Proceedings of Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[4] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

[5] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[6] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[7] Wei-Ning Hsu and James Glass. Disentangling by Partitioning: A Representation Learning Framework for Multimodal Sensory Data. 2018.

[8] Changde Du, Changying Du, Jinpeng Li, Wei-long Zheng, Bao-liang Lu, and Huiguang He. Semi-supervised Bayesian Deep Multi-modal Emotion Recognition. 2017.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] James T Becker, François Boiler, Oscar L Lopez, Judith Saxton, and Karen L McGonigle. The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis. *Archives of Neurology*, 51(6):585–594, 1994.

[11] Dua Dheeru and Efi Karra Taniskidou. {UCI} Machine Learning Repository, 2017.

[12] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 448–456, 2015.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[15] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceesings of International Conference on Learning Representations (ICLR)*, 2014.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

4 Appendix

4.1 Detailed description of datasets

The Bank Marketing dataset is from the UCI machine learning repository[11]. used for predicting whether the customer will subscribe a term deposit in a bank marketing campaign via telephone[12]. There are originally 4,640 positive samples (subscribe) and 36,548 negative ones (did not subscribe). Since consensus network models do not work well on imbalanced datasets, we randomly sample 5,000 negative samples to create an (almost) balanced dataset. We also convert the categorical raw features[1] into one-hot representations. We then divide the features into three somewhat arbitrary modalities: basic information, statistical data, and employment-related features.

Three modalities are determined as following. The division are somewhat arbitrary, except that we try to make the binary features resulting from one categorical feature be in the same modality.

1. Basic information: age, marital, education, housing, loan, contact, duration, pdays, previous, management
2. Statistical information: campaign, poutcome, emp.var.rate, unknown, cons.conf.idx, euribor3m, day_in_week (converted to 7 binary features), month (converted to 12 binary features)
3. Employment-related information: consumer price index, never employed, retired, self-employed, technician, services, student, housemaid, entrepreneur, blue-collar

DementiaBank[2] contains 473 narrative picture descriptions of the clinical "cookie-theft picture"[10], containing 240 positive samples (the Dementia class) and 233 negative samples (the Control class). We extract 413 linguistic features from each speech sample and their transcriptions, including acoustic (e.g., pause durations), semantic-syntactic (e.g., complexity of the syntactic parse structures), and lexical modality(e.g., average word length).

1. Acoustic-related features: phonation rate, mean pause duration, pause word ratio, total speech duration, short/medium/long pause count, speech rate, word/audio/(filled or unfilled) pauses durations, the mean/variance/kurtosis/skewness of the first 42 Mel Frequency Cepestral Coefficients
2. Syntactic-semantic features: probabilistic context-free grammar parsing tree heights (average / max / etc.), and the occurrences of 104 production rules (e.g: NP $\rightarrow$ DT).
3. Lexical and POS-derived features: the occurrences of part-of-speech tags, Brunet's index, Honore's statistics, word length, cosine distances between words in sentences, etc.

4.2 Implementation

For simplicity, we use fully connected networks for all of $I_{1..M}$, D, C, and $R_{1..M}$ in this paper. To enable faster convergence, all fully connected networks have a batch normalization[13] layer. For training neural networks, the batch size is set to 10. The neural network models are implemented using PyTorch[14], and supervised learning benchmark algorithms (SVM, MLP) in scikit-learn[15].

We use the Adam optimizer[16] with an initial learning rate of 0.001. In training TCN, TCN-embed, and TCN-AE, optimizations are stopped when the classification loss does not change by more than 10^{-5} in comparison to the previous step, or when the step count reaches 100. In the pre-training phase of TCN-embed and TCN-svm, training is stopped when the discrimination loss changes by less than 10^{-5}, or when pretraining step count reaches 20.

