CONTRASTIVE LEARNERS ARE SEMANTIC LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we explore the definition of semantic equivalence to establish a connection between contrastive tasks and their downstream counterparts. Specifically, we investigate when a contrastive dataset can learn representations that encode formal semantic equivalence relations for a specific downstream task. In our analysis, we recover a surprising hypothesis resembling the distributional one—dubbed distributional alignment hypothesis. Under this assumption, we demonstrate that the optimal model for simple contrastive learning procedure must generate representations that encode formal semantic equivalence relations for the downstream task. Furthermore, we support the theory with a series of experiments designed to test the presented intuitions.

1 INTRODUCTION

Overview. Contrastive Learning has emerged as a prominent self-supervised training technique, demonstrating success across diverse data modalities. These include, among others, images (Chen et al. (2020); He et al. (2020)), audio (van den Oord et al. (2018); Saeed et al. (2021)), video (Dave et al. (2022); Pan et al. (2021)), text (Fang et al. (2020); Xiong et al. (2020)), graphs (Thakoor et al. (2021); Ling et al. (2023)), and time series (Zheng et al. (2024)). Notable contrastive learning techniques encompass SimCLR (Chen et al. (2020)), MoCo (He et al. (2020)), BYOL (Grill et al. (2020)), SwAV (Caron et al. (2020)), MSimCLR (Korman (2021)), and Barlow Twins (Zbontar et al. (2021)).

Problem Statement. The primary goal of these techniques is to generate high-quality embedding representations that can be effectively utilized for downstream tasks (potentially with limited data availability). Empirical evidence suggests that semantically similar objects tend to be encoded into proximate embedding representations during contrastive training (Chen et al. (2020); Jaiswal et al. (2020)). While the emergence of semantic structure in these embeddings is well-documented (Le-Khac et al. (2020); Kang et al. (2020); Poklukar et al. (2022)), to the best of our knowledge, a formal treatment of this process remains lacking in the current literature.

Proposed Solution. This work aims to address the aforementioned gap by formalizing the concept of *semantic equivalence*, as proposed by Bertolotti & Cazzola (2024) for sequence modeling tasks. Briefly, two symbols are defined as semantically equivalent if they can be substituted without altering the outcome distribution of a certain task (see Definition 2.1). This property is fundamentally rooted in the notion of semantic equivalence in programming languages (Scott & Strachey (1971)), where two code snippets are considered semantically equivalent if their substitution does not affect the program's output regardless of the context (borrowing the denotational semantics for an instant, we would write $\forall \rho : [\![p_1]\!](\rho) = [\![p_2]\!](\rho)$).

Research Questions. To summarize, our research aims to address the following questions:

- **RQ_1: Under what conditions is a contrastive task useful for a downstream task?**
- **RQ_2: How can we train embeddings that effectively encode semantical equivalence relations?**

Findings. Our findings demonstrate that the SimCLR contrastive learning procedure (Chen et al. (2020)) inherently encodes semantically equivalent symbols in close proximity within the embedding space. Furthermore, we find a fundamental requirement for a contrastive task to be useful for the downstream counterpart. We dub this requirement *distributional alignment hypothesis*, for its similarity with the classical *distributional hypothesis* (Bertolotti & Cazzola (2024)). These re-

sults provide a theoretical foundation for the empirically observed semantic structure in contrastive learning embeddings.

2 Background & Notation

Let us begin by introducing some recurring notations. From a classical supervised learning standpoint, one often uses a data distribution from which input-label pairs are sampled, $x, y \sim \mathcal{D}$. In this work, we split the *input* component into *symbol* and *context* components. Therefore, we will use a data distribution from which we can sample symbol-context-label triplets, $\sigma, \rho, y \sim \mathcal{D}$. In practice, a symbol may correspond to a word, while the context may represent the rest of the sentence. Similarly, with images, a symbol may represent an image patch, while the context may represent the rest of the image. We will also use Σ, P, and $\mathcal{Y}$ to denote the symbol, context, and label domains respectively. We will use the notation $p_{\mathcal{D}}$ (or simply p when the underlying data distribution is evident) to denote the probability function for data sampled according to $\mathcal{D}$. For example, $p_{\mathcal{D}}(y|\sigma, \rho)$ represents the probability of label y when conditioned on the symbol context pair (σ, ρ) when σ, ρ and y are sampled according to $\mathcal{D}$.

2.1 Semantic Equivalence

Intuitively, two symbols u and v are semantically equivalent if we can use them interchangeably without affecting the output distribution, or formally:

Definition 2.1 (semantic equivalence (Bertolotti & Cazzola (2024))). Given $u, v \in \Sigma$, $u \mathrel{\overset{\circ}{=}} v$ (u is semantically equivalent to v) iff

$$\forall \rho \in \mathrm{P}, y \in \mathcal{Y} \,.\, p(y|u, \rho) = p(y|v, \rho)$$

For example, consider a simple Masked Language Modeling (MLM) tasks (Devlin et al. (2019)), and, consider the sentence

`the <MASK> of water is half` <u>`empty`</u>`/`<u>`full`</u>.

Here, we could use both `empty` and `full` without affecting the outcome distribution of the mask token. Therefore, we can say that, in this context, `empty` and `full` are semantically equivalent. If this relation were to hold true for all contexts, then, we would say that `empty` and `full` are semantically equivalent in general.

In a different scenario, consider a cat-vs-dog image classifier where each symbol represents an image patch—similar to the setup used in Vision Transformers (Dosovitskiy et al. (2020)). In this context, two image patches, such as one depicting a clouded sky and another showing a clear sky, would be considered semantically equivalent since they do not influence the classification outcome.

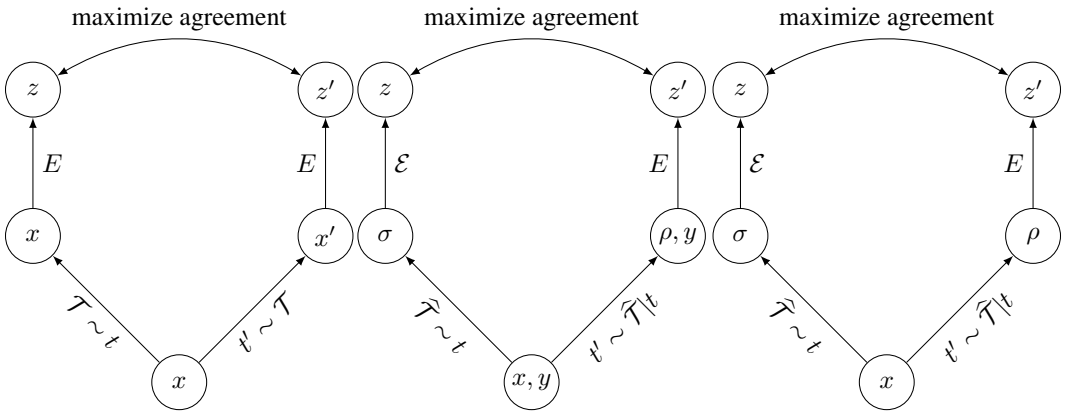
A similar notion of semantics is also used by Chiang & Yogatama (2023).

