Dynamic Latent Separation for Deep Learning

Yi-Lin Tuan 1 Zih-Yun Chiu 2 William Yang Wang 1

Abstract
A core problem in machine learning is to learn expressive latent variables for model prediction on complex data that involves multiple sub-components in a flexible and interpretable fashion. Here, we develop an approach that improves expressiveness, provides partial interpretation, and is not restricted to specific applications. The key idea is to dynamically distance data samples in the latent space and thus enhance the output diversity. Our dynamic latent separation method, inspired by atomic physics, relies on the jointly learned structures of each data sample, which also reveal the importance of each sub-component for distinguishing data samples. This approach, atom modeling, requires no supervision of the latent space and allows us to learn extra partially interpretable representations besides the original goal of a model. We empirically demonstrate that the algorithm also enhances the performance of small to larger-scale models in various classification and generation problems.

1. Introduction
Deep neural networks with multiple hidden layers are trained to be expressive models that learn complicated relationships between their inputs and outputs (Srivastava et al., 2014). Among various data types, data samples that consist of many sub-units, such as images and texts, can require models to be more expressive to consider nuanced differences among sub-units. The demand for this delicacy leads to developing large-scale and complex model architectures (Vaswani et al., 2017), which cause drawbacks such as compromised model interpretability (Ribeiro et al., 2016; Bastani et al., 2017; Rudin, 2019; Jain & Wallace, 2019).

Various algorithms exist that improve model expressiveness not by advancing model architectures. For instance, contrastive learning ameliorates classification expressiveness (Dosovitskiy et al., 2014; Chen et al., 2020) by pushing away latent features from different classes. Vector quantization tackles the expressiveness of autoencoders (Van Den Oord et al., 2017) by learning discrete representations using a preset codebook. As a separate effort, post-hoc methods or designed models that follow self-explaining protocol (Alvarez-Melis & Jaakkola, 2018) reveal some underlying reason for model behaviors. While these methods show promising results in their bundled applications, it is yet certain of their transferability and usefulness to other applications. Meanwhile, it is yet underexplored of generalizable training algorithms that can simultaneously help expressiveness and uncover partial explanations.

We present a novel algorithm that simultaneously improves model expressiveness, provides an interpretation of sub-component importance, and is generalizable to multiple applications. Our method, atom modeling, first maps the latent representations of each sub-component in a data sample to a learnable token importance and then dynamically distances data samples based on token importance using a loss function inspired by Coulomb force (Coulomb, 1785). After training, token importance reveals which sub-components in a data sample contribute to its semantic meaning and are key to distinguishing itself from other data samples. The dynamic separation between data samples encourages a model to predict diverse outputs, thus boosting expressiveness.

This method can be viewed as connecting sub-component importance and inter-sample relationships to elevate impacts from local details. A similar observation can be found in atomic physics, where the balance distance between atoms, fundamental particles that form every matter in nature, depends on the structure of sub-atomic particles in each atom (Brown, 2009; Halliday et al., 2013). In addition, applying atom modeling in a neural network also amounts to regularizing the representation space to preserve each data sample’s uniqueness. Finally, atom modeling promotes expressiveness using a loss function with no latent supervision, enabling it to be flexibly applied to different applications.

We demonstrate the utility of atom modeling objective functions by training or finetuning convolution neural networks, generative adversarial networks, and transformers on Gaussian mixtures, natural texts (CoLA, Poem), and nat-
2. Related Work

Atom modeling can be interpreted as a way of learning representation by spacing data samples. The idea of keeping a distance among data samples has previously been used in manifold learning (Tenenbaum et al., 2000; Saul & Roweis, 2003; Cayton, 2005; Lin & Zha, 2008), graph representations (Perozzi et al., 2014; Grover & Leskovec, 2016; Hamilton et al., 2017), kernel tricks (Mueller et al., 2001; Keerthi & Lin, 2003; Hofmann et al., 2008), and contrastive learning (Weinberger & Saul, 2009; Gutmann & Hyvärinen, 2010; Sohn, 2016; Oord et al., 2018; Chen et al., 2020) where representations are trained to fit a predefined inter-sample relationship. For instance, contrastive learning, the most related one to our method, requires preset negative and positive pairs in order to push away opposing pairs while bringing together the positive pairs. These methods necessitate prior knowledge of inter-sample relationships.

Since atom modeling can be seen as a technique to discretize representations within a continuous space, it is natural to consider its discrete space counterpart: vector quantized variational autoencoder (Van Den Oord et al., 2017; Razavi et al., 2019; Esser et al., 2021), which maps encoder output to an additional codebook with preset number of codes as the way of discretization. This shows success in reconstruction but is not easy to generalize to other models. In comparison, atom modeling that promotes separation of the original continuous embedding space frees the restriction on the preset number of codes and the autoencoder architecture.

While atom modeling leverages fine-grained component importance to determine the balanced data sample distances, it also provides model-agnostic partial interpretation that is orthogonal to extensions of model-dependent self-interpretable designs (Alvarez-Melis & Jaakkola, 2018) and post-hoc explanation methods (Ribeiro et al., 2016). However, this work does not focus on explanation but demonstrates an outgrowth of atom modeling.

3. Method

Our goal is to define a flexible method that makes a model more expressive for data samples with multiple sub-units, such as images and texts, and does not need latent space supervision. We say a model is expressive if it can accommodate various distinct outputs for different inputs.

