Learning Case Features with Proxy-Guided Deep Neural Networks

Vibhas Vats, Zachary Wilkerson*, Hiroki Sato*,
David Leake[0000−0002−8666−3416], and David Crandall

Luddy School of Informatics, Computing, and Engineering, Indiana University
Bloomington IN 47408, USA

Abstract. The cost and difficulty of acquiring case features motivates interest in machine learning for feature acquisition. For computer vision domains, manual feature extraction has proven infeasible, but previous studies have shown the effectiveness of extracting features from deep neural models for case-based classification. Such approaches have generally been based on training the network for stand-alone classification accuracy, under the assumption that effective classification reflects high quality network features. However, it is not clear that the features best suited to network processing will be best for CBR. In response, this paper proposes refining previous network feature extraction approaches by adapting network training to reflect the goal of using network features for CBR. Specifically, it proposes augmenting conventional cross-entropy loss with a proxy term that reflects how the CBR system will use extracted features for similarity assessment. To this end, we investigate using Pairwise Distance, Cosine Similarity, and Sinkhorn Divergence as proxy functions within a triplet loss training framework. Evaluations on the benchmark image classification datasets MNIST, Animals with Attributes 2, and CIFAR-10 support the effectiveness of this method, with an integrated case-based classification system using the extracted features outperforming the feature extraction network applied end-to-end as well as integrated models developed in our previous research.

Keywords: Deep Learning, Case-Based Reasoning, Feature Learning

1 Introduction

The performance of case-based reasoning (CBR) systems depends on the quality of case retrieval. Traditionally, retrieval is based on hand-crafted feature vocabularies generated by experts [13, 25, 31]. However, manual feature acquisition is costly, hard to scale, and risks insufficient domain coverage in poorly-understood domains. For some tasks, such as image classification—the focus of this paper—symbolic feature generation is infeasible. Deep learning (DL) neural network methods have shown strong performance for a range of machine vision problems

* Equal contribution

(e.g., [8, 19, 37, 41]). enabled by the networks' ability to learn large sets of useful features from raw data via a scalable objective function optimization process. The ability of deep neural networks to generate useful domain features has kindled interest from the CBR community in integrating CBR with DL feature learning (e.g., [26–28, 34, 35, 38]). In these approaches, DL provides robust feature extraction from raw data to ease knowledge acquisition overhead for the case-based classifier, which can render decisions that are more explainable than those made by a DL-only system.

In previous work on extracting features from DL networks, including our own (e.g., [39]), the networks from which to extract features have been trained for stand-alone classification accuracy. However, we hypothesize that a neural model trained in this way may be encouraged to learn features that discriminate between examples from different classes, more so than ascribing similarity to examples from the same class (which is critically important for case-based classification). In response, this paper proposes augmenting the objective function for network training to encourage the DL model to learn features that are especially suitable for case retrieval. A naïve approach to this would be to integrate a measure of case-based classification accuracy into the objective function to influence feature learning directly. However, this is infeasible because the output of a case-based classifier using nearest-neighbor retrieval is non-differentiable; consequently, it is unsuitable for gradient-descent-based learning. Instead, we propose augmenting the DL framework with a differentiable proxy function that approximates the requirements of a case-based classifier during the learning process. We explore this approach using three different proxy functions—Pairwise Distance, Cosine Similarity, and Sinkhorn Divergence [17], which quantifies the divergence between feature distributions. These proxy functions quantify loss as a measure of "closeness" for two examples in the feature space, enabling the model to consider loss terms during training that are based both on feature-level differences between examples and on the model's overall classification accuracy, which is captured by cross-entropy.

We evaluate the benefit of including these proxy functions for feature extraction for case retrieval informing case-based classification on the MNIST [12], Animals with Attributes 2 (AwA2) [40], and CIFAR-10 [23] datasets. Results show that while optimal parameterizations may vary for different datasets, the proxy-based approach enables CBR classification accuracy that outperforms the analogous DL-only approach. Furthermore, this approach also outperforms our previous non-proxy-guided DL-CBR integrated systems (e.g., [39]).

2 Related Work

Effective feature vocabularies for retrieval are a critical requirement for CBR approaches. Traditionally, this requirement has been addressed through knowledge engineering (e.g., [13, 25, 31]). However, acquiring feature information in this way can be costly, and insufficient expert knowledge or capability to express this knowledge for some domains may make knowledge-engineered feature

sets incomplete or inadequate for certain tasks. Some symbolic learning methods have been successfully applied to this problem (e.g., [4, 6, 7, 10, 16]). More recently, integrating DL models with case-based methods has shown promise for leveraging the powerful inference capability of deep neural models to augment CBR systems. One such integration is an "inherently interpretable model" that uses a DL model to perform predictions based on similarity scores comparing generated features and known prototypes, reminiscent of retrieval in CBR (e.g., [5, 9]). In addition, DL models that focus on similarity (e.g. [22, 33]) have been used as metric learners for CBR similarity assessment [1, 30]. Other research investigates using CBR for post-hoc DL model explanation [3] or in integrated "twin systems" that extract weights from the DL model to retrieve an effective explanation for its decision [21].