Throughout this work, we will often use the term semantic with its formal meaning in mind rather than it liguistic meaning. Therefore, unless stated otherwise, when speaking of semantic relations or semantic learning, we refer to the formal definition of semantic equivalence (Definition 2.1).

2.2 Simple Contrastive Learning

SimCLR (Chen et al. (2020)) is a straightforward contrastive learning procedure initially proposed for image data (see Figure 1a). The process begins with a batch of N data points $\{x_i\}_{i=1}^N$ sampled from the data distribution $x_i \sim \mathcal{X}$, and a corresponding batch of N augmentation pairs $\{(t_i, t'_i)\}_{i=1}^N$ sampled from a transformation distribution $t_i, t'_i \sim \mathcal{T}$. The objective is to maximize agreement between different augmentations of the same data point ($t_i(x_i)$ and $t'_i(x_i)$) while maximizing disagreement between different data points ($t_i(x_i)$ and $t'_j(x_j)$ for $i \neq j$). This is achieved through the following loss function (InfoNCE (van den Oord et al. (2018))):

$$\mathcal{L}(D) = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{\exp(\mathrm{sim}(E(t_i(x_i)), E(t'_i(x_i))))}{\sum_j \exp(\mathrm{sim}(E(t_i(x_i)), E(t'_j(x_j))))} \right) \tag{1}$$



(a) SimCLR (b) SimCLR variant with labels. (c) SimCLR variant without labels.

Figure 1: Figure 1a illustrates the SimCLR architecture. Figure 1b shows an asymmetric SimCLR architecture, where two different encoder functions, $\mathcal{E}$ and E, are used in separate branches. Additionally, a different set of augmentations, $\widehat{\mathcal{T}}$, is applied, with the right branch's augmentation conditioned on the left branch. Figure 1c depicts the previous variants without the use of labels

Here, E denotes an encoder network and sim represents a similarity function, typically scalar product or cosine similarity (we assume the former in this work). The i-th component in the summation is minimized when the similarity between $t_i(x_i)$ and $t'_i(x_i)$ is maximized, while the similarity between $t_i(x_i)$ and any other $t'_j(x_j)$ is minimized. The full loss $\mathcal{L}$ is obtained by averaging over the batch (D).

3 Contrastive Learners (with labels) are Semantic Learners

Contrastive learning techniques typically assume that labels for downstream tasks are unavailable, as they are often costly to obtain. However, as we will show, if labels are provided to a simple variation of SimCLR, it is possible (under appropriate assumptions) to recover representations that encode semantic equivalence relations (as in Definition 2.1). This means that if two symbols are semantically equivalent, their representations will also be equivalent. In the next section, we will explore how to eliminate the need for labels in this framework.

The variant of SimCLR, depicted in Figure 1b, introduces two main differences. First, we employ two specific types of augmentation functions: $t_i : \mathcal{X} \to \Sigma$ and $t'_i : \mathcal{X} \to \mathrm{P}$. The former is responsible for extracting the i-th symbol from the input, while the latter returns the context of the i-th symbol. Here, the i-th symbol may represent the i-th image patch, and the i-th context may represent the remaining patches. Using SimCLR terminology, the first augmentation could be a *crop and resize*, resulting in the *symbol,* while the second could be a *Cutout* specifically removing the previous *symbol* to produce the *context.* Let us refer to the set of these augmentations with $\widehat{\mathcal{T}}$.

Secondly, we introduce an asymmetric architecture to SimCLR. Specifically, we denote the embedding function $\mathcal{E} : \Sigma \to \mathbb{R}^d$, which maps symbols to a d-dimensional representation. Meanwhile, let $E : \mathrm{P} \times \mathcal{Y} \to \mathbb{R}^d$ represent the function that maps context-label pairs to a d-dimensional representation. The parameters between the embedding function and the encoder function can be shared, partially shared, or separate.

Consider a batch $D = \{(\sigma_i, \rho_i, y_i)\}_{i=1\ldots N}$ consisting of symbol-context-label triplets, where $\sigma_i = t(x)$ and $\rho_i = t'(x)$ for $x, y \sim \mathcal{D}$, $t \sim \widehat{\mathcal{T}}$, and $t' \sim \widehat{\mathcal{T}}|t$. Here, $\mathcal{D}$ represents the input-label data distribution, $\widehat{\mathcal{T}}$ denotes the augmentation distribution for extracting the *symbol* from x, and $\widehat{\mathcal{T}}|t$ represents the augmentation distribution for extracting the context of σ. Our goal is to minimize this variation of the InfoNCE loss between the encoder E and the embedding function $\mathcal{E}$:

$$\mathcal{L}_{NCE}(D) = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(\rho_i, y_i)))}{\sum_j \exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(\rho_j, y_j)))}\right) \tag{2}$$

The effect of this loss is to encourage the embedding function $\mathcal{E}$ to encode symbol σ_i close to their corresponding context-label pairs ρ_i, y_i and far apart from the noisy pairs ρ_j, y_j, for $j \neq i$. This fact can be seen by simply deriving the update rule for the embedding function $\mathcal{E}$ and the encoder function E. This is provided in Appendix A.5.

Further, it can be shown that, an optimal model, made of $\mathcal{E}^*$ and E^*, for $\mathcal{L}_{NCE}$ needs to encode semantically equivalent symbols with exactly the same vector. This is the subject of the following theorem:

Theorem 3.1. *Given the symbols $u, v \in \Sigma$ such that 1. $u \mathrel{\hat{=}} v$, 2. $\forall \rho : p(\rho|u) = p(\rho|v)$, and 3. if there are $(\rho_1, y_1), \ldots, (\rho_d, y_d)$ context-label pairs such that $E^*(\rho_i, y_i)$ form a basis for $\mathbb{R}^d$ then $\mathcal{E}^*(u) = \mathcal{E}^*(v)$*

On one hand, semantic equivalence relations do not depend on the conditional distribution ($p(\rho|u) = p(\rho|v)$); therefore, these hypotheses may seem unnecessary. On the other hand, the data must still reflect these semantic equivalence relations to some extent. For instance, consider an edge case where two symbols are semantically equivalent, but one of them never appears in the dataset. In this scenario, it is natural that we would be unable to discover this semantic equivalence relation.

