Homework 4

Instructions: Complete the following questions. There are a total of 25 points for this homework. Each question is marked with its corresponding points. For all plotting questions, do not hand-draw the functions, use plotting tools instead.

Collaboration: If you collaborate with other students on the homework, list the names of all your collaborators.

Submission: Upload a **PDF** of your response through **Canvas** by **11/7 at 1pm**.

Notation: We will use this set of math notation specified on course website, whose LaTeX source is available on Canvas. For example, c is a scalar, $\mathbf{b}$ is a vector and $\mathbf{W}$ is a matrix. You are encouraged (although not enforced) to follow this notation.

This homework has two sections, roughly of equal work.

- **(13pt)** The first section focuses on similarity-based learning in both supervised and self-supervised learning, with a focus on the latter. We derive certain properties of such learned representations, and highlight a connection between the supervised objective and the self-supervised objective.
- **(12pt)** The second section focuses on empirically analyzing learned representations from reconstruction-based and similarity-based objectives. Via nearest neighbors, we show how choices of self-supervised objectives can affect the information captured in representations.

Similarity-Based Learning: Self-Supervised and Supervised (13pt)

Recall from lecture that contrastive learning optimizes an encoder network on certain similarity measures. In this section, we will explore such similarity-based objectives, and see how they relate to contrastive learning and supervised learning.

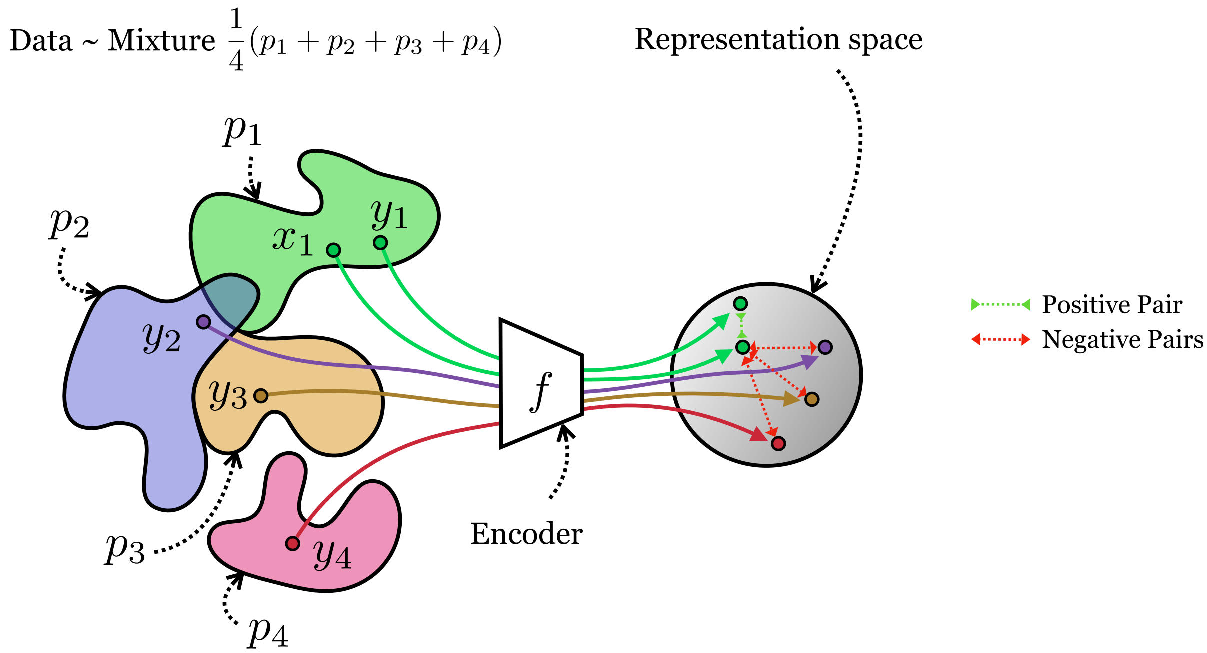
Setting: For some data space $\mathcal{X}$, we consider data coming from a (equal) mixture of N distributions $\{p_i\}_{i=1}^N$, where each p_i is a distribution over $\mathcal{X}$, and N is finite. These represent N groups of data samples, where samples within each group are considered *similar*. Essentially, $\{p_i\}_{i=1}^N$ defines the similarity relation that we will learn from.