Multiple projects have investigated extracting features for case-based classification from DL models. Turner et al. apply this approach to perform relative classification for novel or low-confidence evaluation examples, leveraging extracted features to inform a CBR model that uses clusters in its search space to form implicit classes [34, 35]. Feature extraction for CBR classification has also been shown to outperform other approaches in some case studies [30, 39]. Additional research has studied where best to extract features within a network, suggesting extraction from after the densely-connected layers in a convolutional neural network, at which point features have ideally been combined into more comprehensive indices [26]. Model-level structural parameters, such as DL model architecture and number of features extracted, have also been shown to affect feature quality [26, 27]. Extracted features have been combined with knowledge-engineered features for greater model accuracy [27, 38], and pretrained models have been used as the basis for extracting high-quality features for tasks in small-data domains, supporting integrated DL-CBR models that can outperform analogous DL-only models [39].

Notably, these works train the DL model independently, and then extract features for use in case retrieval, reflecting the plausible assumption that both CBR and DL models will find similar features useful. However, we hypothesize that higher quality features may be extracted from a DL model that is sensitized to the needs of the case-based classifier. Work in other contexts has shown the benefit of modifying loss functions to incorporate domain knowledge (e.g., [14]). In this work, we investigate using a proxy function within the DL objective function, ideally biasing the DL model training to select for features that highlight similarity relationships between cases.

3 Network Training Reflecting Case-Based Classification

Our method for sensitizing the feature extraction network to the needs of a case-based classifier focuses on using three different distance functions – Pairwise Distance, Cosine Similarity, and Sinkhorn Divergence [17]—to favor features that facilitate effective similarity assessment between pairs of examples. These distance functions are incorporated in a triplet loss optimization term [36] that

is combined with cross-entropy loss to guide weight updates during training. We discuss these model components in detail below.

DL classifiers typically learn features that enable discriminating between instances of different classes, while CBR models generally focus on ascribing similarity to examples that belong to the same class. Thus, we hypothesize that it may not be appropriate to use the same kind of features for both approaches. To address this, we target the DL model's loss function; typically, a DL model uses cross-entropy loss, which maximizes the number of correct classifications while minimizing the incidence of incorrect classifications for a given task. This process emphasizes features that are highly predictive of the target labels, ideally ensuring that each class is represented by a set of features in the learned feature space that differentiate it from each other class.

By contrast, a case-based classifier classifies new cases by comparing their feature representations to those of cases stored in a case base, relying on similarity scores to identify the closest matches. Consequently, features associated with intra-class similarity may be more appropriate. We hypothesize that these characteristics suggest that training networks in alignment with case-based classification will result in better features for CBR than training the network for standalone classification. Therefore, instead of using cross-entropy loss alone, we integrate another loss term that serves as proxy for the needs of the case-based classifier.

3.1 Three Proxy Functions for DL Training Guidance

We consider three functions to use as the proxy term in the DL loss function, targeting three candidate distance measures that might be important to a CBR system: Pairwise Distance considers simple Euclidean distance, Cosine Similarity the angular similarity between features in the feature space, and Sinkhorn Divergence the distance between feature distributions. We note that many distance measures are used within CBR, and that others might be most suitable for particular tasks; these are simply illustrative examples.

Pairwise Distance measures the similarity between two data points in a feature space by computing a distance measure. It is often used in contrastive learning frameworks, where the goal is to ensure that similar samples have smaller distances while dissimilar samples are pushed farther apart [36]. Eq. 1 shows a generalized formula of Pairwise Distance for two feature vectors f_i and f_j of input samples x_i and x_j. It estimates the *p-norm* of the difference between two vectors by adding some constant ϵ to avoid division by zero if p is negative:

$$D(f_i, f_j) = ||f_i - f_j + \epsilon||_p \tag{1}$$

Cosine Similarity focuses on the angular relationship between feature vectors, making it more robust to variations in magnitude. Specifically, it measures the cosine of the angle between two non-zero feature vectors defined in an inner product space and then normalizes this value relative to the vector magnitudes.

The Cosine Similarity value always lies within the interval $[-1, 1]$, with a similarity of 1 indicating that two vectors are perfectly aligned, -1 signifying that they are diametrically opposed, and 0 meaning that they are orthogonal (i.e., uncorrelated, Eq. 2).

$$S(f_i, f_j) = \frac{f_i . f_j}{max(||f_i||_2 . ||f_j||_2, \epsilon)}, \tag{2}$$

Here, f_i and f_j are the two feature vectors corresponding to examples x_i and x_j. ϵ is a small non-zero constant used to avoid division by zero.