The final hypothesis, concerning the basis, ensures that the only solution is $\mathcal{E}^*(u) = \mathcal{E}^*(v)$. Given that most realistic scenarios encompass thousands of possible context-label pairs while the embedding size typically does not exceed $d = 1024$. Further, it should be noted that establishing a basis for $\mathbb{R}^d$ only requires d linearly independent embeddings. The determinant forms a polynomial of the entries $E^*(\rho_1, y_1), \ldots, E^*(\rho_d, y_d)$ and has a Lebesgue measure of zero. Thus, slight perturbation of the encoded vectors would establish a basis almost surely. This hypothesis is employed in the context of invertible neural networks Finzi et al. (2019), where assuming a weight matrix to be a basis implies non-singularity, allowing for inversion. Additionally, one can always reduce the embedding size to ensure that this hypothesis holds. Ultimately, this hypothesis can be considered realistic for most use cases.

Please refer to Appendix A.2 for a formal proof of Theorem 3.1.

4 Contrastive Learners are Semantic Learners

As we have seen, the notion of semantic equivalence is strongly tied to a label distribution. However, the rise of contrastive learning is partially due to the fact that it does not require labels (which are usually expensive to obtain) to generate embeddings organized in a semantic structure (Gao et al. (2021); Le-Khac et al. (2020); He et al. (2020)). This raises a natural question: how do contrastive learning techniques learn formal semantics without access to labels?

To answer this question, we need to formalize the concepts of downstream and contrastive tasks. Let us denote with $\mathcal{D}$ the data distribution for the downstream task. We also assume to be able to sample symbol-context-label triplets, $\sigma, \rho, y \sim \mathcal{D}$. Further, we will denote with $p_{\mathcal{D}}(y|\sigma, \rho)$ the probability of label y given the symbol-context pair σ and ρ.

Now that the downstream task has been defined, we proceed to the contrastive task. Here, we are given a data distribution $\mathcal{C}$. From this distribution, we can only sample context-symbol pairs, $\sigma, \rho \sim \mathcal{C}$ but we have no access to a label.

Of course, we need to realize that we cannot apply contrastive learning to any data distribution and hope to learn useful representations for a downstream task of interest. For example, we cannot hope to learn useful representations for an animal classifier from a white noise distribution. Therefore, we will need some kind of hypothesis between contrastive and downstream data to hold. We refer to this hypothesis as the distributional alignment hypothesis:

Definition 4.1 (Distributional Alignment Hypothesis)**.** A downstream distribution $\mathcal{D}$ and a contrastive distribution $\mathcal{C}$ are aligned iff.

$$\forall y \in \mathcal{Y}, \rho \in \mathrm{P} : \; p_{\mathcal{D}}(y|u, \rho) = p_{\mathcal{D}}(y|v, \rho) \iff \forall \rho \in \mathrm{P} : p_{\mathcal{C}}(\rho|u) = p_{\mathcal{C}}(\rho|v)$$

On the left, we state that u and v are semantically equivalent according to the downstream task. On the right, we state that u and v are conditionally equivalent for the contrastive task. Notice that, this definition highly resembles the definition of distributional hypothesis as formalized by Bertolotti & Cazzola (2024). The main and only difference between the distributional hypothesis and the distributional alignment hypothesis is simply that the first is concerned with a single data distribution while the latter is concerned with two distributions (one for the downstream data and one for the contrastive data).

Now, suppose we aim to minimize the loss in Equation 2 without access to the labels. Then we minimize the following:

$$\mathcal{L}_{NCE}(D) = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(\rho_i)))}{\sum_j \exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(\rho_j)))}\right)$$

One can now show that the following theorem holds:

Theorem 4.2. *Given the symbols $u, v \in \Sigma$ such that: 1. $\forall \rho : p(\rho|u) = p(\rho|v)$. 2. If there are $\rho_1, \ldots, \rho_d$ contexts such that $E^*(\rho_i)$ form a basis for $\mathbb{R}^d$ then $\mathcal{E}^*(u) = \mathcal{E}^*(v)$*

This theorem shows that an optimal embedding function needs to encode such symbols u and v with the same vector. The proof of this theorem is provided in Appendix A.3.

Similar argument discussed for Theorem 3.1 can be applied here. The hypothesis concerning the basis is necessary to ensure that the only solution is $\mathcal{E}^*(u) = \mathcal{E}^*(v)$, and it can be considered realistic. The $p(\rho|u) = p(\rho|v)$ hypothesis, despite being more strong, is necessary to ensure that the data reflects the semantic equivalence relations.

By combining Definition 4.1 and Theorem 4.2, it is easy to derive the following corollary:

Corollary 4.3. *1. Let $\mathcal{C}$ and $\mathcal{D}$ be a distributionally aligned contrastive and downstream distributions, respectively. 2. Let $\mathcal{E}^*$ and E^* be the optimal embedding and encoder functions for the contrastive task. 3. Let $\rho_1, \ldots, \rho_d$ contexts such that $E^*(\rho_i)$ form a basis for $\mathbb{R}^d$. Then,*

$$u \mathrel{\overset{\circ}{=}_{\mathcal{D}}} v \iff \mathcal{E}^*(u) = \mathcal{E}^*(v)$$

Here, $u \mathrel{\overset{\circ}{=}_{\mathcal{D}}} v$ denotes the semantic equivalence relation according to the downstream task $\mathcal{D}$. The proof of this corollary is immediate. Firstly, from the distributional alignment hypothesis (Definition 4.1), we have that, two symbols are semantically equivalent for the downstream task if and only if they are conditionally equivalent for the contrastive task. Further, two conditionally equivalent symbols are encoded with the same vector by the optimal embedding and encoder functions for the contrastive task (Theorem 4.2). Therefore, two symbols are encoded within the same vector only if they were semantically equivalent to begin with. The backward implication is derived in Proposition A.2 of the Appendix Section A.4.

This result highlight a fundamental connection between contrastive learning and semantic learning (we refer to semantic learning as the process to learn formal semantic relations). In particular, it shows that contrastive learning can be seen as a form of semantic learning, where the semantic equivalence relations (precisely those state in Definition 2.1) are learned from the data distribution itself.

a	b	y
45	5	0
12	21	3
66	12	8

Table 1: Examples of downstream data for the ModAdd task. $a, b, c \in \{0, \ldots, 100\}$ and $k = 10$.

a	b	c	y
45	?	0	5
12	?	3	21
?	12	8	66

Table 2: Examples of contrastive data from the CModAdd task. $a, b \in \{0, \ldots, 100, ?\}$ and $k = 10$.

5 EXPERIMENTS

In this section, we will focus on designing a small controlled experiment where the hypothesis of the previous theorem can be easily verified.