We define a model in a general form:

\[ y = f_\theta(z), z \sim D, \]

where \( D \) is the data distribution or a random noise distribution. We can easily fit models for practical applications, such as generation or classification, into this form: \( y \) is often a real vector \( y \) for generation and a probability distribution \( P(Y) \) for classification. If \( f_\theta \) is expressive, different \( z \) is more likely to give different \( y \) or \( P(Y) \). This distinction is desirable for promoting diversity in generation models (Razavi et al., 2019) and encouraging entropy in classification models (Dubey et al., 2018).

To achieve the goal of distinct outputs, we first write a model in its composite form:

\[ f_\theta(\cdot) = o(\ell(\cdot)), \]

where \( \ell(\cdot) \in \mathbb{R}^{N \times h} \) gives the latent representation of an input, and \( o(\cdot) \) outputs the result given the latent representation. \( N \) is the number of sub-components, and \( h \) is the dimension of the latent space. An intuitive way to increase the probability that \( f_\theta(z^A) \) differs from \( f_\theta(z^B) \) is to let \( \ell(z^A) \) distance from \( \ell(z^B) \). Here, we show the properties of the output function, \( o(\cdot) \), that lead to this concurrent increase.
Figure 2. $L_A$ with varied atomic structure similarity $k$. The distance having the minimum loss depends on the intra-sample structures. As the structures are more similar (decayed $k$), the minimum loss distance becomes larger. Simultaneously, the distance cannot be zero.

Lemma 1. A $G$-Lipschitz function $o(\cdot)$ and a $K$-Lipschitz inverse function of $o(\cdot)$ returns the output space distance such that:

$$K\|v - u\| \leq \|o(v) - o(u)\| \leq G\|v - u\| ,$$

where $v$ and $u$ are any vector in the latent space.

Equation 3 indicates that if the latent distance increases, the bounds of output distance also increase. All proofs in this paper are in appendix A.

The next challenge is that, in a general case without latent space supervision, how we should set apart the latent variables produced by $\ell(\cdot)$ and $e(\cdot)$. We propose to dynamically distance latent representations by separating the currently close variables and neglecting the already distant variables. Whether the variables are close or distant depends on their intra-sample structures. This leads us to first map a latent variable to a new embedding space by a learnable mapping function $A(\cdot)$ such that:

$$\{(q_i, p_i)\}_{i=1}^N = A(\ell(z)) ,$$

where $q_i \in \mathbb{R}$ is the importance score of the $i$-th row (token) in $\ell(z)$, and $p_i \in \mathbb{R}^{h'}$ is the position of the same token in the new space.

Then, we propose a dynamic distancing loss function:

$$L_A = E_{x^A,x^B \sim D} \sum_{i \in A,j \in B} d(q_i^A q_j^B, p_i^A p_j^B) ,$$

where $d(q_i^A q_j^B, p_i^A p_j^B) \in \mathbb{R}$ is a distance between $i$-th and $j$-th tokens in $z^A$ and $z^B$ and is derived from their intra-sample structures. We also use $A$ and $B$ as sets $\{1, \cdots, N_A\}$ and $\{1, \cdots, N_B\}$.

By minimizing $L_A$ in Equation 5, the optimal distance between $\ell(z^A)$ and $\ell(z^B)$ cannot be 0. That is, our proposed atomic loss forces $\ell(z^A)$ and $\ell(z^B)$ to be apart. In addition, the optimal distances are not identical for different data pairs, and these optimal values depend on each data’s intra-sample structure. Figure 2 shows examples of the atomic loss function and optimal distances.

3.1. Token Importance

In Equation 5, $q_i^A, q_j^B \in \mathbb{R}$ is a learnable importance score of a token in a data sample $A$. Given a latent representation of $A$, $\ell(z^A) = [e_1^A e_2^A \cdots e_{N_A}^A] \in \mathbb{R}^{N_A \times h}$, we define the token importance as:

$$q_i^A = 2\sigma(Q(e_i^A)) - 1 \in [-1, 1] ,$$

where $\sigma(\cdot)$ is the sigmoid function, and $Q(\cdot) : \mathbb{R}^h \rightarrow \mathbb{R}$ maps the original h-dimension latent variable to an unnormalized importance score. We rescale the score to $[-1, 1]$ as it is a simple way to have three types of multiplication $q_i q_j$ needed in Equation 5: polarity (negative), likeness (positive), and no effect (zero). Since only when $q_i$ is not zero, $q_i q_j$ attributes to $L_A$, one role of token importance is as asking if the $i$-th token in a data sample makes it distinguishable from other data samples. As shown in Table 1, token importance +1 or -1 helps distinguish data while 0 does not.

### Table 1. Interpretation of token importance $q_i$

| Distinguish data? | Attribute to Meaning? |
|-------------------|-----------------------|
| Yes               | +1                    |
| No                | -1                    |
| Not considered    | 0                     |

3.2. Atomic Distance

We define the atomic distance between data samples $A$ and $B$ by:

$$d_{AB} = \|\mu_A - \mu_B\|_p ,$$

and the distances among their $i$-th and $j$-th tokens $d(q_i^A, q_j^B, p_i^A, p_j^B) \in \mathbb{R}$ in Equation 5 or $d_{ij}$ for brevity by:

$$d_{ij} = d_{AB} + \frac{r_A + r_B}{2} \text{step}(-q_i^A q_j^B), \forall i \in A, j \in B .$$

Here $\mu_A$ and $\mu_B$ are respectively an average of the most crucial tokens of $A$ and $B$, $r_A$ and $r_B$ are the token deviation within a data sample, and $\text{step}(\cdot)$ is the step function. We formally define them as:

$$\mu_A = \frac{1}{N_A} \sum_{i \in A} m_i^A p_i^A ,$$

$$r_A = \frac{1}{N_A} \sum_{i \in A} (1 - m_i^A)\|p_i^A - \mu_A\|_p ,$$

$$m_i^A = 1 - \max(-q_i^A, 0) \in [0, 1] ,$$

$$p_i^A = P(e_i^A) \in \mathbb{R}^{h'} ,$$
where \( P(\cdot) : \mathbb{R}^h \rightarrow \mathbb{R}^{h'} \) maps the original latent variable to the position of the \( i \)-th token in a new space, and \( m_i^A \in \mathbb{R} \) is the mass of \( i \)-th token, indicating how much the token attributes to the meaning of \( A \). As shown in Table 1, tokens with \( q_i \) being +1 or 0 play a key role in the data meaning and have \( m_i = 1 \), while tokens with importance -1 are less likely to attribute to data meaning.