Contrastive Learning: Consider an encoder $f\colon \mathcal{X} \rightarrow \mathbb{S}^{d-1}$, where $\mathbb{S}^{d-1}$ is the $(d-1)$-dimensional unit hypersphere $\subset \mathbb{R}^d$ (where the normalization removes a dimension). Usually, f is a deep encoder network computing a vector $\in \mathbb{R}^d$, followed by an l_2-normalization step (dividing the vector by its norm). $\mathbb{S}^{d-1}$ is our representation space. We consider the following specific form of contrastive loss:

$$\mathcal{L}_{\mathsf{contr}}(f) \triangleq \frac{1}{N}\sum_{i=1}^{N} \mathop{\mathbb{E}}_{\substack{x_i\sim p_i \\ \forall j\in\{1,\dots,N\},\ y_j\sim p_j}} \left[-\log \frac{e^{f(x_i)^\mathsf{T} f(y_i)}}{\sum_{j=1}^{N} e^{f(x_i)^\mathsf{T} f(y_j)}}\right], \tag{1}$$

where (x_i, y_i) is the *positive pair* and, for $j \neq i$, (x_i, y_j) are the *negative pairs*.

An example with $N = 4$ mixtures:



For this section, we consider optimizing over all possible functions. In practice, f is usually a parametrized encoder using deep neural networks.

Remarks on our setting (not required for completing the questions): Our setting and the contrastive loss form in Equation (1) may look a bit different from what you encounter elsewhere. For example, we assume that

- N, the number of similarity groups (p_i's), is finite,
- The logits are not scaled by some temperature parameter
- For an element $x_i \sim p_i$. the negatives are always formed from exactly one sample from each other similarity group p_j ($j \neq i$), and never from the same group p_i.

These assumptions make our analysis easier. While the exact results we derive may be slightly different from other analyses, the core messages are the same.

1. **(1pt; Augmentations and Image Contrastive Learning)** For contrastive learning on images, recall that we often choose positive pairs as augmentations of the same image, and negative pairs as augmentations of different images.

   Consider an image dataset $\{\mathsf{img}_i\}_{i=1}^K$, and $p_{\mathsf{aug}}(\,\cdot \mid \mathsf{img})$ the distribution over augmented versions of an image img.

   To cast this into our setting, write down formulas for N and $\{p_i\}_i$ in terms of $\{\mathsf{img}_i\}_{i=1}^K$,

K, and p_{aug}.

2. **(1pt; Instance Discrimination/Classification)** Equation (1) is essentially a cross-entropy loss. For x_i and $\{y_j\}_{j=1}^N$, write down the number of classes C, the logits vector $\in \mathbb{R}^C$, and the target class $\in \{1, \dots, C\}$. Are entries of the logits vector bounded within some range? If so, write down the range.

3. **(0.5pt; Dot Products and Distances)** For $u, v \in \mathbb{S}^{d-1}$, write down the formula of $\|u - v\|_2$ in terms of $u^\mathsf{T} v$.

4. **(7pt; From Classification to Representation Geometry)** In Question 2, we cast the contrastive loss as a classification loss, where each combination of x_i and $\{y_j\}_{j=1}^N$ gives a classification question for the encoder f. Unlike the usual supervised classification training, the target label here is constructed differently, and the logits follow a weird form involving f.

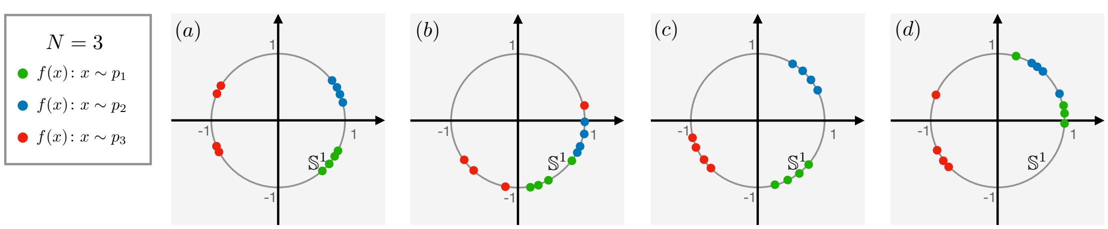
   Let's build some intuition on what this classification task even means. In particular, we characterize what encoders f can achieve perfect classification *accuracy*.

   Assumption:

   $$\text{Each } p_i \text{ is restricted to a finite set } S_i, \text{ where } p_i(s) > 0 \text{ for all } s \in S_i. \tag{2}$$

   (a) **(2pt)** Suppose $N = 3$, each p_i is a uniform distribution over 4 elements, and f encoder simply outputs a 2-dimensional unit vector (on $\mathbb{S}^1$).