Modular Addition Task. We will consider an algorithmic task commonly referred to as modular addition, or ModAdd for short (Power et al. (2022); Gromov (2023); Furuta et al. (2024)). In ModAdd, a model is tasked with solving simple modular addition equations of the form:

$$(a + b) \mod k = y$$

where y represents the unknown quantity, and $a, b, k \in \{0, \ldots, N\}$ with $N \in \mathbb{N}$. Typically, k is fixed at the start of training, and the neural network (NN) is presented with pairs of natural numbers and tasked with predicting the correct modulo class. A few training examples are provided in Table 1.

Semantic Equivalence in ModAdd. It should be apparent that all symbols of the form $a + ik$ are semantically equivalent to a in ModAdd. For example, consider the problem instantiated with $k = 10$ and $N = 100$. In this case, symbols like $0, 10, 20, \ldots, 100$ are semantically equivalent. In other words, if we swap 10 with 80, we do not alter the outcome distribution of the problem (i.e., $\forall y, a : p(y \mid a + 80 \bmod 10) = p(y \mid a + 10 \bmod 10)$).

Contrastive Modular Addition Task. The contrastive modular addition task, or CModAdd for short, is a simple extension of the previous ModAdd task. In CModAdd, we generate sequences of three symbols. The first symbol represents the first addend, a, which is sampled according to $a \sim \mathcal{U}(0, N)$. The second symbol represents the second addend, b, sampled according to $b \sim \mathcal{U}(0, N)$. The third symbol represents the result of the modulus operation, $c = (a + b) \bmod k$. Additionally, we randomly mask one of the symbols a or b, and refer to the masked symbol as y. The goal of this task is to train a representation for the symbols that can be useful for the downstream task ModAdd. Table 2 provides a few examples for clarity.

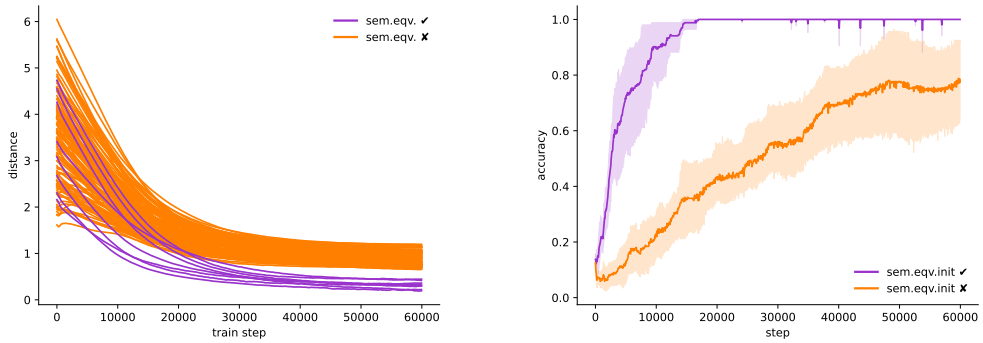
Conditional Equivalence in CModAdd. It should be apparent that all symbols of the form $a+ik$ are conditionally equivalent to a in CModAdd. For example, consider the problem instantiated with $k = 10$ and $N = 100$. In this case, symbols like $0, 10, 20, \ldots, 100$ are conditionally equivalent. In other words, the probability of seeing 10 under the mask token is always the same as the probability of seeing 80 under the mask token (i.e., $\forall y, a : p(a{+}? \bmod 10 = y \mid 10) = p(a{+}? \bmod 10 = y \mid 80)$).

CModAdd and ModAdd are aligned. Let u and v be semantically equivalent symbols according to ModAdd (i.e., $\forall y, a : p(y \mid a + u \bmod k) = p(y \mid a + v \bmod k)$). Then, it is easy to see that u and v are conditionally equivalent for CModAdd ($\forall y, a : p(a{+}? \bmod k = y \mid u) = p(a{+}? \bmod k = y \mid v)$) and vice versa. This property aligns exactly with Definition 4.1. Therefore, we can say that CModAdd and ModAdd are distributionally aligned. Furthermore, by applying Theorem 4.2, we find that employing a simple contrastive procedure will result in conditionally equivalent symbols for CModAdd being encoded into spatially close vectors. Since conditionally equivalent symbols in CModAdd are semantically equivalent in ModAdd, we can conclude that semantically equivalent symbols in ModAdd will be encoded into spatially close embeddings as stated from Corollary 4.3.

ModAdd & CModAdd Datasets. We generated all possible samples for a ModAdd and CModAdd problem with $N = 16$ and $k = 8$. Of this dataset, only the 80% is used for training purposes, the other 20% is used as a validation set to perform periodic evaluation (one each 10 epochs).

CModAdd Architecture. The CModAdd architecture consists of two components: 1. an embedding function, and 2. an encoder function. As previously mentioned, the embedding function maps symbols to their d-dimensional representations. The encoder function, on the other hand, maps contexts to their d-dimensional representations. The embedding function is simply a parameter matrix of size $N \times d$, mapping each symbol index to a randomly initialized d-dimensional vector. The encoder function consists of: 1. an embedding layer, 2. a 3-layer transformer encoder, and 3. a linear layer that maps the latent space to the d-dimensional representation. To comply with the third hypothesis of Theorem 4.2, d is chosen to be small, specifically $d = 8$.

ModAdd Architecture. The ModAdd architecture consists of: 1. an embedding layer, 2. a 3-layer transformer encoder, and 3. a linear layer that maps the latent space to the label. Again, the embed-



(a) Symbol Embedding distances between semantically equivalent (purple) pairs and semantically different pairs (orange) measured at different training steps

(b) Validation accuracy (mean and 95% confidence interval) of 5 models measured at different training steps. In purple, a model that is initialized with SimCLR embeddings. In orange, the same model but with random embedding initialization

Figure 2: Embedding distances for the CModAdd task and validation accuracy for the ModAdd task with and without pre-trained embeddings

ding size is chosen to be small, $d = 8$. This allows for initializing these embeddings with those trained from the CModAdd architecture.

Training Hyperparameters. The training proceed for $1e4$ epochs with Adam optimizer (Kingma & Ba (2015)). The learning rate is set to $1e{-}4$ and we employ weight decay with value $1e{-}2$. The batch size is set to 32.

Contrastive Results. The results are shown in Figure 2a. In this figure, we plot how the Euclidean distance between different symbol embeddings evolves during training. The orange line represents symbol pairs that are not semantically equivalent, while the purple line represents symbol pairs that are semantically equivalent. As observed, during training, only semantically equivalent symbols become close to each other, while the distance between semantically different pairs does not converge beyond a certain level. We note that the distance does not become zero (as the theory would suggest) this is likely due to the loss becoming close to zero preventing noticeable movement between the embeddings.