Since Equation 5 optimizes the atomic distance \( \bar{d}_{AB} \) that is not a conventional distance metric, we show its relationship to Euclidean distance.

**Theorem 1.** Consider equal token importance distribution, Equation 7 returns the atomic distance such that:

\[
\bar{d}_{AB} \leq C\|\bar{\nu}^A - \bar{\nu}^B\|_2, \tag{13}
\]

where \( \bar{\nu}^A \) is a permutation of \( \ell(z^A) \) from data sample \( A \).

Theorem 1 implies that rising \( \bar{d}_{AB} \) encourages separation of \( \ell(z^A) \) and \( \ell(z^B) \) in the Euclidean space. Therefore, according to Lemma 1, rising \( \bar{d}_{AB} \) can increase the bounds of \( \|o(\ell(z^A)) - o(\ell(z^B))\|_2 \).

Next, we show that by optimizing Equation 5, the distance between two data samples depends on how similar their atomic structures are. In other words, the inter-sample relationship depends on the intra-sample structures. This dependency is crucial to achieve dynamic distance among data samples.

**Theorem 2.** Let \( c = \sum_{q_i,q_j>0} q_i q_j \) and \( c^* = \sum_{q_i,q_j<0} q_i q_j \) for all \( i \in A \) and \( j \in B \); given data samples \( A, B \). Without loss of generality, \( c^* = kc \) and \( k \in (1, \infty) \) gives the optimal atomic distance in Equation 5 as:

\[
\bar{d}_{AB}^* = \left( \frac{r_A + r_B}{2} \right) \sqrt{\frac{k+1}{k-1}} \tag{14}
\]

If \( k \to 1 \), \( \bar{d}_{AB}^* \to \infty \) and \( k \to \infty \), \( \bar{d}_{AB}^* \to 0 \).

Theorem 2 shows that optimizing Equation 5 forces data samples with similar intra-sample structures to separate more than the ones with dissimilar structures. Hence, our method results in dynamic distancing.

### 3.3. Training

We further prevent the model from learning every tokens equally important using a soft constraint on the token importance distribution, which is essential to form reasonable intra- and inter-sample relationship. We regularize the number of tokens with different importance scores to be similar. The complete loss function of Equation 5 becomes:

\[
\mathcal{L}_A = E_{A,B \sim D} \sum_{i \in A,j \in B} q_i^A q_j^B d_{ij} + \left( \sum_{i \in A} q_i \right)^2 + \left( \sum_{i \in A} q_i^2 - \frac{2}{3} N_A \right)^2 \tag{15}
\]

Algorithm 1 lists the complete training process.

### 3.4. Relation to Atomic Physics.

Our proposed method has high correspondence with atomic physics (Halliday et al., 2013; Brown, 2009), the scientific study of the structure of an atom and its interaction with others. Therefore, we name this method *atom modeling*.

Among atoms in nature, there are *inter-atomic* forces, similar to our proposed loss function in Equation 5, that bind atoms and avoid them collapsing by maintaining a balanced distance. The distances among a group of atoms depend on their *atomic structures*, similar to our theoretical result in Theorem 2.

Within an atom, its structure consists of three types of particles: neutrons, protons, and electrons, where a neutron has no charge, a proton has one positive charge with similar weight as a neutron, and an electron has a negative charge and weight significantly less than a proton (Mohr et al., 2008). The charges correspond to our token importance.

Simultaneously, Bohr (1913) introduced one way to describe an atomic structure, where the protons and neutrons form a *nucleus*, similar to \( \mu \), that occupies a small volume of the atom while the electrons orbit around the nucleus with a *radius*, similar to \( r \).

The soft constraint in Equation 15 is also similar to that the protons, electrons, and neutrons often have similar numbers within one atom.

### 4. Experiments

We test atom modeling’s effects and flexibility by training linear classifiers on synthetic data, GANs on unconditional image generation, ResNets on image classification, and transformers on text classification.

In the experiments, while \( Q(\cdot) \) and \( P(\cdot) \) can be any mapping functions, we use simple extraction functions with a selected hidden layer such that \( Q(\cdot) \) extracts one dimension from the original \( \ell(\cdot) \in \mathbb{R}^h \) and \( P(\cdot) \) extracts the rest \( h - 1 \) dimensions. More discussion is in appendix B.
Dynamic Latent Separation for Deep Learning

Cross-Entropy  Hinge (1-norm)  Hinge (2-norm)  SimCLR  Atom Modeling

(a) inter-sample relationship in latent space
(b) atom modeling
token importance

Figure 3. (a) Visualization of the latent space of synthetic data by only cross-entropy training loss or integrated with hinge losses (L1 and L2), SimCLR, or Atom Modeling. Blue and red indicates the two ground-truth classes. The dashed circles annotate the overlaps of the learned representations from different classes, which is correlated with the easiness to classify the samples. Atom modeling separates representations with a gap using no latent supervision. (b) Visualization of token importance.