   In the four plots below, we show four different feature distributions from different f choices. For each one, decide whether f achieves a perfect classification accuracy (over all choices of x_i and $\{y_j\}_{j=1}^N$).



   (Recall that a prediction is correct if the largest entry of logit is *uniquely* target.)

   (b) **(1pt; Feature Geometry)** Recall that we have Assumption (2). For example, the scenario in Question 4a is a case satisfying the assumption.

   Consider the following quantities:

   $$\text{diameter}_i(f) \triangleq \max_{s,t \in S_i} \|f(s) - f(t)\|_2 \qquad \forall i \tag{3}$$

   $$\text{margin}_{i \to j}(f) \triangleq \min_{\substack{s_i \in S_i \\ s_j \in S_j}} \|f(s_i) - f(s_j)\|_2 \qquad \forall i, j \tag{4}$$

These are metrics closely related to the positive and negative pair distributions. In particular, $\mathrm{diameter}_i$ is defined with the positive pairs from p_i (supported on set S_i), and $\mathrm{margin}_{i \to j}$ is defined with the negative pairs from p_i and p_j.

Describe what $\mathrm{diameter}_i(f)$ and $\mathrm{margin}_{i \to j}(f)$ geometrically represent in terms of feature vectors.

(c) **(1pt; Optimal Feature Geometry)** Which of the following conditions on encoder f ensures a perfect accuracy? Choose one:

i.
$$\forall i, \quad \exists j \neq i \quad \text{such that} \quad \mathrm{diameter}_i(f) < \mathrm{margin}_{i \to j}(f)$$

(i.e., for all distributions i, its diameter is smaller than the margin between i and **some** other j.)

ii.
$$\forall i \neq j, \quad \mathrm{diameter}_i(f) < \mathrm{margin}_{i \to j}(f)$$

(i.e., for all distributions i, its diameter is smaller than the margin between i and **all** other j.)

iii.
$$\exists i^* \quad \text{such that} \quad \mathrm{diameter}_{i^*}(f) < \min_{i \neq j} \mathrm{margin}_{i \to j}(f)$$

(i.e., the diameter of i_* is smaller of any margin between two different distributions.)

(d) **(2pt; Alignment, Invariance, Margin, and Uniformity)** Consider the case where p_i's have disjoint supports (i.e., S_i's). Answer the following questions:

i. **(0.5pt; Alignment)** To more easily satisfy the condition you obtained in Question 4c, should we have larger or smaller $\mathrm{diameter}_i$'s?

ii. **(0.5pt; Margin)** To more easily satisfy the condition you obtained in Question 4c, should we have larger or smaller $\mathrm{margin}_{i \to j}$'s?

iii. **(1pt)** Consider an optimal encoder f^* that extremizes $\mathrm{diameter}_i$'s and $\mathrm{margin}_{i \to j}$'s based on your answers above.

A. **(0.5pt; Invariances)** We say that an encoder f is *invariant* to variations within a set $S \subset \mathcal{X}$ if $f(x)$ is constant $\forall x \in S$.

What are the *invariances* that f^* learn (if any)? Provide your answer in the form of a collection of **sets** $\subset \mathcal{X}$, where f is invariant to variations within each, and these sets are "largest" in the sense that f^* is **not** invariant to the union of any two of them.

B. **(0.5pt; Uniformity)** In geometry, a well known problem is to find max-margin placements of N unit-vectors on $\mathbb{S}^{d-1}$ [Tammes, 1930], whose solution is "roughly uniformly" placing the points $\mathbb{S}^{d-1}$.

Consider the above Tammes' problem and the problem of extremizing diameter_i's and $\text{margin}_{i \to j}$. Discuss the relation between them in 1-3 sentences.

(e) **(1pt; Overlapping Distributions) Without the assumption that** p_i's have disjoint supports (i.e., S_i's). Answer the following questions:

- **(0.5pt)** Can the condition you obtained in Question 4c be satisfied with possible overlapping supports?
- **(0.5pt)** What will happen to two p_i and p_j that have overlapping support:

$$\text{supp}(p_i) \bigcap \text{supp}(p_j) = S_i \bigcap S_j \neq \varnothing, \qquad i \neq j \tag{5}$$

Provide 1-2 sentence description on how their features are placed by an optimal encoder f.
Your answer doesn't need to provide an exact characterization, but should mention the distance between feature vectors from these two distributions.