Sinkhorn Divergence is derived from *optimal transport theory* [11], capturing a nuanced notion of similarity by considering the cost of transforming one probability distribution into another. In context of this work, Sinkhorn Divergence assesses how difficult it would be to execute a given *transport plan* (a transformation) that maps the features belonging to one example to match another, given a cost function for the transformation. This gives a refined notion of how similar two sets of data are, because it takes into account the effort of 'matching up' the points, not just their overall difference. That is, it considers the topography of the feature space between the features of the two examples in addition to the numeric distance, ideally sensitizing it to "challenging" regions that might signify decision boundaries. In this way, it has the potential to focus learning on dimensions that correlate strongly with a change in class. In image processing, Sinkhorn divergence can be used as a distance measure comparing images represented as distributions of local features, such as features generated by a deep network.

For features f_i and f_j, modeled as probability distributions over possible feature values, the regularized Wasserstein distance [2] is:

$$W_\epsilon(f_i, f_j) = \min_{\gamma \in U(f_i, f_j)} \sum_{a,b} \gamma(a, b)\, c(a, b) \; - \; \epsilon\, H(\gamma), \tag{3}$$

where $U(f_i, f_j)$ is the set of valid transport plans, $c(a, b)$ is the cost function (e.g. squared Euclidean distance), $\gamma(a, b)$ is the transport plan, and $H(\gamma)$ is an entropy term (regularized by ϵ). Sinkhorn Divergence then refines this by removing self-similarity biases [17]:

$$\mathcal{S}_\epsilon(f_i, f_j) = W_\epsilon(f_i, f_j) \; - \; \tfrac{1}{2}\Big(W_\epsilon(f_i, f_i) + W_\epsilon(f_j, f_j)\Big). \tag{4}$$

This encourages the model to learn features that reflect meaningful alignments in the data rather than simple distances, providing richer estimates for properties relevant to case-based classification [11].

3.2 Triplet Optimization: Enhancing Similarity Learning in DL

Triplet optimization is a technique in metric learning approaches [20] that encourages a deep neural network to learn features that capture meaningful relationships between data points [32]. Specifically, it encourages a model to learn

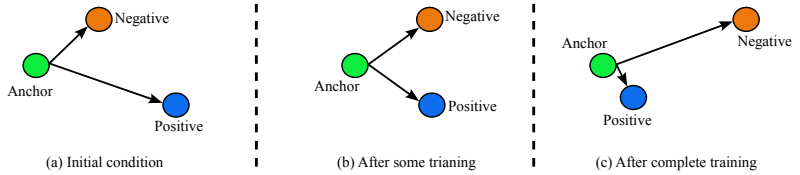

Fig. 1: Triplet loss minimizes the distance between *anchor* and a *positive* examples and maximizes the distance between the *anchor* and *negative* examples [32].

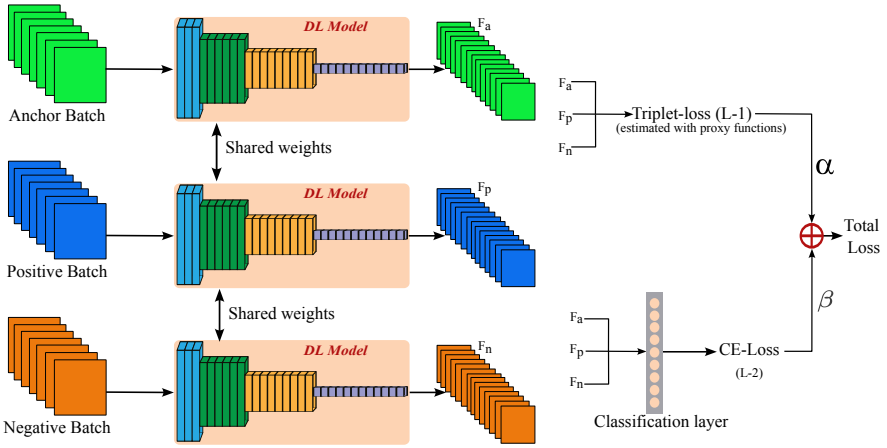

Fig. 2: Triplet-style training learns similarity-based features using three example types: anchor, positive, and negative. Features are extracted via the DL model, optimizing similarity between anchor and positive samples while minimizing it for negative samples.

how to distinguish between similar examples that belong together and those that do not. Consider a target scenario where the goal is to find the "closest" match from a set of past cases. In this context, the goal is to teach the model that the correct (or similar) case should be closer to the target than any incorrect (or dissimilar) case. Concretely, this is achieved by organizing the training data into triplets, each consisting of a target example (called the *anchor*, denoted by a), a similar example (the *positive*, denoted by p), and a dissimilar example (the *negative*, denoted by n). During training, the model is encouraged to pull the anchor and positive examples close together while pushing the negative example farther away, as shown in Fig. 1. By continuously adjusting the model's internal parameters according to this rule, the model learns a representation that better captures both underlying similarities and differences.