Classification Results. The results are shown in Figure 2b. To simulate a common procedure involving contrastive pre-training followed by classification training, we first train an embedding function by minimizing the loss shown in Equation 2 On the CModAdd data. Next, we train a classification model for the ModAdd task. The orange line represents the results for a randomly initialized model, while the purple line represents the results for a model initialized with the pre-trained embeddings. The model initialized with pre-trained embeddings achieves better results earlier in training compared to the randomly initialized model.

In Section A.6, we present additional experiments that vary, model size and training procedure.

6 Discussion

Let us recall that the main goal of contrastive learning is to learn *good* representations of data for downstream tasks, which often have limited availability. But what constitutes a good representation? One might be tempted to define it as a representation that facilitates fast and efficient learning of the downstream task. However, this definition doesn't specify any particular properties that the representation should possess. As a result, questions like **When is a contrastive dataset good for a specific downstream task?** remain unanswered.

We argue that good representations should, at a minimum, encode semantic equivalence relations (Definition 2.1). In other words, if two symbols u and v are semantically equivalent (as defined in

Definition 2.1), their representations should be close to each other, meaning they should be encoded as vectors in close proximity. Conversely, non-equivalent symbols should have representations that are further apart. This belief is strongly supported by Figure 2b, where a model initialized with semantic equivalence relations achieves perfect accuracy early in training compared to the randomly initialized one.

If we accept that good representations need to encode semantic equivalence relations, the next question is: **how do we train embeddings that encode these equivalence relations?** To address this, we turn to a popular pre-training method—contrastive learning. Empirically, contrastive learning has often been observed to generate semantically meaningful relations (Wu et al. (2018); Ge et al. (2022)), although semantics is rarely discussed formally. If we accept the definition of semantic as the one presented in Definitions 2.1, we notice that it heavily depends on the presence of labels. Now, if we imagine feeding labels into a contrastive procedure, one could show that a model minimizing Equation 2 would need to encode semantic equivalence relations. This is the subject of Theorem 3.1. However, contrastive learners usually do not have access to labels. In such cases, it can be shown that when the contrastive data are aligned with the downstream data (Definition 4.1), the trained representation must encode these equivalence relations for the downstream task. This is the subject of Corollary 4.3. This behavior is also empirically demonstrated in Figure 2a, where, during contrastive training, embeddings of semantically equivalent symbols converge, while embeddings of semantically different symbols do not.

To summarize the results of this work let us answer briefly the proposed research question:

Under what conditions is a contrastive task useful for a downstream task?

A contrastive dataset can be used for a downstream task when the distributional alignment hypothesis holds.

How can we train embeddings that effectively encode semantical equivalence relations?

The optimal SimCLR variant (under the proper hypotheses), discussed in Section 3, is formally guaranteed to encode semantic equivalence relations. Empirically, the research community has observed this behavior in a variety of different settings.

7 Threats to Validity

In this section, we aim to address the limitations and potential threats that could impact our conclusions.

External Validity We claimed that the distributional alignment hypothesis is crucial to the success of contrastive learning. We supported this claim with both empirical and theoretical evidence. However, it should be noted that, in most realistic scenarios, verifying this hypothesis is impractical. Consequently, in real-world applications, the achieved performance on downstream tasks remains the only empirical measure of the quality of the representation.

Internal Validity While Theorems 3.1, and 4.2 formalize the intuitions of this work, they also rely on somewhat strong hypotheses that limit their applicability to most real-world scenarios.

One common assumption in this manuscript is the conditional equivalence $\forall \rho : p(\rho|u) = p(\rho|v)$. This assumption is often too strong for practical applications. However, it can be justified by the fact that if the symbols u and v never appear in the same context, discovering their semantic equivalence would be impossible. Nevertheless, it is important to note that contrastive learners are often capable of encoding semantic similarities even when this assumption is violated to some extent. This suggests that the results of this work could be generalized by relaxing the conditional equivalence assumption.

The alignment hypothesis (Definition 4.1) used in the proof of Corollary 4.3 is quite strong. In practice, the alignment between representations of semantically equivalent symbols is often only partial, yet contrastive learning has shown success even in such cases. Nevertheless, the conditions under which a contrastive dataset can effectively support a downstream task remain very stringent. For example, randomly generated data is unlikely to be useful for any downstream task, and genomic

contrastive data would provide little value to a natural language sentiment classifier. Ultimately, this suggests that strong hypotheses are necessary even in practical applications.

8 Related Works

Contrastive Learning in practice. Contrastive learning is one of the most popular and studied pre-training techniques. While we present a fairly general framework, other works focus on specific data modalities. For example, contrastive learning techniques for text sentences are discussed by Gao et al. (2021) and Aberdam et al. (2021), see Xu et al. (2023) for a review. On source code contrastive learning has been applied by Bertolotti & Cazzola (2023); Wang et al. (2022b) and Ding et al. (2023). You et al. (2020); Zhu et al. (2020), and Qiu et al. (2020) focus on graph representations, see Liu et al. (2022) for a review. In particular, the work of Xia et al. (2022) focuses on a framework that does not require augmentation. Meanwhile, the work of Wang et al. (2020); Park et al. (2020) and Wang et al. (2023) focuses on images. Tsai et al. (2021a) combine a mixture of expert techniques with contrastive learning image representations. Several long-term forecasting problems are addressed by Park et al. (2024). Other works focus on multiple data modalities (Daunhawer et al. (2023) and Yuan et al. (2021)), see Zong et al. (2024) for a recent survey. Lo et al. (2024) applied contrastive learning to the agent communication problem. Refer to Rani et al. (2023) and Gui et al. (2024) for general survey regarding self-supervised learning and contrastive learning.

Contrastive Learning Techniques. While we mainly focus on the InfoNCE loss with SimCLR. However, there are several other contrastive objectives and technique that have been proven useful in practice that we do not address in the previous sections. Cho (2005) propose one of the first contrastive objectives introduced in the literature that aims to maximize similarity from pairs of the same class and minimize otherwise. A similar objective in principle is the triple loss (Schroff et al. (2015)). An extension to the triple loss, known as Lifted Structured Loss is proposed by Oh Song et al. (2016). NCE and InfoNCE losses developed by Gutmann & Hyvärinen (2010) and van den Oord et al. (2018) exploit the concept of matching a data point with the correct one among a set of noise data points. More recently, Shidani et al. (2024) proposes a multi-view objective. Lu et al. (2022) tackles the out-of-distribution problem proposing invariant Causal Representation Learning—iCaRL. Zero-CL (Zhang et al. (2022)) is a contrastive learning technique that does not rely on negative pairs. Wang et al. (2022a) develops PiCO, a contrastive learning framework for partial label learning. Zhang et al. (2023) proposes a contrastive loss based distributionally robust optimization. The work of Fort et al. (2021); Ge et al. (2021); Wen & Li (2021) focuses on studying and designing the effect multiple and different augmentations.