Remark: This accuracy-based analysis already reveals how contrastive learning groups features together based on the positive pair distributions. In particular, Question 4d says that contrastive learning might prefer tight clusters that are uniformly placed in the representation space. While many assumptions are used to obtain these insights, a more involved analysis shows that they hold true generally [Wang and Isola, 2020].

5. **(2pt; Normalization in Contrastive Learning)** Consider an encoder f with perfect accuracy on the data distributions (as defined in Question 4). Answer the following questions:

- **(1pt)** Does it achieve zero contrastive loss (Equation (1))?
- **(1pt)** Suppose we relax the constraint that encoders must output to $\mathbb{S}^{d-1}$, and instead allow them to output any value $\in \mathbb{R}^d$. Consider optimizing such encoders to minimize the contrastive loss.
  - **(0.5pt)** Is there an encoder that attains arbitrarily small contrastive loss?
    Hint: One approach, temperature scaling divides the logits by a scalar parameter $softmax = e^{z/T} / \sum_i e^{z_i/T}$. Think about the effect of logit scaling on cross-entropy classification loss. For more information on temperature scaling, see `https://arxiv.org/pdf/1706.04599.pdf`
  - **(0.5pt)** Are there any potential numerical stability issues?

6. **(1.5pt; Cross-Entropy Supervised Learning as Dual-Encoder Contrastive Learning)**

In fact, the familiar cross-entropy supervised classifier learning can be exactly cast as a Dual-Encoder contrastive learning method.

Dual-Encoder Contrastive Learning: Consider **two data domains** $\mathcal{X}_1$ and $\mathcal{X}_2$ (e.g., images and text), **two encoders**

$$f_1 \colon \mathcal{X}_1 \to \mathbb{R}^d \tag{6}$$
$$f_2 \colon \mathcal{X}_2 \to \mathbb{R}^d, \tag{7}$$

and positive pair distributions $\{p_i\}_{i=1}^N$, where each p_i is a distribution over the **product space** $\mathcal{X}_1 \times \mathcal{X}_2$. Define the following form of **Dual-Encoder** contrastive loss:

$$\mathcal{L}_{\textsf{dual-enc-contr}}(f_1, f_2) \triangleq \frac{1}{N}\sum_{i=1}^N \mathop{\mathbb{E}}_{\substack{\forall j \in \{1,\dots,N\},\\ (x_j^{(1)}, x_j^{(2)}) \sim p_j}} \left[-\log \frac{e^{f_1(x_i^{(1)})^\mathsf{T} f_2(x_i^{(2)})}}{\sum_{j=1}^N e^{f_1(x_i^{(1)})^\mathsf{T} f_2(x_j^{(2)})}} \right], \tag{8}$$

where $(x_i^{(1)}, x_i^{(2)})$ is the *positive pair* and, for $j \neq i$, $(x_i^{(1)}, x_j^{(1)})$ are the *negative pairs*.

This is referred to as *cooccurrence-based* contrastive learning in notes.

Remark : We use unconstrained encoder (output $\in \mathbb{R}^d$) to make connections precise. In practice, dual-encoder methods usually still use normalized outputs $\in \mathbb{S}^{d-1}$.

Assume a labelled dataset for classification $\mathcal{D}_{\textsf{labelled}} \triangleq \{(x_k, y_k)\}_{k=1}^K$, where $x_k \in \mathcal{X}$ are data samples, and $y_k \in \{1, 2, \dots, C\}$ are labels from C evenly-represented classes.

With standard supervised training, consider optimizing a **deep classifier** $g \colon \mathcal{X} \to \mathbb{R}^C$ w.r.t. cross-entropy loss on $\{x_k, y_k\}_{k=1}^K$:

$$\mathcal{L}_{\textsf{xce}}(g) \triangleq \mathop{\mathbb{E}}_{x_k, y_k \sim \mathcal{D}_{\textsf{labelled}}} \left[-\log \underbrace{\frac{e^{g(x_k)[y_k]}}{\sum_c e^{g(x_k)[c]}}}_{\textsf{softmax}(g(x_k))[y_k]} \right]. \tag{9}$$

Let's refer to this training procedure as **supervised cross-entropy classification**, where we say that g correctly predicts on pair (x, y) iff

$$\mathop{\arg\max}_{c \in \{1,2,\dots,C\}} \underbrace{g(x)}_{\textsf{logits}}[c] = y, \quad \text{where } \arg\max \text{ is unique.} \tag{10}$$

Compare it with the **dual-encoder contrastive learning** procedure, and answer the following questions in write-up:

- **(1pt)** Provide the exact forms of N, $\{p_i\}_i$, d, $\mathcal{X}_1$, $\mathcal{X}_2$, f_1, and f_2, so that training g w.r.t. $\mathcal{L}_{\mathsf{xce}}$ exactly corresponds to training f_1 and f_2 w.r.t. $\mathcal{L}_{\mathsf{dual\text{-}enc\text{-}contr}}(f_1, f_2)$.