Fig. 2 shows how we apply this approach to the feature extractor network. Features corresponding to anchor (F_a), positive (F_p), and negative (F_n) are extracted just before the classification layer and sent to the triplet loss estimator function, which uses the proxy function for similarity-based distance estimation, with an aim to keep similar features closer to each other (F_a and F_p) and dissimilar features away from each other (F_a and F_n). A visualization of this op-

timization process is shown in Fig. 1, assuming at the start of the optimization process the anchor and the positive features were far apart as compared to the anchor and negative (Fig. 1(a)). The aim of this optimization is to bring anchor and positive feature close together and push away the negative features, as shown in Fig. 1(c). Following is the mathematical representation of the constraint used for this optimization

$$\begin{aligned} &||f(x_i^a) - f(x_i^p)||_2^2 + m < ||f(x_i^a) - f(x_i^n)||_2^2 \\ &\forall f(x_i^a), f(x_i^p), f(x_i^n) \in \tau \end{aligned} \tag{5}$$

Here, x_i^a, x_i^p, and x_i^n are an anchor, a positive, and a negative example, respectively; f is a functional form of feature extraction network, m is a margin imposed on the distance between positive and negative pairs, meaning these pairs need at least be m distant, and τ is the set of all possible triplet pairs defined in the training set. During training, the triplet loss is written as

$$\mathcal{L}_{\text{triplet}} = \max(0, d(f_a, f_p) - d(f_a, f_n) + m) \tag{6}$$

For a given triplet of examples (x_a, x_p, x_n), the network generates the corresponding feature vectors f_a, f_p, and f_n, which are then passed to the chosen distance function d (i.e., Pairwise Distance, Cosine Similarity, or Sinkhorn Divergence), which consists of three parts: 1) $d(f_a, f_p)$ is minimized, ensuring similar instances cluster together in feature space, 2) $d(f_a, f_n)$ is maximized, ensuring that dissimilar classes remain well-separated, and 3) the margin m prevents trivial solutions where the network pushes all cases arbitrarily far apart by only activating the loss function when the negative sample is closer to the anchor than the positive by less than m.

We integrate triplet loss alongside cross-entropy loss ($\mathcal{L}_{CE}$) with two independently controlled weight coefficients α and β as shown in Fig. 2. As a result, the overall loss formulation for each proxy function can be estimated as:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{triplet} + \beta \mathcal{L}_{CE} \tag{7}$$

4 Evaluation

4.1 Research Hypotheses

We evaluate the following hypotheses in our experiments:

1. **The triplet-style, proxy-guided approach will achieve greater classification accuracy than the analogous DL-only approach.** This may not be the case for all proxy functions, but there will exist a parameterization for every evaluated dataset such that our DL-CBR model outperforms the DL-only model.
2. **More sophisticated proxy functions will yield superior quality features, as shown by higher case-based classification accuracy.** Thus, Sinkhorn Divergence will lead to the highest classification accuracy, and Pairwise Distance (being the simplest method) will lead to the lowest overall accuracy.

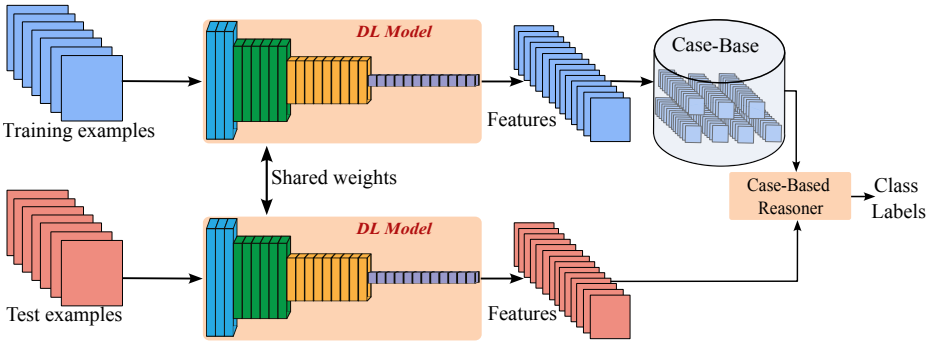

Fig. 3: The case base is constructed from training set images, and the case-based classifier assigns classification labels by comparing query features with stored cases based on Manhattan distance.

4.2 The DL-CBR Testbed System

These experiments use AlexNet [24] and pretrained DenseNet121 [18] architectures for the feature extractor model, with the Geomloss library [15] used to implement a Sinkhorn distance loss calculation compatible with the PyTorch environment [29]. During training, we keep the value of β fixed to 10 and vary the value of α from 5 to 20 in increments of 5 in our experiments. The values used for α and β were found to be empirically useful based on preliminary experiments but could feasibly be normalized such that $\alpha = 1 - \beta$; we do not present $\alpha = 0$ results, as these would repeat DL-only results that have been presented in other papers (e.g., [39]), with no triplet loss component. The case-based classifier is a retrieval-only system (no adaptation) that uses an unweighted similarity metric based on normalized Manhattan distance for similarity assessment; such an approach facilitates straightforward explanations based on case presentation. The case base is initialized using a random selection of examples that are passed through the DL model to generate representative feature vectors that are extracted from the penultimate network layer (Fig. 3). Once the case base is established, test images from the same dataset are processed through the DL model, extracting their features in an identical manner. These extracted features serve as query cases for the case-based classifier. The case-based classifier then determines the class label of each test query by comparing its feature representation to stored cases in the case base, using the similarity metric to identify the closest match.