Understanding Contrastive Learning. In their works, Arora et al. (2019); Lee et al. (2021) present some of the first theoretical analyses of contrastive learning, establishing a connection between the contrastive task and the downstream task. Similarly, HaoChen et al. (2021) and Wang et al. (2022c) relax some of the strong assumptions made in earlier analyses. Their work is based on the existence of a latent variable from which positive and negative contrastive data pairs can be sampled. The work of Saunshi et al. (2022) challenge prior theoretical result in scenarios with different inductive biases. Wang & Isola (2020) prove that cross-entropy contrastive loss optimizes closeness and uniformity. An information-theoretical perspective is provided by Tian et al. (2020); Tsai et al. (2021b); Tosh et al. (2021), focusing on learning sufficient statistics for the downstream task. Johnson et al. (2022) unveils a connection between contrastive learning and kernel PCA (Schölkopf et al. (1997)). Alon et al. (2023) focus on providing a bound on the sample complexity for several contrasting learning scenarios. Simon et al. (2023) study the stepwise learning phenomenon observed in contrastive learners through the lenses of linearized model. The work of Shen et al. (2022) and Garg et al. (2024) compares these techniques when distributional shifts occur. Cui et al. (2023) focuses on contrastive learning when joined with weak supervision. The works of Tschannen et al. (2019); Poole et al. (2019); Lee et al. (2024); Gálvez et al. (2023) study the relation between contrastive learning and mutual information. In this line of work, Levy & Goldberg (2014) shows that skip-grams with negative sampling is equivalent to perform matrix factorization. Wu et al. (2023) offers a perspective from distributionally robust optimization theory. Compared to these works, our analysis stems from the notion of semantic equivalence and explores under what conditions this relationship can be learned using contrastive data.

9 CONCLUSION

In this work, we analyzed what constitutes a good contrastive task from the theoretical perspective of semantics. In Sections 3 and 4 we formalized the conditions under which contrastive learning can effectively encode semantics equivalence relations in both scenarios—with and without labels available. In the latter scenario, we found that the distributional alignment hypothesis (Definition 4.1) plays a fundamental role. Furthermore, we empirically verified the intuitions formalized earlier in Section 5 through small, controlled experiments where all hypotheses could be easily tested. To further validate the presented intuitions, we conducted additional experiments discussed in Appendix A.6. The code to reproduce the experiments is freely available at the following URL:

https://redacted-for-anonimity.com/

10 FUTURE WORK

In this work, we have shown that an optimal contrastive model must learn to encode semantic equivalence relations under the distributional alignment hypothesis (Definition 4.1) and ideal assumptions. However, in practical scenarios, the alignment between contrastive data and downstream tasks is often partial, and the conditional equivalence assumption is frequently violated. A potential direction for future research is to explore the robustness of the proposed theory when these assumptions are only partially met or entirely invalid.

Multitask learning is a paradigm focused on training models to perform multiple tasks simultaneously. In the work of Maurer et al. (2016), the learning model is structured with a shared representation function across tasks and a task-specific classification (or regression) head for each task. We hypothesize that the shared representation function should encode semantic equivalence relations across all tasks concurrently. However, an intriguing question arises: what happens when the semantic structure is not well-aligned across tasks? Addressing this issue presents a compelling avenue for future work.

ACKNOWLEDGMENTS

Redacted for Anonimity

REFERENCES

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.

Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15302–15312, 2021.

Noga Alon, Dmitrii Avdiukhin, Dor Elboim, Orr Fischer, and Grigory Yaroslavtsev. Optimal sample complexity of contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *36th International Conference on Machine Learning, ICML 2019*, pp. 9904–9923. International Machine Learning Society (IMLS), 2019.

Francesco Bertolotti and Walter Cazzola. Fold2Vec: Towards a Statement Based Representation of Code for Code Comprehension. *Transaction on Software Engineering and Methodology*, 32(1): 6:1–6:31, February 2023. doi: 10.1145/3514232.

Francesco Bertolotti and Walter Cazzola. By Tying Embeddings You Are Assuming the Distributional Hypothesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, Wien, Austria, 21-27 of July 2024.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Ting Chen, Simon Kornblith, and Mohammed Norouzi. A Simple Framework for Contrastive Learning of Visual Representations. In Emtiyaz Daumé III and Po-ling Loh (eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, pp. 1597–1607, Vienna, Austria, July 2020. PMLR.

Ting-Rui Chiang and Dani Yogatama. The distributional hypothesis does not fully explain the benefits of masked language model pretraining. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=NMMnxhQm01.

Jingyi Cui, Weiran Huang, Yifei Wang, and Yisen Wang. Rethinking weak supervision in helping contrastive learning. In *International Conference on Machine Learning*, pp. 6448–6467. PMLR, 2023.

Imant Daunhawer, Alice Bizeul, Emanuele Palumbo, Alexander Marx, and Julia E Vogt. Identifiability results for multimodal contrastive learning. In *International Conference on Learning Representations*, 2023.

Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19)*, pp. 4171–4186, Minneapolis, MN, USA, June 2019. Association for Computational Linguistics.

Yangruibo Ding, Saikat Chakraborty, Luca Buratti, Saurabh Pujar, Alessandro Morari, Gail E. Kaiser, and Baishakhi Ray. Concord: Clone-aware contrastive learning for source code. *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2023. URL https://api.semanticscholar.org/CorpusID:259089107.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.

Marc Finzi, Pavel Izmailov, Wesley Maddox, Polina Kirichenko, and Andrew Gordon Wilson. Invertible Convolutional Networks. In *Proceedings of the 1st Workshop on Invertible Neural Nets and Normalizing Flows (INNNF'19)*, Long Beach, CA, USA, June 2019. PMLR.

Stanislav Fort, Andrew Brock, Razvan Pascanu, Soham De, and Samuel L Smith. Drawing multiple augmentation samples per image during training efficiently decreases test error. *arXiv preprint arXiv:2105.13343*, 2021.

Hiroki Furuta, Gouki Minegishi, Yusuke Iwasawa, and Yutaka Matsuo. Interpreting grokked transformers in complex modular arithmetic. *arXiv e-prints*, pp. arXiv–2402, 2024.

Borja Rodrıguez Gálvez, Arno Blaas, Pau Rodríguez, Adam Golinski, Xavier Suau, Jason Ramapuram, Dan Busbridge, and Luca Zappella. The role of entropy and reconstruction in multi-view self-supervised learning. In *International Conference on Machine Learning*, pp. 29143–29160. PMLR, 2023.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP'21)*, pp. 6894–6910, Punta Cana, Dominican Republic, November 2021. ACL.