  Hint: Think about how to structure the cross entropy loss as a dual encoder contrastive loss problem.

- **(0.5pt)** Show that under this view, the supervised cross-entropy classification accuracy of g coincides with the contrastive loss accuracy defined in Question 4.

Reconstruction and Similarities in Representation Learning (12pt)

We consider performing unsupervised representation learning on a dataset of colored shapes. Each image has resolution 64×64 . Across the dataset, there are three underlying varying factors, as shown in Figure 2.

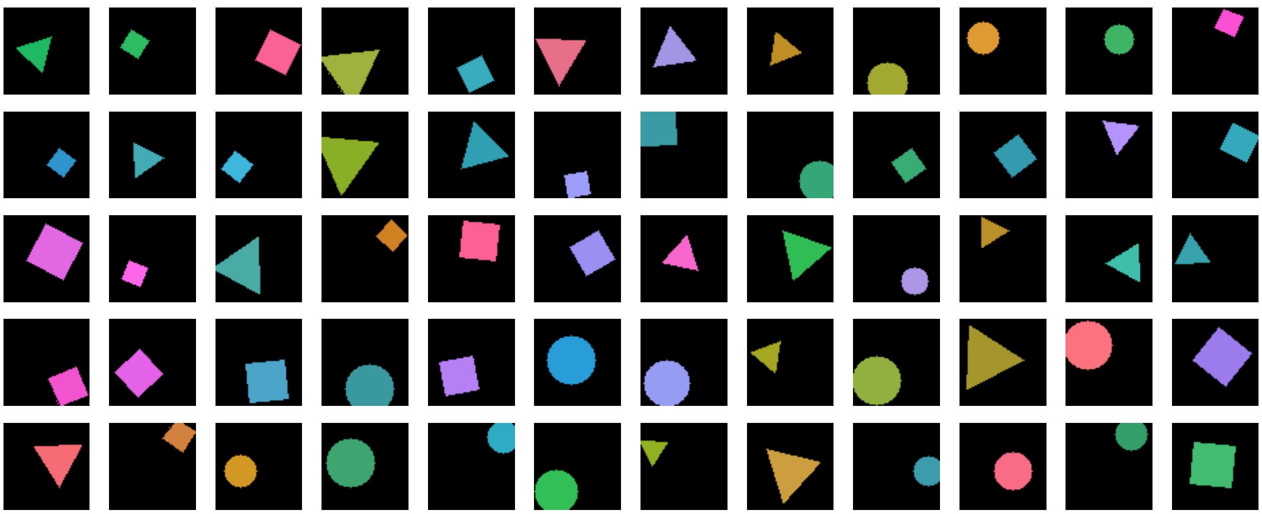


Figure 1: Random unconditional samples of the dataset.