4.3 Evaluation Datasets and Relevant Parameters

During training for all datasets, batches of triplets are created that associate anchor examples at random with a positive example from the same class and a negative example from an unlike class. Pairwise Distance and Cosine Similarity values are calculated as described above, and Sinkhorn Divergence values are also calculated as described above, following a softmax operation performed on

the intermediate activations. We train and evaluate the model on each of the following datasets:

MNIST: The MNIST (Modified National Institute of Standards and Technology) dataset [12] contains images of 70,000 handwritten digits from 0 to 9, paired with the name of the numeral. We train an AlexNet [24] model from scratch on the training set of 60,000 images and test on 10,000 test set images. For the results shown below, we use a batch size of 128 and a learning rate of $3 * 10^{-4}$ with Adam optimizer (batch size of 256 with learning rate of $3 * 10^{-5}$ for Sinkhorn Divergence, as using the other parameters resulted in errors using the Geomloss library). We train the model for 30 epochs and do not use early stopping. Mean and standard deviation values are calculated by segmenting the 10,000 test examples into separate trials and then conducting each end-to-end experiment (including training) three times, resulting in thirty overall test trials for reliable mean and standard deviation calculation.

AwA2: . The Animals with Attributes 2 (AwA2) dataset contains 37,322 images distributed across 50 classes [40]. Experiments using this dataset enable comparison with recent work on feature extraction for CBR [39]. In these trials, a training set of 4096 examples is randomly selected from all possible training examples in the dataset; the pretrained DenseNet121 [18] model (as in [39]) is trained with the same batch size and learning rate as MNIST. During evaluation, 10,000 test images are selected at random and divided into sets as in the MNIST evaluation, again producing thirty overall trials.

CIFAR-10: The CIFAR-10 dataset [23] consists of 60,000 color tiny images with 6000 images per class for 10 classes. In the experiments, we use the entire training set to train an AlexNet model from scratch. We use a batch size of 128 and a learning rate of 3 * $10^{-}3$ with Adam optimizer. We trained the model for 100 epochs without early stopping. For the results shown below, test image sets are again selected and segmented as for MNIST evaluation.

5 Results and Discussion

Experimental results are displayed in Figs. 4, 5, and 6. In numerous instances the case-based classifier using extracted features outperforms the analogous DL-only model, supporting hypothesis 1; surprisingly, the proxy-guided model also achieves higher classification accuracy on AwA2 ($88.7 \pm 1.4\%$) than our best-performing DL-CBR model to date ($87.4 \pm 0.5\%$) and the analogous DL-only model ($85.9 \pm 2.4\%$), underscored by a student T-test p-score of $p \leq 0.0013$ which indicates a high probability of significance. Additional fine-tuning might further increase this improvement. However, as long as accuracy is sufficient, another motivation for the case-based approach is the ability to explain decisions in terms of cases.

Concerning the proxy loss functions, results show that there is not a one-size-fits-all similarity metric that performs best for all datasets. We hypothesized that

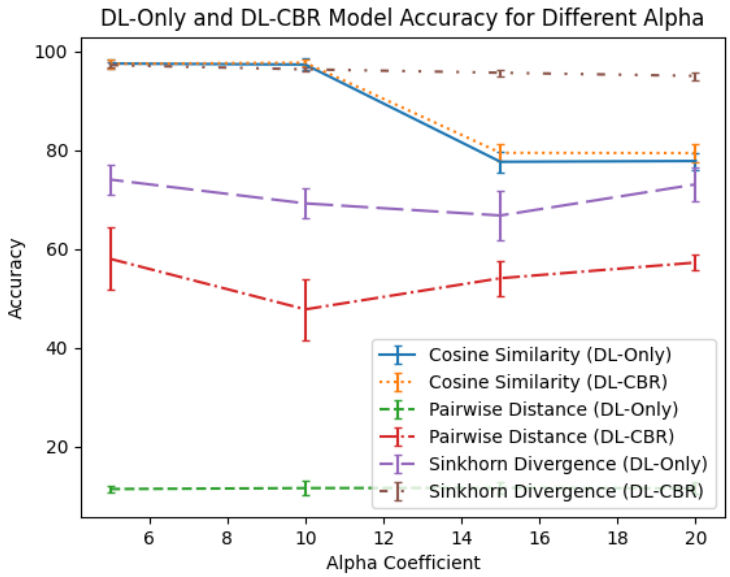

Fig. 4: Classification accuracy on the MNIST dataset, for different α coefficient values. Error bars represent one standard deviation.