Saurabh Garg, Amrith Setlur, Zachary Lipton, Sivaraman Balakrishnan, Virginia Smith, and Aditi Raghunathan. Complementary benefits of contrastive learning and self-training under distribution shift. *Advances in Neural Information Processing Systems*, 36, 2024.

Songwei Ge, Shlok Kumar Mishra, Haohan Wang, Chun-Liang Li, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *ArXiv*, abs/2110.14189, 2021. URL https://api.semanticscholar.org/CorpusID:239998348.

Songwei Ge, Shlok Kumar Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6840–6849, 2022. URL https://api.semanticscholar.org/CorpusID:254125715.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Andrey Gromov. Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*, 2023.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/gutmann10a.html.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Daniel D Johnson, Ayoub El Hanchi, and Chris J Maddison. Contrastive learning can find an optimal basis for approximately view-invariant functions. In *The Eleventh International Conference on Learning Representations*, 2022.

Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International conference on learning representations*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*, San Diego, CA, USA, May 2015.

Eric O Korman. Self-supervised representation learning on manifolds. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.

Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021.

Kyungeun Lee, Jaeill Kim, Suhyun Kang, and Wonjong Rhee. Towards a rigorous analysis of mutual information in contrastive learning. *Neural Networks*, 179:106584, 2024.

Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In Z. Ghahramani, M. Welling, N. Cortes, C.and Lawrence, and K. Q. Weinberger (eds.), *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*, pp. 2177–2185, Montréal, Canada, December 2014. Curran Associates Inc.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Hongyi Ling, Zhimeng Jiang, Youzhi Luo, Shuiwang Ji, and Na Zou. Learning fair graph representations via automated data augmentations. In *International Conference on Learning Representations (ICLR)*, 2023.

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6): 5879–5900, 2022.

Yat Long Lo, Biswa Sengupta, Jakob Nicolaus Foerster, and Michael Noukhovitch. Learning multi-agent communication with contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vZZ4hhniJU.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-e4EXDWXnSn.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016. URL http://jmlr.org/papers/v17/15-242.html.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and Their Compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Killian Q. Weinberger (eds.), *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013. Curran Associates Inc.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv e-prints*, arXiv:1807.03748:1–13, July 2018.

Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11205–11214, 2021.

Junwoo Park, Daehoon Gwak, Jaegul Choo, and Edward Choi. Self-supervised contrastive learning for long-term forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nBCuRzjqK7.

Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pp. 319–345. Springer, 2020.

Petra Poklukar, Vladislav Polianskii, Anastasiia Varava, Florian T. Pokorny, and Danica Kragic Jensfelt. Delaunay component analysis for evaluation of data representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HTVch9AMPa.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets. *arXiv e-prints*, arXiv:2201.02177: 1–10, January 2022.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1150–1160, 2020.

Veenu Rani, Syed Tufael Nabi, Munish Kumar, Ajay Mittal, and Krishan Kumar. Self-supervised learning: A succinct review. *Archives of Computational Methods in Engineering*, 30(4):2761–2775, 2023.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3875–3879. IEEE, 2021.

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pp. 19250–19286. PMLR, 2022.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pp. 583–588. Springer, 1997.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In David Forsyth, Ivan Laptev, Deva Ramanan, and Aude Oliva (eds.), *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 815–823, Salt Lake City, UT, USA, June 2015. IEEE.

Dana Scott and Christopher Strachey. *Toward a Mathematical Semantics for Computer Languages*. Oxford University, Oxford, United Kingdom, August 1971.

Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International conference on machine learning*, pp. 19847–19878. PMLR, 2022.

Amitis Shidani, R Devon Hjelm, Jason Ramapuram, Russell Webb, Eeshan Gunesh Dhekane, and Dan Busbridge. Poly-view contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iHcTLIor0m.

James B Simon, Maksis Knutins, Liu Ziyin, Daniel Geisz, Abraham J Fetterman, and Joshua Albrecht. On the stepwise nature of self-supervised learning. In *International Conference on Machine Learning*, pp. 31852–31876. PMLR, 2023.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *Journal of Machine Learning Research*, 22(281):1–31, 2021.

Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. Mi{ce}: Mixture of contrastive experts for unsupervised image clustering. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=gV3wdEOGy_V.

Y-H Tsai, Y Wu, R Salakhutdinov, and L-P Morency. Self-supervised learning from a multi-view perspective. In *Proceedings of the International Conference on Learning Representations (ICLR), 2021*, 2021b.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=EhYjZy6e1gJ.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Xin Wang, Yasheng Wang, Yao Wan, Jiawei Wang, Pingyi Zhou, Li Li, Hao Wu, and Jin Liu. Code-mvp: Learning to represent source code from multiple views with contrastive pre-training. In *NAACL-HLT*, 2022b. URL https://api.semanticscholar.org/CorpusID:248512635.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3023–3032, 2020. URL https://api.semanticscholar.org/CorpusID:227012687.

Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022c.

Zhaoqing Wang, Ziyu Chen, Yaqian Li, Yandong Guo, Jun Yu, Mingming Gong, and Tongliang Liu. Mosaic representation learning for self-supervised visual pre-training. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JAezPMehaUu.

Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pp. 11112–11122. PMLR, 2021.

Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. Understanding contrastive learning via distributionally robust optimization. *ArXiv*, abs/2310.11048, 2023. URL https://api.semanticscholar.org/CorpusID:264172924.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. URL https://api.semanticscholar.org/CorpusID:4591284.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. *Proceedings of the ACM Web Conference 2022*, 2022. URL https://api.semanticscholar.org/CorpusID:246634345.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

Lingling Xu, Haoran Xie, Zongxi Li, Fu Lee Wang, Weiming Wang, and Qing Li. Contrastive learning models for sentence representations. *ACM Transactions on Intelligent Systems and Technology*, 14(4):1–34, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *ArXiv*, abs/2010.13902, 2020. URL https://api.semanticscholar.org/CorpusID:225076220.

Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6995–7004, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Zhonghang Li, and Siu keung Yiu. Automated spatio-temporal graph contrastive learning. *Proceedings of the ACM Web Conference 2023*, 2023. URL https://api.semanticscholar.org/CorpusID:258333641.

Shaofeng Zhang, Feng Zhu, Junchi Yan, Rui Zhao, and Xiaokang Yang. Zero-CL: Instance and feature decorrelation for negative-free symmetric contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RAW9tCdVxLj.

Xu Zheng, Tianchun Wang, Wei Cheng, Aitian Ma, Haifeng Chen, Mo Sha, and Dongsheng Luo. Parametric augmentation for time series contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EIPLdFy3vp.