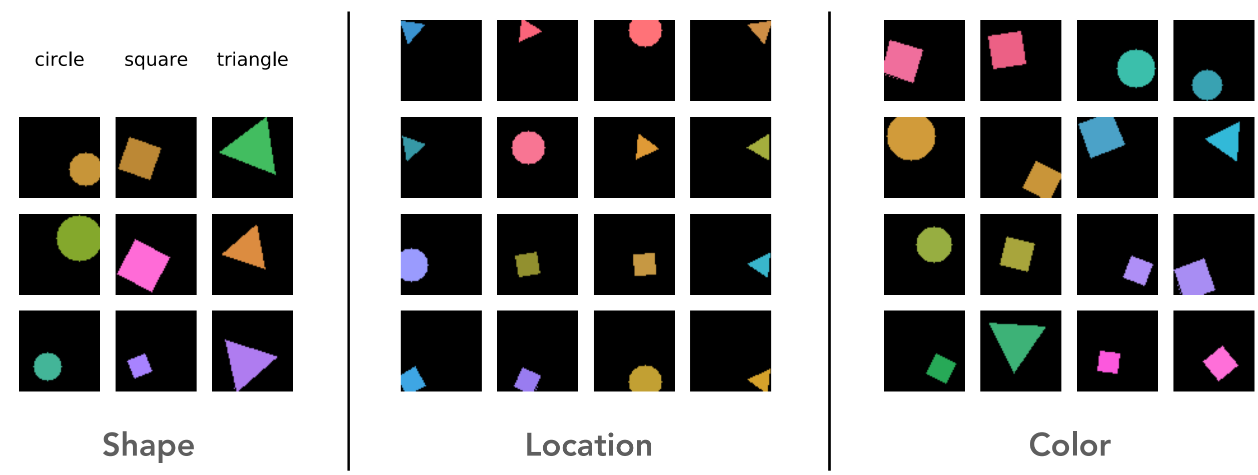


Figure 2: Three types of variations in the dataset.

We will use a finite training dataset of size N, denoted as

$$\mathcal{D}_{\text{train}} \triangleq \{x_i\}_{i=1}^{N}, \tag{11}$$

where each $x_i \in \mathcal{X}$ is such an image. The goal of representation learning is to optimize an encoder function $f\colon \mathcal{X} \to \mathbb{R}^d$, so that $f(x)$ is a good representation/latent/embedding/encoding of x. Here $\mathbb{R}^d$ is called *the representation space*, with usually a small d aiming for a compact representation of the dataset that captures important features.

Instruction: We will use this notebook. There are only a few `FIXME` place where you need to fill in and submit the code. Majority of the code are already provided. You are not required to fully read through the implementation of them, but should conceptually understand what the functions/classes with **docstrings** are doing. You will be asked to attach certain plots, and provide certain discussions about the results in your final write-up. The training time for some contrastive encoders can be close to 1 hour, while the autoencoder and some other contrastive encoders are much faster. **Please include all code as a PDF attached to the end of your HW submission**

7. **(6pt; Autoencoder)** Autoencoder objective involves two functions:

$$f\colon \mathcal{X} \to \mathbb{R}^d \qquad \text{(encoder)}$$
$$g\colon \mathbb{R}^d \to \mathcal{X}. \qquad \text{(decoder)}$$

The objective is reconstruction, saying that g should be able to fully reconstruct input data x, by only looking at the representation $f(x)$:

$$\mathcal{L}_{\mathsf{AE}}(f; g) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathsf{MSE}(g(f(x)), x)}_{\text{mean squared error between reconstruction and original } x}. \tag{12}$$

(a) **(3pt; Theoretical Optimum)** Consider two functions f^* and g^*. All we know is that they perfectly solves the Autoencoder objective in Equation (12) on the finite dataset $\mathcal{D}_{\mathsf{train}} \triangleq \{x_i\}_{i=1}^N$, where all samples are assumed to be **distinct** and samples from some p_x.

Answer each of the following questions with one of {"Yes", "No", "Maybe"} and briefly explain your answer (approx. 1 sentence).

i. **(0.5pt)** For some $x_i \in \mathcal{D}_{\mathsf{train}}$, can its dataset index i be decided by $f^*(x_i)$ alone?

(If a property of x can be written as a function of $f^*(x)$, it can be decided by $f^*(x)$ alone.)

ii. **(0.5pt)** For some $x_i \in \mathcal{D}_{\mathsf{train}}$, can the color of x_i be decided by $f^*(x_i)$ alone?

iii. **(0.5pt)** For three $\{x, y, z\} \subset \mathcal{D}_{\mathsf{train}}$, where x and y are both red, and z is blue. Compared to x and z, are x and y closer in representation space?

I.e., is the following true?

$$\|f^*(x) - f^*(y)\|_2 < \|f^*(x) - f^*(z)\|_2. \tag{13}$$

iv. **(0.5pt)** For random $x \sim p_x$, can the color of x be decided by $f^*(x)$ alone (**Hint**: For the technical reader, we are asking if this can be decided with probability 1. Same for the following questions.)?

v. **(0.5pt)** For random $x \sim p_x$, can the color of x be decided by the colors of $f^*(x)$'s nearest neighbors (in training set) alone?

(To find such nearest neighbors, we compute representations of all training samples, and take the samples whose representations are closest to $f^*(x)$.)

vi. **(0.5pt)** For random $x \sim p_x$, can the background color of x be decided by $f^*(x)$ alone?