Sinkhorn Divergence might best capture important similarity characteristics, but this is not necessarily the case. Sinkhorn Divergence does appear to lead to very consistent model performance, regardless of the relative contributions of triplet and cross-entropy loss terms, but Cosine Similarity appears to produce at least as strong performance. Across the board, Pairwise Distance leads to the least accurate model performance, as expected.

Finally, while the experiments show reasonable standard deviation values overall, for isolated trials the model fails to converge or otherwise exhibits extreme, out-of-distribution behavior. We consider these as outliers, but they underscore the need to train and evaluate this model carefully, potentially using multiple iterations, to ensure useful feature extraction post-training.

Evaluation on MNIST: In our tests for the MNIST dataset, the proxy-guided system consistently achieves comparable or greater classification accuracy compared with the DL-only model (Fig. 4). Often, this improvement is statistically significant. This supports that the combination of triplet-style training and the proxy loss function benefits case-based classification compared to the end-to-end DL approach, especially given that the DL-only accuracy often decreases as the contribution of the similarity-based loss function increases. One notable exception is when Pairwise Distance is used, but that model consistently fails to converge, exhibiting random-guess accuracy. Sinkhorn Divergence appears to be the best proxy function, with Cosine Similarity a close second.

Another interesting trend is the relative flatness of the DL-CBR trend lines in general; this suggests that the relative contribution of the similarity loss is

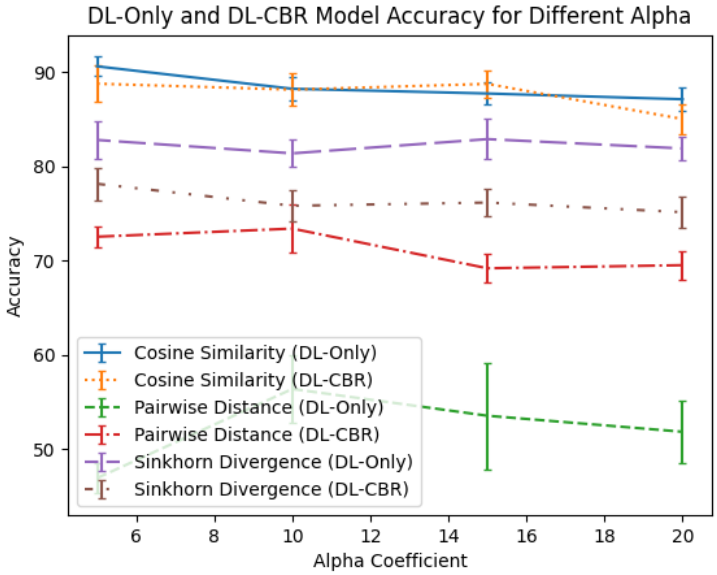

Fig. 5: Classification accuracy on the AwA2 dataset, for different α coefficient values. Error bars represent one standard deviation.

relatively unimportant. This could mean that the relative contribution of the proxy and cross-entropy loss components is less significant to the DL model during training than the fact that both are present. Alternatively (or perhaps additionally), the case-based classifier might be sensitive to subsets of features developed by both the proxy and cross-entropy loss components in such a way that their relative contribution does not impact classification accuracy.

Evaluation on AwA2: For AwA2, Cosine Similarity yields higher classification accuracy across the board relative to Sinkhorn Divergence; Pairwise Distance is once again the least performant metric. Interestingly, it is less clear-cut whether the DL-CBR approach outperforms the DL-only model when using Cosine Similarity (for Sinkhorn Divergence, it performs worse). However, given that the case-based classification accuracy is higher for the proxy-guided approach than for other DL-CBR approaches for the same experimental parameters [39], it supports the proxy-guided approach.

Trends for the AwA2 dataset are generally flat, similar to the MNIST results. Again, this suggests that the relative contributions of the proxy and cross-entropy loss terms are relatively inconsequential. Especially in light of the greater image resolution and class variety present in AwA2 relative to MNIST, this suggests future work to contextualize these findings and focus specifically on the α and β coefficients (e.g., setting one of them to zero during training).

Evaluation on CIFAR-10: Performance on CIFAR-10 arguably does not support our first hypothesis, as all DL-CBR models perform no more accurately