4.1. Linear Classifier on Synthetic Data

To demonstrate the shifts of inter-sample relationships after atom modeling, we first conduct experiments on synthetic data. We mimic data of multiple sub-components by generating input features $X$ and the corresponding labels $y$ as follows:

$$
\begin{align*}
P(A) & = N(\mu_a, \sigma_a) \\
P(B) & = N(\mu_b, \sigma_b) \\
X & = \{x_i| x_i \in A \cup B\}_{i=1}^N \\
y & = 1 (P(x_i \in A|x_i \in X) > P(x_i \in B|x_i \in X))
\end{align*}
$$

(16)

where $A$ and $B$ are two events of normal distributions with $(\mu_a, \sigma_a)$ and $(\mu_b, \sigma_b)$ being the mean and standard deviation respectively. $X$ is the input composed of $N = 5$ sub-units $x_i$ sampled from $A \cup B$. The goal is to find a function $f : X \rightarrow y$.

In this experiment, we use a neural network with two fully-connected linear layers and apply atom modeling to the hidden state of the first layer. For comparison, we employ Hinge loss with p-norm distances (Bromley et al., 1993; Chopra et al., 2005; LeCun et al., 2006) and SimCLR (Chen et al., 2020) that uses cosine similarity as the metric on the same hidden state, as our baselines. Figure 4 shows the classification accuracy across ten random runs. Atom modeling enhances the classifier to achieve an average of 96% accuracy and is superior to baselines.

We visualize how atom modeling alters inter-sample relationships and learns token importance. Figure 3(a) demonstrates that atom modeling spreads out the representation distribution, especially the high-density region. This further creates a gap between the blue and red classes in the latent space, thus enhancing the classifier expressivity. The corresponding token importance is plotted in Figure 3(b). The model takes sub-units near the $\mu_a$ ($q = +1$) and $\mu_b$ ($q = -1$) as the most crucial ones to distinguish data samples and takes sub-units near $\mu_a$ ($q = +1$) and $(\mu_a + \mu_b)/2$ ($q = 0$) as the keys to data meaning.

4.2. GANs on Unconditional Image Generation

We investigate atom modeling on generative models with unconditional image synthesis tasks: MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky et al., 2009), and CelebA-HQ256x256 (Karras et al., 2018). For MNIST and CIFAR10, we performed experiments with DCGAN (Radford et al., 2015). For CelebA-HQ256x256, we performed experiments with the SOTA model, StyleSwin (Zhang et al., 2022a), and followed their implementation.

For comparison, we employed self-supervised contrastive learning (Chen et al., 2020) and vector quantization (Van...
Dynamic Latent Separation for Deep Learning

(a) CelebA-HQ 256x256

(b) MNIST

Figure 5. Examples of generated images and the learned token importance by atom modeling on unconditional image generation. The distributions show that importance score close to one indicates it is a crucial part of the image to distinguish from others.

| Method              | Dataset  | FID   |
|---------------------|----------|-------|
| DCGAN (Radford et al., 2015) | MNIST    | 88.4  |
| VQ (Van Den Oord et al., 2017) | MNIST    | 303.9 |
| SSCL (Chen et al., 2020)      | MNIST    | 69.4  |
| Atom Modeling              | MNIST    | 49.0  |
| DCGAN (Radford et al., 2015) | CIFAR10  | 110.4 |
| VQ (Van Den Oord et al., 2017) | CIFAR10  | -     |
| SSCL (Chen et al., 2020)      | CIFAR10  | 130.0 |
| Atom Modeling              | CIFAR10  | 97.4  |
| VQGAN (Esser et al., 2021)   | CelebA-HQ| 10.2  |
| StyleSwin (Zhang et al., 2022a) | CelebA-HQ| 5.26  |
| Atom Modeling              | CelebA-HQ| 5.18  |

Table 2. FIDs of image synthetic on MNIST, CIFAR10, and CelebA-HQ256x256.

Den Oord et al., 2017) as additional loss to regularize a given representation layer and compared their ability with atom modeling for end-to-end training. In this experiment, for DCGAN, we use the output from the last hidden layer of the generator as the representation to be discretized. For StyleSwin, we use the output from the last hidden layer with half resolution. All the generated results are evaluated by Fréchet Inception Distance (FID) (Heusel et al., 2017; Obukhov et al., 2020) with 2k images for MNIST, 10k for CIFAR10, and 30k for CelebA-HQ256x256. Lower FID indicates higher generation quality.

Empirically, Table 2 demonstrates the effectiveness of atom modeling. The proposed atom modeling outperforms VQ- or SSCL-enhanced DCGAN on MNIST (49.0 vs 69.4) and CIFAR10 (97.4 vs 110.4). Additionally, our approach improves StyleSwin on CelebA-HQ256x256 in our reproduced results (5.18 vs 5.26). Note that this promising improvement is gained under the original setup of DCGAN and StyleSwin without extra tuning. This shows that atom modeling can improve generative model expressivity in flexible settings.

Figure 5 shows examples of our generated images and the associated token importance after applying atom modeling. Areas mapped to hair, eyes, nose, and philtrum in CelebA-HQ256x256, as well as the cores of digits in MNIST, have
Figure 6. Alteration of atomic distance distributions by atom modeling (initial stage, half, end of training) and comparison with standard training criterion. Atom modeling gradually disseminates the atomic distance distribution.