(b) **(1pt; $\mathcal{L}_{\mathsf{AE}}$ Implementation)** In colab, understand the parts from beginning to `train_autoencoder`. Implement the `FIXME` in `train_autoencoder` that computes MSE reconstruction loss within 5 lines of code.

Attach your code to write-up.

(c) **(1pt; Visualize Encoders via Nearest Neighbors)** In colab, we provide a function `nn_visualize` that visualizes the nearest neighbors of training samples and/or unseen validation samples, where the nearest neighbors are selected based on embedding distances of a learned encoder.

Run the provided code cells to

i. Train an encoder-decoder pair using Autoencoder objective;

ii. Visualize the trained encoder via the nearest neighbors for training samples and validation samples.

Attach your two plots to write-up.

(d) **(1pt; Understand Nearest Neighbors)** Based on the nearest neighbors plots you obtain above, answer the following questions in write-up:

i. Is the representation meaningful for training images? Validation images? In particular, are similar samples grouped together?

ii. The Autoencoder objective alone (on two generic functions) doesn't enforce any grouping or smoothness of the representation space. If you observe some clusters of similar images, why do you think it happens? If you do not, write down some ideas of encouraging grouping/smoothness.

8. **(6pt; Contrastive Learning)** In this part, we consider the more standard contrastive loss (not the one we used above for theoretical analysis).

For encoder $f \colon \mathsf{data} \to \mathbb{S}^{d-1}$ and a temperature hyperparameter $\tau > 0$, the contrastive loss we use is:

$$\mathcal{L}_{\mathsf{contr}}(f;\tau) = \mathbb{E}_{\substack{(x,x^+)\sim p_{\mathsf{positive}} \\ (y_1^-,\dots,y_K^-) \overset{\text{i.i.d.}}{\sim} \mathcal{D}_{\mathsf{train}}}} \left[-\log \frac{e^{f(x)^T f(x^+)/\tau}}{e^{f(x)^T f(x^+)/\tau} + \sum_{k=1}^K e^{f(x)^T f(y_k^-)/\tau}} \right], \tag{14}$$

where p_{positive} is the distribution of positive pair. Here the pairs are

$$\begin{aligned} &(x, x^+) && && \text{(positive)} \\ &(x, y_k^-) && k \in \{1, 2, \dots, K\}. && \text{(negative)} \end{aligned}$$

Notable things about implementation:

- Encoder f is required to output l_2-normalized feature vectors.
- τ is a (fixed) temperature hyperparameter, often selected as $\tau \in [0.05, 0.3]$. We use $\tau = 0.07$ in code.
- As common in image contrastive learning, the positive pairs p_{positive} are defined as two random transformed versions of the **same** underlying data sample, where the negative samples are random transforms of **different** data samples.
- To efficiently sample features of negative pairs, we do not independently sample negatives for each positive pair. Instead, we only fetch a batch of positive pairs $\mathbf{x}$ (of b samples) and $\mathbf{x}^+$ (of b samples), where for each $\mathbf{x}[i]$,
  - $(\mathbf{x}[i], \mathbf{x}^+[i])$ forms the positive pair
    (i.e., two random augmentations of the same underlying image).
  - For $j \neq i$, $(\mathbf{x}[i], \mathbf{x}^+[j])$ form the $(b-1)$ negative pairs

  This means that, for encoded latents

  $$f(\mathbf{x}) \in \mathbb{R}^{b\times d} \text{ and } f(\mathbf{x}^+) \in \mathbb{R}^{b\times d}, \tag{15}$$

  the logits (i.e., pairwise dot products) can be efficiently computed using a single matrix multiplication.
  This is a common trick.

(a) **(1pt; $\mathcal{L}_{\mathsf{contrastive}}$ Implementation)** In colab, follow the implementation notes above, and implement the `FIXME` in `train_contrastive` that computes contrastive loss within 5 lines of code.

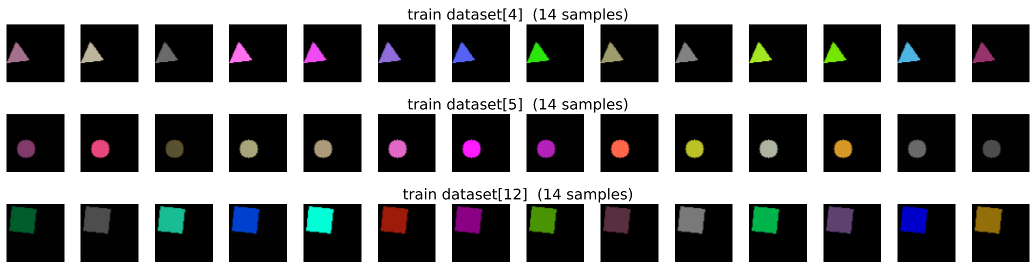
Attach your code to write-up.