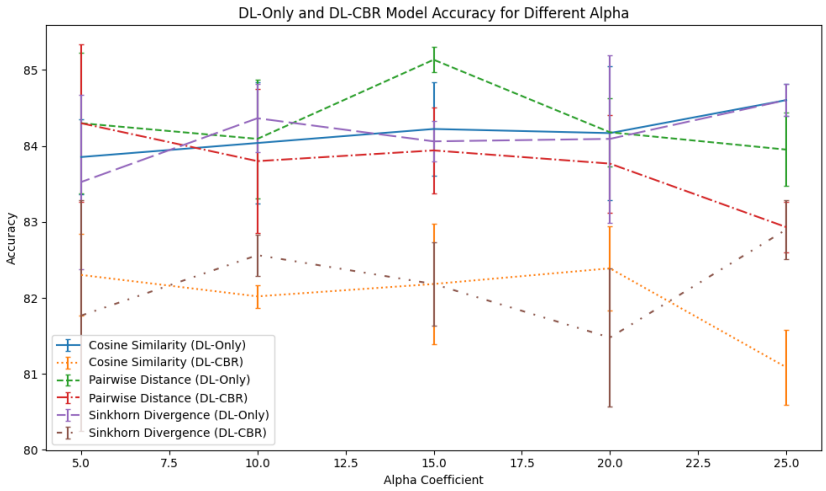

Fig. 6: Classification accuracy on the CIFAR-10 dataset, for different α coefficient values. Error bars represent one standard deviation.

than the corresponding DL-only model. Our second hypothesis is challenged as well, with Pairwise Distance being the most useful proxy function. That said, general trends bear broad similarities to the AwA2 results, and the range of all accuracy values is less than 5%. Considering this in the context of the additional explainability offered by using a DL-CBR system rather than an end-to-end DL model, the proxy-guided approach may still be useful.

To explain these results, we hypothesize that CIFAR-10's high intra-class variability and small input image size makes it difficult to define feature similarity. Images in the same category (e.g., "bird") can differ drastically in species, color, pose, and scale, forcing classification to rely more on context and texture than subject features. Additionally, the dataset's limited size struggles to capture the full diversity required for robust feature learning. While deep learning can leverage sheer data volume, the case-based models depend on clearly defined features – a challenge given CIFAR-10's variability and constrained dataset size.

6 Conclusions and Future Work

This paper presents a novel method for training DL models for feature extraction, using similarity-focused proxy functions in the DL training loss to encourage generating features that are useful for case retrieval. In our experimental results, this approach is successful relative to the analogous DL-only model for multiple domains, and is capable of outperforming our previous DL-CBR approaches (e.g., [39]). Considering the three evaluated similarity functions, Pairwise Distance is generally the least effective (though it had some success with CIFAR-10), Sinkhorn Divergence is the most consistent, and Cosine Similarity is appealing in its combination of simplicity and effectiveness. Avenues for future work include considering potential parameterizations for increasing classification accu-

racy, including different neural architectures, other loss functions or methods for calculating the aggregate loss value, and differentiable CBR methods that might allow for direct inclusion of CBR performance as a loss term itself. Additional work involves evaluating this model for additional datasets and distance metrics and developing practical approaches for selecting metrics for different domains.

References

1. Amin, K., Lancaster, G., Kapetanakis, S., Althoff, K.D., Dengel, A., Petridis, M.: Advanced similarity measures using word embeddings and siamese networks in CBR. In: Intelligent Systems and Applications. pp. 449–462. Springer, Cham (2020)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan (2017)
3. Bach, K., Mork, P.: On the explanation of similarity for developing and deploying CBR systems. In: Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2020) (05 2020)
4. Barletta, R., Mark, W.: Explanation-based indexing of cases. In: Kolodner, J. (ed.) Proceedings of a Workshop on Case-Based Reasoning. pp. 50–60. DARPA, Morgan Kaufmann, Palo Alto (1988)
5. Barnett, A.J., Schwartz, F.R., Tao, C., Chen, C., Yinhao, R., Lo, J.Y., Rudin, C.: Interpretable mammographic image classification using case-based reasoning and deep learning. In: Proceedings of IJCAI-21 Workshop on Deep Learning, Case-Based Reasoning, and AutoML (2021), https://arxiv.org/pdf/2107.05605
6. Bhatta, S., Goel, A.: Model-based learning of structural indices to design cases. In: Proceedings of the IJCAI-93 Workshop on Reuse of Design. pp. A1–A13. IJCAI, Chambery, France (1993)
7. Bonzano, A., Cunningham, P., Smyth, B.: Using introspective learning to improve retrieval in CBR: A case study in air traffic control. In: Case-Based Reasoning Research and Development: Proceedings of the Second International Conference on Case-Based Reasoning, ICCBR-97. pp. 291–302. Springer, Berlin (1997)
8. Chai, J., Zeng, H., Li, A., Ngai, E.W.: Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications **6**, 100134 (2021)
9. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: Deep learning for interpretable image recognition. In: Advances in Neural Information Processing Systems 32, pp. 8930–8941. Curran (2019)
10. Cox, M., Ram, A.: Introspective multistrategy learning: On the construction of learning strategies. Artificial Intelligence **112**(1-2), 1–55 (1999)
11. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transportation distances (2013)
12. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
13. Domeshek, E.: Indexing stories as social advice. In: Proceedings of the Ninth National Conference on Artificial Intelligence. pp. 16–21. AAAI Press, Menlo Park, CA (1991)
14. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1606–1615. ACL, Denver (2015)