Table 3. Results of fine-grained classification on Oxford-IIIT Pets, Oxford-Flowers102, CoLA, and Poem datasets.

| Method                | Pets Acc | Flowers Acc |
|-----------------------|----------|-------------|
| Cross-Entropy         | 21.0     | 56.7        |
| Hinge (1-norm)        | 20.0     | 54.7        |
| Hinge (2-norm)        | 22.4     | 58.0        |
| Rank-H (Wang & Gupta, 2015) | 22.4     | 58.2        |
| SimCLR (Chen et al., 2020) | 20.7     | 58.1        |
| Atom Modeling         | 22.5     | 59.1        |

Table 4. Results of ImageNet-1K with (He et al., 2016; Khosla et al., 2020) data augmentation approaches.

| Method                                      | Top-1  | Top-5  |
|---------------------------------------------|--------|--------|
| Cross-Entropy w/ (He et al., 2016)          | 74.97  | 92.17  |
| Atom Modeling w/ (He et al., 2016)          | 75.10  | 92.25  |
| Cross-Entropy w/ (Khosla et al., 2020)      | 75.02  | 92.20  |
| Atom Modeling w/ (Khosla et al., 2020)      | 75.19  | 92.35  |

positive importance scores. They play a crucial role in both distinguishing from other images and semantic meaning. Areas map to skin and background have negative importance scores. They are pivotal for differing from others but not the meaning. Other regions are less likely to identify an image but attribute to the semantic meaning. They are, therefore, assigned importance scores of zero.

We plot the atomic distance distribution over training time in Figure 6. At the initial training stage using atom modeling, the distances among data samples are similar to standard training results. The distances in the latent space concentrate to a small value. During training, atom modeling modifies the latent space and gradually disseminates the distance distribution. This matches our expectation of what atom modeling has done during model learning.

4.3. ResNet on Image Classification

We also validate atom modeling on fine-grained image classification Oxford-IIIT Pets (Parkhi et al., 2012) and Oxford-Flowers102 (Nilsback & Zisserman, 2008) as well as ImageNet-1K (Deng et al., 2009) to justify its flexibility.

In a fine-grained classification problem, intra-class diversity is higher than inter-class diversity (Wei et al., 2019), so we can use higher expressivity and, thus, atom modeling.

For comparison, we train ResNet18 (He et al., 2016) with cross-entropy loss, while employing hinge loss with p-norm distances, Rank-H (Wang & Gupta, 2015), SimCLR (Chen et al., 2020). In Table 5, we present the mean top-1 accuracy (Acc). The empirical results show that atom modeling consistently improves cross-entropy and is the best among distancing-representation-like approaches.

We further examine the ability of atom modeling applied to a larger-scale general classification problem on ImageNet-1K. We follow prior work implementations to use ResNet50 (He et al., 2016) as the backbone and run 90 epochs with two data augmentation methods used in (He et al., 2016; Khosla et al., 2020). The first includes only the crop and horizontal
Dynamic Latent Separation for Deep Learning

Figure 8. Visualization of the learned charges by atomic modeling on text classification (COLA). The charge distributions again show that protons are often mapped to the keywords in a sentence.

Table 5. Results of fine-grained classification on Oxford-IIIT Pets, Oxford-Flowers102, CoLA, and Poem datasets.

| Method             | CoLA MCC | Poem F1 |
|--------------------|----------|---------|
| Cross-Entropy      | 60.0     | 60.3    |
| Hinge (1-norm) (LeCun et al., 2006) | 59.8     | 62.0    |
| Hinge (2-norm) (Chopra et al., 2005) | 60.1     | 61.8    |
| SimCSE (Gao et al., 2021) | 60.4     | 60.8    |
| MixCSE (Zhang et al., 2022b) | 60.4     | 60.8    |
| Atom Modeling      | 61.3     | 62.7    |

flip, and the second adds color jitters and grayscale. Table 4 shows that atom modeling improves cross-entropy under different data augmentations, which has been found to impact the results of image classification (Cubuk et al., 2019; 2020; Khosla et al., 2020). Note that this performance gain has only been made by introducing atom modeling to one representation layer and using the exact same setup of conventional training of ResNet50 on ImageNet-1K; with a more elaborate setting, the performance could be improved. More importantly, the outcomes show that atom modeling can be flexibly applied to diverse data, models, and scales.

4.4. Transformer on Text Classification

We further experimented on fine-grained text classification: CoLA (Wang et al., 2018; Warstadt et al., 2019) and Poem (Sheng & Uthus, 2020). For comparison, we finetuned BERT (Devlin et al., 2019) for language with cross-entropy loss, while employing hinge loss with p-norm distances, SimCSE (Gao et al., 2021), and MixCSE (Zhang et al., 2022b). In Table 5, we present the Matthews’s correlation coefficients (MCC) and F1 scores as used in prior work for each task. The empirical results show that our method consistently improves cross-entropy and is superior to the baselines.

The trained intra-sample relationship shows similarity to the vision domain. In Figure 8, we observe that tokens with special meanings have positive importance scores. They contribute to both the uniqueness and semantics of the sentence. The often-seen tokens, such as prepositions and articles, have negative importance scores. They contribute to distinguishing some sentences but less the semantics. Visualizability of the learned token importance exhibits the partial interpretability provided by atom modeling without post-hoc processing (Ribeiro et al., 2016).

4.5. Ablation Study

We studied the impacts of the soft constraint \( \sum_{i \in A} q_i \) as shown in Figure 9 and Table 6. We observe that only using the term \( E_{A,B} \sim P \sum_{i \in A, j \in B} \frac{q_i q_j}{d_{i,j}} \) in most cases, can achieve good performance but suffers from high variance. When having the soft constraint, the training performance is stabilized.