(b) **(3pt; Augmentation and Invariances)** Recall from lecture and our analysis in Question 4d that the positive pair distributions define what the learned representation should be invariant to, while sensitive to variances in all other factors (i.e., other factors are preserved in the representations). (⋆)

Here, our positive pairs are generated by random augmentations. We consider three sets of augmentations, visualized below.

In write-up, for each set of augmentations, answer whether the learned representation would be invariant or sensitive to **shape**, **location** and **color**. Base your answer on the ($\star$) analysis results (not on empirically trained encoders).

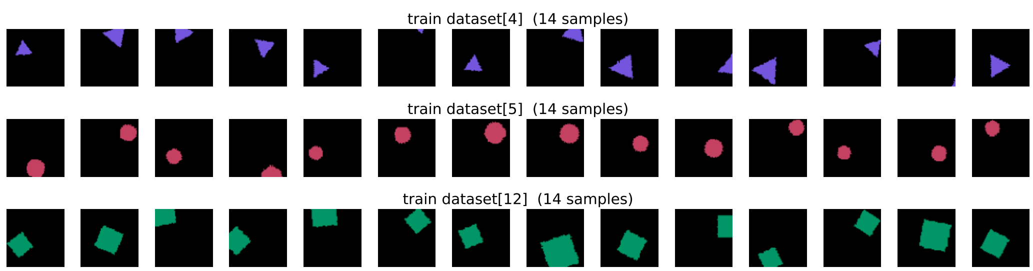
i. **(1pt) Samples from Augmentation A:**



(Each row shows 14 random augmentations of the same underlying image.)

Will contrastive encoder learned w.r.t. **Augmentation A** be invariant or sensitive to **shape**? What about **location**? What about **color**?

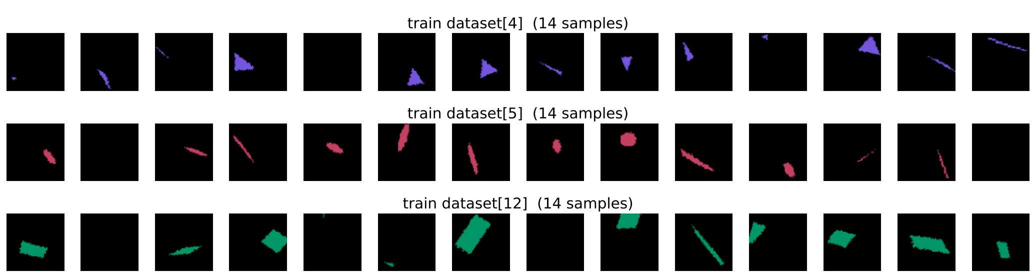
ii. **(1pt) Samples from Augmentation B:**



(Each row shows 14 random augmentations of the same underlying image.)

Will contrastive encoder learned w.r.t. **Augmentation B** be invariant or sensitive to **shape**? What about **location**? What about **color**?

iii. **(1pt) Samples from Augmentation C:**



(Each row shows 14 random augmentations of the same underlying image.)

Will contrastive encoder learned w.r.t. **Augmentation C** be invariant or sensitive to **shape**? What about **location**? What about **color**?

(c) **(1pt; Visualize Encoders via Nearest Neighbors)**

In colab, use provided code to

- Train **three** contrastive encoders, one for each set of augmentations
- Visualize the learned encoder via nearest neighbors.

Attach the three nearest neighbor plots (3 encoders on only val set) to write-up.

(d) **(1pt; Understand Nearest Neighbors)**

In write-up, answer the following questions:

i. For **Augmentation B**, are the learned encoder sensitive to **shape**? To **color**? To **location**?

ii. Are the nearest neighbor plots consistent with your answer in Question 8b?

iii. If they are not in some cases, think about which factors are easier/harder for the encoder network to learn, and write-down some reasons on why this discrepancy happens.

References

Pieter Merkus Lambertus Tammes. On the origin of number and arrangement of the places of exit on the surface of pollen-grains. *Recueil des travaux botaniques néerlandais*, 27(1): 1–84, 1930.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.