15. Feydy, J., Séjourné, T., Vialard, F.X., Amari, S.i., Trouve, A., Peyré, G.: Interpolating between optimal transport and mmd using sinkhorn divergences. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 2681–2690 (2019)
16. Fox, S., Leake, D.: Introspective reasoning for index refinement in case-based reasoning. The Journal of Experimental and Theoretical Artificial Intelligence **13**(1), 63–88 (2001)
17. Genevay, A., Peyre, G., Cuturi, M.: Learning generative models with sinkhorn divergences. In: Storkey, A., Perez-Cruz, F. (eds.) Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 84, pp. 1608–1617. PMLR (09–11 Apr 2018), https://proceedings.mlr.press/v84/genevay18a.html
18. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks (2016), https://arxiv.org/abs/1608.06993
19. Jasim, W.N., Mohammed, R.J.: A survey on segmentation techniques for image processing. Iraqi Journal for Electrical & Electronic Engineering **17**(2) (2021)
20. Kaya, M., Bilge, H.Ş.: Deep metric learning: A survey. Symmetry **11**(9), 1066 (2019)
21. Kenny, E.M., Keane, M.T.: Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (2019)
22. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Proceedings of the 32nd International Conference on Machine Learning (2015)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. vol. 1, pp. 1097–1105 (2012)
25. Leake, D.: An indexing vocabulary for case-based explanation. In: Proceedings of the Ninth National Conference on Artificial Intelligence. pp. 10–15. AAAI Press, Menlo Park, CA (July 1991)
26. Leake, D., Wilkerson, Z., Crandall, D.: Extracting case indices from convolutional neural networks: A comparative study. In: Case-Based Reasoning Research and Development, ICCBR 2022 (2022)
27. Leake, D., Wilkerson, Z., Vats, V., Acharya, K., Crandall, D.: Examining the impact of network architecture on extracted feature quality for CBR. In: Case-Based Reasoning Research and Development, ICCBR 2023. Springer (2023)
28. Martin, K., Wiratunga, N., Sani, S., Massie, S., Clos, J.: A convolutonal siamese network for developing similarity knowledge in the Selfback dataset. In: Proceedings of the International Conference on Case-Based Reasoning Workshops, CEUR Workshop Proceedings, ICCBR. pp. 85–94 (2017)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)

30. Sani, S., Wiratunga, N., Massie, S.: Learning deep features for kNN-based human activity recognition. In: Proceedings of ICCBR 2017 Workshops (CAW, CBRDL, PO-CBR), Doctoral Consortium, and Competitions co-located with the 25th International Conference on Case-Based Reasoning (ICCBR 2017), Trondheim, Norway, June 26-28, 2017. CEUR Workshop Proceedings, vol. 2028, pp. 95–103. CEUR-WS.org (2017)
31. Schank, R., Brand, M., Burke, R., Domeshek, E., Edelson, D., Ferguson, W., Freed, M., Jona, M., Krulwich, B., Ohmayo, E., Osgood, R., Pryor, L.: Towards a general content theory of indices. In: Proceedings of the 1990 AAAI spring symposium on Case-Based Reasoning. AAAI Press, Menlo Park, CA (1990)
32. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (Jun 2015). https://doi.org/10.1109/cvpr.2015.7298682, http://dx.doi.org/10.1109/CVPR.2015.7298682
33. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
34. Turner, J.T., Floyd, M.W., Gupta, K.M., Aha, D.W.: Novel object discovery using case-based reasoning and convolutional neural networks. In: Case-Based Reasoning Research and Development, ICCBR 2018. pp. 399–414 (2018)
35. Turner, J.T., Floyd, M.W., Gupta, K.M., Oates, T.: NOD-CC: A hybrid CBR-CNN architecture for novel object discovery. In: Case-Based Reasoning Research and Development, ICCBR 2019. pp. 373–387 (2019)
36. Vassileios Balntas, Edgar Riba, D.P., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 119.1–119.11. BMVA Press (September 2016). https://doi.org/10.5244/C.30.119, https://dx.doi.org/10.5244/C.30.119
37. Vats, V.K., Joshi, S., Crandall, D.J., Reza, M.A., Jung, S.: Gc-mvsnet: Multi-view, multi-scale, geometrically-consistent multi-view stereo. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3230–3240. IEEE Computer Society (jan 2024)
38. Wilkerson, Z., Leake, D., Crandall, D.: On combining knowledge-engineered and network-extracted features for retrieval. In: Case-Based Reasoning Research and Development, ICCBR 2021. pp. 248–262 (2021)
39. Wilkerson, Z., Leake, D., Vats, V., Crandall, D.: Extracting indexing features for CBR from deep neural networks: A transfer learning approach. In: Case-Based Reasoning Research and Development, ICCBR 2024. Springer (2024)
40. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (2018)
41. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: A survey. Proceedings of the IEEE **111**(3), 257–276 (2023). https://doi.org/10.1109/JPROC.2023.3238524