5. Conclusion

We have presented atom modeling, a new algorithm to achieve model expressiveness by (1) learning token importance for each sub-unit in a data sample, and (2) dynamically distancing data samples based on their structural similarity with no supervision. The learned token importance may be of independent interests as a partial model explanation. Our method is also highly practical, demonstrating effectiveness across diverse deep learning problems.
**Potential Broader Impact**

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

**References**

Alvarez-Melis, D. and Jaakkola, T. S. Towards robust interpretability with self-explaining neural networks. In NeurIPS, 2018.

Bastani, O., Kim, C., and Bastani, H. Interpreting blackbox models via model extraction. arXiv preprint arXiv:1705.08504, 2017.

Bohr, N. I. on the constitution of atoms and molecules. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 26(151):1–25, 1913.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a” siamese” time delay neural network. Advances in neural information processing systems, 6, 1993.

Brown, T. L. Chemistry: the central science. Pearson Education, 2009.

Cayton, L. Algorithms for manifold learning. Univ. of California at San Diego Tech. Rep, 12(1-17):1, 2005.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pp. 1597–1607. PMLR, 2020.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), volume 1, pp. 539–546. IEEE, 2005.

Coulomb, C. A. Premier mémoire sur l’électricité et le magnétisme. Histoire de l’Académie royale des sciences, 569, 1785.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 113–123, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Rand_augment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 702–703, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2019.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. Advances in neural information processing systems, 27, 2014.

Dubey, A., Gupta, O., Raskar, R., and Naik, N. Maximum-entropy fine grained classification. Advances in neural information processing systems, 31, 2018.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12873–12883, 2021.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910, 2021.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864, 2016.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Halliday, D., Resnick, R., and Walker, J. Fundamentals of physics. John Wiley & Sons, 2013.

Hamilton, W. L., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update
rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *The annals of statistics*, 36(3):1171–1220, 2008.

Jain, S. and Wallace, B. C. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk992CeAb.

Keerthi, S. S. and Lin, C.-J. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Lin, T. and Zha, H. Riemannian manifold learning. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):796–809, 2008.

Mohr, P. J., Taylor, B. N., and Newell, D. B. Codata recommended values of the fundamental physical constants: 2006. *Journal of Physical and Chemical Reference Data*, 80(3):633–1284, 2008.

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201, 2001.

Nilsson, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Obukhov, A., Seitzer, M., Wu, P.-W., Zhydenko, S., Kyl, J., and Lin, E. Y.-J. High-fidelity performance metrics for generative models in pytorch. 2020. URL https://github.com/toshas/torch-fidelity. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Ribeiro, M. T., Singh, S., and Guestrin, C. “why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Saul, L. K. and Roweis, S. T. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of machine learning research*, 4(Jun):1929–1958, 2014.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
Tenenbaum, J. B., Silva, V. d., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Wang, X. and Gupta, A. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2015.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040.

Wei, X.-S., Wu, J., and Cui, Q. Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069*, 2019.

Weinberger, K. Q. and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., and Guo, B. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11304–11314, 2022a.

Zhang, Y., Zhang, R., Mensah, S., Liu, X., and Mao, Y. Unsupervised sentence representation via contrastive learning with mixing negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11730–11738, 2022b.
A. Proofs

A.1. Proof of Lemma 1

Lemma 1. A G-Lipschitz function \( o(\cdot) \) and a K-Lipschitz inverse function of \( o(\cdot) \) returns the output space distance such that:

\[
K \| \mathbf{v} - \mathbf{u} \| \leq \| o(\mathbf{v}) - o(\mathbf{u}) \| \leq G \| \mathbf{v} - \mathbf{u} \|, \tag{17}
\]

where \( \mathbf{v} \) and \( \mathbf{u} \) are any vector in the latent space.

Proof. RHS: The definition of Lipschitz continuity with constant \( G \).

LHS: Starting from Lipschitz continuity with constant \( K \) of the inverse function \( o^{-1}(\cdot) \):

\[
\| o^{-1}(o(\mathbf{v})) - o^{-1}(o(\mathbf{u})) \| \leq K \| o(\mathbf{v}) - o(\mathbf{u}) \|. \tag{18}
\]

Using the fact that \( o^{-1}(o(x)) = x \) for any \( x \), we have

\[
\| o(\mathbf{v}) - o(\mathbf{u}) \| \geq K \| \mathbf{v} - \mathbf{u} \|.
\]

A.2. Proof of Theorem 1

Theorem 1. Consider equal token importance distribution, Equation 7 returns the atomic distance such that:

\[
\bar{d}_{AB} \leq C \| \tilde{\mathbf{v}}^A - \tilde{\mathbf{v}}^B \|_2, \tag{19}
\]

where \( \tilde{\mathbf{v}}^A \) is a permutation of \( \ell(\mathbf{z}^A) \) from data sample \( A \).

Proof. The key idea is using the sorted token importance vector \( \delta \) and the sorting permutation matrices \( \pi_A \) and \( \pi_B \), such that:

\[
\delta := \pi_A q_A = \pi_B q_B, \tag{20}
\]

and,

\[
\tilde{\mathbf{v}}^A = \pi_A \mathbf{v}^A, \quad \tilde{\mathbf{v}}^B = \pi_B \mathbf{v}^B. \tag{21}
\]

We can rewrite Equation 7 and use Hölder’s inequality:

\[
\tilde{d} := \| \hat{\mu}_A - \hat{\mu}_B \|_1 = \| \delta \tilde{\mathbf{v}}^A - \delta \tilde{\mathbf{v}}^B \|_1 \leq \| \delta \|_2 \| \tilde{\mathbf{v}}^A - \tilde{\mathbf{v}}^B \|_2. \tag{22}
\]

A.3. Proof of Theorem 2

Theorem 2. Let \( c = \sum_{q_i > 0} q_i q_j \) and \( c^* = \sum_{q_i, q_j < 0} q_i q_j \) for all \( i \in A \) and \( j \in B \), given data samples \( A, B \). Without loss of generality, \( c^* = kc \) and \( k \in (1, \infty) \) gives the optimal distance in Equation 5 as:

\[
d^* = \left( \frac{r_A + r_B}{2} \right) \sqrt{K + 1} \frac{k}{k - 1}. \tag{23}
\]

If \( k \to 1 \), \( d^* \to \infty \).