Sometimes the iterative optimization (i.e., the I-D-CI cycle for TCN / TCN-embed, and the I-D-RI-CI cycle for the TCN-AE variant) is trapped in local saddle points – the training classification loss does not change while the training classification loss is higher than log2≈ 0.693[3]. We check once more when training stops. If the training classification loss is higher than log2, the model is re-initialized

[1] https://archive.ics.uci.edu/ml/datasets/bank+marketing

[2] `https://dementia.talkbank.org`

[3] expected loss of a binary classifier with zero knowledge

with a new random seed and the training is restarted. Empirically this re-initialization only happen no more than once per ten runs, but the underlying cause need to be examined further.

4.3 Monitoring the similarity of interpretations

The similarity of interpretations It is important to evaluate whether the adversarial and classifier mechanisms make the interpretations more similar. To evaluate the similarity, we treat the hidden dimensions of each interpretation vector $\mathbf{v_m} = [v_{m,1}, v_{m,2}, .., v_{m,j}, ...]$ (after normalization by their sum) as discrete values of a probability mass function[4], which we write as p_m. The M modalities for each data point are therefore approximated by M probability distributions. Now, we can measure the relative JS divergences between each pair of interpretation vectors $\mathbf{v_m}$ and $\mathbf{v_n}$ derived from the same data sample ($\hat{D}(p_m||p_n)$). To acquire the relative value, we normalize the JS divergence by the total entropy in p_m and p_n:

$$\hat{D}(p_m||p_n) = \frac{1}{2(\mathbb{H}_{p_m} + \mathbb{H}_{p_n})}(D_{KL}(p_m||p_n) + D_{KL}(p_n||p_m))$$
$$\text{where } D_{KL}(p_m||p_n) = \sum_j v_{m,j} \log \frac{v_{n,j}}{v_{m,j}}$$

where $v_{m,j}$ and $v_{n,j}$ are the j^{th} component of $\mathbf{v_m}$ and $\mathbf{v_n}$ respectively. In total, for each data sample, $\frac{M(M-1)}{2}$ pairs of relative divergences are calculated. We average the negative of these divergences to get the similarity for the interpretations:

$$\text{Similarity} = \mathbb{E}_i \mathbb{E}_{m,n \in \{1..M\} \text{ and } m \neq n} \big\{ -\hat{D}(p_m^{(i)}||p_n^{(i)}) \big\}$$

Note that the "similarity" is defined such that its maximum possible value is 0 (where there is no JS divergence between any pair of the interpretation vectors), and it has no theoretical lower bound.