Proof. We first reparameterize the distance term in Equation 5 by Equation 7:

\[
L_A = \sum_{i \in A, j \in B} \frac{q_i q_j}{d_{ij}},
\]

\[
= \sum_{q_i, q_j > 0} \frac{q_i q_j}{d_{AB}} + \sum_{q_i, q_j < 0} \frac{q_i q_j}{d_{AB} + \frac{r_A + r_B}{2}}. \tag{24}
\]

To derive an equilibrium status of the loss, we first simplify the notation by defining \( d := \bar{d}_{AB} \) and \( \tilde{r} := \frac{r_A + r_B}{2} \).
\[
\frac{\partial \mathcal{L}_A}{\partial d} = \frac{\sum_{q, q_j > 0} q_iq_j(d + \hat{r})^2 + \sum_{q, q_j < 0} -q_iq_j d^2}{d^2(d + \hat{r})^2} = 0
\] (25)

We further set \( c \triangleq \sum_{q, q_j > 0} q_iq_j \) and \( c^* \triangleq -\sum_{q, q_j < 0} q_iq_j \). Therefore,

\[
\begin{align*}
\sum_{q, q_j > 0} -q_iq_j(d + \hat{r})^2 + \sum_{q, q_j < 0} -q_iq_j d^2 \\
= (-c + c^*)d^2 - 2cd\hat{r} - c\hat{r}^2 \\
= (-c + c^*)(d - \frac{c\hat{r}}{c - c^*})^2 - (c + c^*)(\frac{c\hat{r}}{c - c^*})^2 - c\hat{r}^2 = 0
\end{align*}
\] (26)

This equation can be reduced to:

\[
(d - \frac{c\hat{r}}{c - c^*})^2 = \frac{c^2 + c(-c + c^*)}{(-c + c^*)^2} \hat{r}^2 = \frac{c^*c}{(-c + c^*)^2} \hat{r}^2
\] (27)

Therefore, we can find a closed-form solution of \( d \):

\[
d = \left(\frac{c}{-c + c^*} \pm \sqrt{\frac{c^*c}{(-c + c^*)^2}}\right)\hat{r}
\] (28)

To this end, we can observe two unsatisfying cases from the derived formulation of \( d \).

- If \( c = c^* \), the right hand side divided by zero is undefined.
- If \( c > c^* \) and recall that \( d = \| \cdot \|_p \geq 0 \), then \( \frac{c}{c - c^*} \leq \sqrt{\frac{c^*c}{(-c + c^*)^2}} \). This results in \( c^2 < c^*c \), contradict to the premise \( c > c^* \).

The formulation turns out to be satisfied if and only if \( c < c^* \). A balance point exists in \( d^* = \frac{c + \sqrt{c^*c}}{c - c^*} \hat{r} \).

Suppose that \( c^* = kc \) with \( k > 1 \), then,

\[
d^* = \frac{c + \sqrt{c^*c}}{c^* - c} \hat{r} = \frac{c + \sqrt{k}c}{(k - 1)c} \hat{r} = \frac{\sqrt{k} + 1}{k - 1} \frac{r_A + r_B}{2}
\] (29)

This mathematical result indicates that optimizing \( \mathcal{L}_A \) directly depends on the relationship \( k \) of the learned data sample structures.

Moreover, the balanced distance \( d^* \) monotonically decreasing with respect to \( k \). This statement can be proved by the partial derivative is less than zero \( \forall k > 1 \):

\[
\nabla_k \frac{\sqrt{k} + 1}{k - 1} = -\frac{1}{2} - \sqrt{k} - \frac{1}{2} k < 0, \forall k > 1
\] (30)

**B. Discussion**

**Atom modeling layer selection.** We take \( l \) as a hyperparameter to indicate that we apply atom modeling on the \( l \)-th hidden layer output among a total of \( L \) layers. Empirically, there is no clear pattern but the nearest layer to the raw data (\( l=1 \) for classifiers and \( l=L-1 \) for generative models) works.
Token importance dimension selection. Both in theory and empirically, selecting which dimension as token importance impacts neither training from scratch nor fine-tuning. For simplicity, our reported experiments use the first dimension of the representation for $Q(\cdot)$.

Time complexity. The complexity of the multiplication-inverse term $E_{(A,B)\sim D} \sum_{i \in A, j \in B} \frac{q_i q_j d_{ij}}{\alpha_i \alpha_j}$ is $O(N^2 M^2)$, where $M$ is the batch size and $N$ is the number of sub-components per data sample. To reduce the complexity, we sample $M$ pairs of data samples in each batch and select a fixed number $S$ of sub-components from each data sample. The used time complexity is thus $O(MS)$.

When setting $S$ as the maximum of 100 and the data size, the empirical average training time per iteration is listed in Table 7. The results show that atom modeling adds around 3-12% training time to the standard methods and the added time depends on the used $S$ and data size.

|                       | ImageNet-1K | MNIST |
|-----------------------|-------------|-------|
| **ResNet**            | 0.665       | 0.024 |
| +Atom Modeling        | 0.687       | 0.027 |
| **Time Increased**    | 3.3%        | 12.5% |

*Table 7. Training time (seconds) per iteration on ImageNet-1K classification and MNIST image synthesis.*

C. Experimental Details

C.1. Computation

There are four types of computation usages in this work: (1) The synthetic dataset experiments were using 1 RTX2070 with Max-Q with 8G capacity. (2) The experiments of Pets, Flowers, CoLA, Poem, MNIST, CIFAR10 were using 1 Titan RTX with 24G capacity. (3) Each experiment of ImageNet-1K was using 4 Titan RTX for two days. (4) Each experiment of CelebA-HQ was using 6 Titan RTX for one week.

C.2. Used Models, Baselines, and Codes

All implementations in this work are based on PyTorch [https://pytorch.org/](https://pytorch.org/). (1) The synthetic dataset experiments are implemented our own. (2) The experiments of CoLA and Poem are our implementation. (3) The experiments of Pets and Flowers are revised from [https://github.com/kuangliu/pytorch-cifar](https://github.com/kuangliu/pytorch-cifar) and [https://github.com/Skuldur/Oxford-IIIT-Pets-Pytorch](https://github.com/Skuldur/Oxford-IIIT-Pets-Pytorch). (4) The experiments of image generation of MNIST and CIFAR10 use exactly the same script as [https://github.com/pytorch/examples/tree/main/dcgan](https://github.com/pytorch/examples/tree/main/dcgan). (5) The experiments of ImageNet-1K use exactly the same script as [https://github.com/pytorch/examples/tree/main/imagenet](https://github.com/pytorch/examples/tree/main/imagenet). (6) The experiments of CelebA-HQ use exactly the same script as [https://github.com/microsoft/StyleSwin](https://github.com/microsoft/StyleSwin).

Our hyperparameter search is limited to the computation we have for this work and is as follows: (1) The synthetic dataset experiment: We first search the best learning rate using only cross-entropy. Secondly, we use the same learning rate for all auxiliary losses and search their coefficients in $\{0.5,0.2,0.1,0.05,0.02,0.01,0.005\}$. (2) The experiments of Pets, Flowers, CoLA, and Poem: We first search the best learning rate using only cross-entropy. Secondly, we use the same learning rate for all auxiliary losses and search their coefficients in $\{0.05,0.02,0.01,0.005,0.002,0.001\}$. (3) The experiments of ImageNet-1K, MNIST, CIFAR10, and CelebA-HQ only use the coefficient 0.02 for atom modeling without further trials.

We summarize the baselines with their math forms or the implementation details in the following, where $h_i$ denotes the concatenation of all sub-units representations in the $i$-th data sample:

- Hinge (1-norm): $\mathcal{L}_i = \sum_{i=1}^{N} \max \{0, \|h_i, h_i^+\|_1 - \|h_i, h_i^-\|_1\}$

- Hinge (2-norm): $\mathcal{L}_i = \sum_{i=1}^{N} \max \{0, \|h_i, h_i^+\|_2 - \|h_i, h_i^-\|_2\}$

- Rank-H ([Wang & Gupta, 2015]): $\mathcal{L}_i = \sum_{i=1}^{N} \max \{0, \text{sim}(h_i, h_i^+) - \text{sim}(h_i, h_i^-) + M\}$, $M = 0.5$. 

• SimCLR (Chen et al., 2020): $\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(h_i, h_j) / \tau)}{\sum_{k=1}^{P} \exp(\text{sim}(h_i, h_k) / \tau)}$, where $(h_i, h_j)$s are positive pairs ($j$ being one data augmentation result of the $i$-th sample) and $(h_i, h_k)$s are all other possible pairs.

• SimCSE (Gao et al., 2021): $\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(h_i, h_j^+) / \tau)}{\sum_{j=1}^{N} \exp(\text{sim}(h_i, h_j^+) / \tau)}$, where the $h^+$ is the representation produced by the same model using dropout, but at different time.

• MixCSE (Zhang et al., 2022b): improved SimCSE by the modified loss

$$L_{i,j} = -\log \frac{\exp(h_i^T h'_j / \tau)}{C + \sum_{j=1}^{N} \exp(h_i^T h'_j / \tau)},$$

where $C = \exp(h_i^T h'_i / \tau) + \sum_{j=1}^{N} \exp(h_i^T h'_j / \tau)$, $h'_j = \frac{\lambda h'_i + (1-\lambda) h'_j}{\|\lambda h'_i + (1-\lambda) h'_j\|_2}$.

• VQ+DCGAN: the quantizer (Van Den Oord et al., 2017) is implemented as https://github.com/MishaLaskin/vqvae.

• SSCL+DCGAN: revised SimCLR loss since no other positive pairs except for the sample itself exist in image synthesis

$$L_i = -\log \frac{\exp(\text{sim}(h_i, h_i) / \tau)}{\sum_{k=1}^{N} \exp(\text{sim}(h_i, h_k) / \tau)}.$$

C.3. Significance Test Results

The results of significance tests of Table 5 are as the following Table 8. The difference is considered significant when p-value $< 0.05$ and no effect when p-value $> 0.1$.

| Method vs Cross-Entropy | Pets | Flowers | CoLA | Poem |
|-------------------------|------|---------|------|------|
| Hinge (1-norm)          | 0.24 | 0.15    | 0.31 | 0.08 |
| Hinge (2-norm)          | 0.09 | 0.19    | 0.43 | 0.14 |
| Rank-H/SimCSE           | 0.12 | **0.03**| 0.13 | 0.22 |
| SimCLR/MixCSE           | 0.39 | 0.09    | 0.32 | 0.24 |
| Atom Modeling           | **0.03**| **0.02**| **0.01**| **0.09** |

Table 8. The p-value of results comparing each method to only using cross-entropy for training.