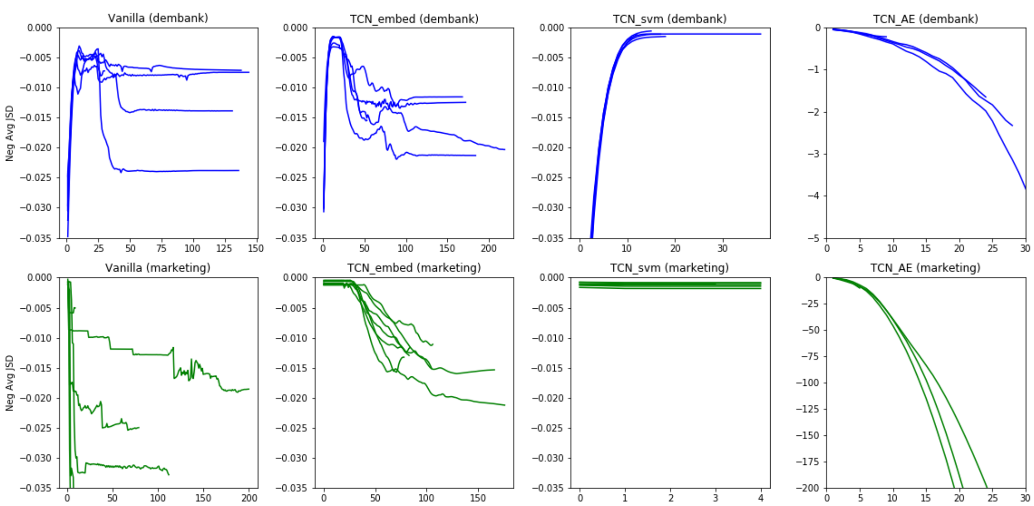


Figure 4: Examples of similarity plots against the number of steps taken, for DementiaBank using 80 labeled samples ("DB80", blue) and Bank Marketing using 20 labeled samples ("BM20", green). The y axis are scaled to (-0.035, 0) except TCN-AE, where the relative JS divergences "explode". Note that training stops when losses converge (as detailed in §4.2), so the trials may stop at different steps.

Experiments monitoring similarities We monitor the similarities (defined as negative relative JS divergence) between interpretation vectors. Several trends can be observed in Figure 4:

1. In vanilla TCN on DementiaBank, the similarity usually first rises, then drops to a stable final value. On Bank Marketing, the similarity drops without first rising. This might be

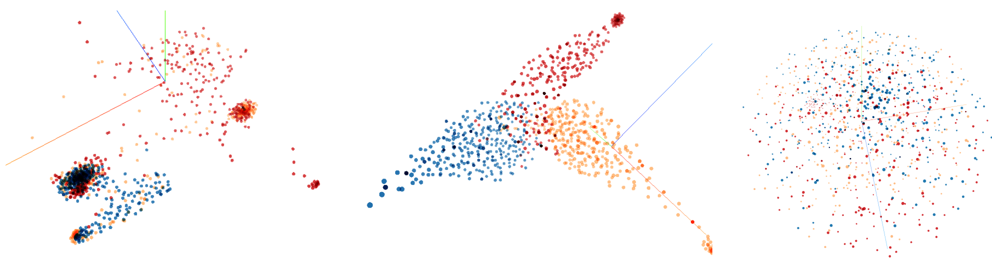
[4]There is a ReLU layer at output of each interpreter, so the probability mass will be non-negative.

attributed to Bank Marketing modalities (containing only ≈ 15 features per modality) being not as "sufficient and redundant" (borrowing from [6]) as DementiaBank (containing ≈110 per modality).

2. In the absence of the classifier mechanism, similarities converge to almost the highest possible value under the consensus mechanism. This can be seen in the pretraining phase of TCN-embed and TCN-svm. Note that there is no explicit steps of 'converging to the high similarity' on the Bank Marketing dataset – they directly go to the max values – because each Bank Marketing step contains more pretraining iterations than DementiaBank (≈10,000 samples vs. ≈500 samples), resulting in much stronger consensus mechanisms.
3. In TCN-embed, the classifier mechanism later "pulls down" the similarity. Note that the accuracy of TCN-svm is around 50%; we can infer that a meaningful consensus state need not have perfect similarity.
4. The addition of reconstructors inhibit the consensus mechanism in terms of reaching a high similarity between interpretations, as shown by the exploding JS divergences in TCN-AE models. This further illustrates that TCN distills information in a different aspect from denoising autoencoders.

4.4 Visualizing the interpretations

Figure 5 shows several 2D visualizations of interpretation vectors drawn from an arbitrary run, as an example of interpretations with low, medium, and high similarity. In §**??**, we illustrate how the similarities between interpretations evolve during optimization in TCN models.



(a) Step 2: sim=-0.0206 (b) Step 30: sim=-0.0072 (c) Step 110: sim=-0.0146

Figure 5: Three 2-D T-SNE[17] visualizations comparing interpretation vectors among modalities, taken from a run of the vanilla TCN (on DementiaBank dataset with 80 labeled data). The three colors represent three modalities. At step 2, the interpretations are distributed randomly. At step 110, they become mixed evenly. The most interesting embedding happens at step 30, when interpretations of the three modalities form three 'drumstick' shapes. With the highest symmetricity visually, this configuration of interpretations also has the highest *similarity* among the three.