Yanqiao Zhu, Yichen Xu, Feng Yu, Shu Wu, and Liang Wang. Cagnn: Cluster-aware graph neural networks for unsupervised graph representation learning. *arXiv preprint arXiv:2009.01674*, 2020.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In Katsushi Ikeuchi, Cristoph Schnörr, Josef Sivic, and René Vidal (eds.), *Proceedings of the 15th International Conference on Computer Vision (ICCV'15)*, pp. 19–27, Santiago, Chile, December 2015. IEEE.

Yongshuo Zong, Oisin Mac Aodha, and Timothy Hospedales. Self-supervised multimodal learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

A Appendix

In these appendices, we present the proofs for Theorems 3.1 and 4.2 in Sections A.2 and A.3, respectively. Additionally, we provide details on the updates to the embedding function $\mathcal{E}$ in Section A.5, followed by a series of supplementary experiments in Section A.6.

A.1 PRELIMINARIES

Firstly, let us begin with a Lemma that will be proven useful in later sections.

Lemma A.1. *Let $\mathcal{E}^*, E^*$ be the optimal embedding and encoder functions such that for an unconstrained hypothesis $\mathcal{H} = \{(\mathcal{E}, E) | \mathcal{E} : \Sigma \to \mathbb{R}^d, E : \mathrm{P}, \mathcal{Y} \to \mathbb{R}^d\}$ we minimize the conditional risk associated with the infoNCE loss:*

$$\mathcal{E}^*, E^* = \arg\min_H \left\{ \mathop{\mathbb{E}}_{i|y,\rho,\sigma_1,\ldots,\sigma_n} \left[-\log \left(\frac{\exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(y,\rho)))}{\sum_j \exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(y,\rho)))} \right) \right] \right\}$$

Where i represents the random variable that associates the context-label pair (ρ, y) with the correct symbol from $\sigma_1, \ldots, \sigma_n$.

Then

$$\forall \rho, y, \sigma_1, \ldots, \sigma_n : p(\sigma_i|\rho, y) = \exp(\mathrm{sim}(\mathcal{E}^*(\sigma_i), E^*(y,\rho)))p(\sigma_i) \sum_j \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)}$$

Proof. Let us begin by considering the conditional risk associated with the infoNCE loss:

$$\mathcal{R}(\mathcal{E}, E, \mathcal{D}) = \mathop{\mathbb{E}}_{i|y,\rho,\sigma_1,\ldots,\sigma_n} \left[-\log \left(\frac{\exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(y,\rho)))}{\sum_j \exp(\mathrm{sim}(\mathcal{E}(\sigma_j), E(y,\rho)))} \right) \right]$$

For brevity, let us denote f_i as the term $\exp(\mathrm{sim}(\mathcal{E}(\sigma_i), E(\rho, y)))$. Then, we obtain:

$$\mathcal{R}(\mathcal{E}, E, \mathcal{D}) = \mathop{\mathbb{E}}_{i|y,\rho,\sigma_1,\ldots,\sigma_n} \left[-\log \left(\frac{f_i}{\sum_j f_j} \right) \right]$$

It should be fairly easy to see that this term amounts to a cross-entropy term between a one-hot distribution, e_i, and the predicted distribution $f = \left[\frac{f_0}{\sum_j f_j}, \ldots, \frac{f_n}{\sum_j f_j} \right]$:

$$\mathcal{R}(\mathcal{E}, E, \mathcal{D}) = \mathop{\mathbb{E}}_{i|y,\rho,\sigma_1,\ldots,\sigma_n} \left[-e_i^T \log(f) \right] = \mathop{\mathbb{E}}_{i|y,\rho,\sigma_1,\ldots,\sigma_n} \left[H(e_i, f) \right]$$

It is a standard result that the probability distribution minimizing the conditional expectation of the cross-entropy, $\arg\min_f \mathbb{E}_{y|x}[H(e_y, f)]$, is $p(y|x)$ Hastie et al. (2009). Applying this result to our context, we conclude that:

$$f_i^* = p(i|\rho, y, \sigma_1 \ldots, \sigma_n)$$

Furthermore, as shown in van den Oord et al. (2018):

$$\begin{aligned}
p(i|\rho, y, \sigma_1, \ldots, \sigma_n) &= \frac{p(\sigma_i|\rho, y) \prod_{k \neq i} p(\sigma_k)}{\sum_j p(\sigma_j|\rho, y) \prod_{k \neq j} p(\sigma_k)} \\
&= \frac{\frac{p(\sigma_i|\rho,y)}{p(\sigma_i)} p(\sigma_i) \prod_{k \neq i} p(\sigma_k)}{\sum_j \frac{p(\sigma_j|\rho,y)}{p(\sigma_j)} p(\sigma_j) \prod_{k \neq j} p(\sigma_k)} \\
&= \frac{\frac{p(\sigma_i|\rho,y)}{p(\sigma_i)} \prod_k p(\sigma_k)}{\sum_j \frac{p(\sigma_j|\rho,y)}{p(\sigma_j)} \prod_k p(\sigma_k)} \\
&= \frac{\frac{p(\sigma_i|\rho,y)}{p(\sigma_i)}}{\sum_j \frac{p(\sigma_j|\rho,y)}{p(\sigma_j)}}
\end{aligned}$$

Therefore, we have:

$$\frac{\frac{p(\sigma_i|\rho,y)}{p(\sigma_i)}}{\sum_j \frac{p(\sigma_j|\rho,y)}{p(\sigma_j)}} = p(i|\rho, y, \sigma_1, \ldots, \sigma_n) \implies$$
$$\frac{\frac{p(\sigma_i|\rho,y)}{p(\sigma_i)}}{\sum_j \frac{p(\sigma_j|\rho,y)}{p(\sigma_j)}} = f_i^* \implies$$
$$p(\sigma_i|\rho, y) = f_i^* p(\sigma_i) \sum_j \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)} \implies$$
$$p(\sigma_i|\rho, y) = \exp(\mathrm{sim}(\mathcal{E}^*(\sigma_i), E^*(y, \rho)))p(\sigma_i) \sum_j \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)}$$

Notice that we made no assumptions about ρ, y, or $\sigma_1, \ldots, \sigma_n$, except that one symbol is distributed conditionally on the context-label pair, while the others are not. Therefore, we conclude that the statement of the lemma holds for any choice of these variables.

□

A.2 THEOREM 3.1

Let us consider the first theorem:

Theorem. *Given the symbols $u, v \in \Sigma$ such that*

1. $u \mathrel{\hat{=}} v$,
2. $\forall \rho : p(\rho|u) = p(\rho|v)$, *and*
3. *if there are $(\rho_1, y_1), \ldots, (\rho_d, y_d)$ context-label pairs such that $E^*(\rho_i, y_i)$ form a basis for $\mathbb{R}^d$*

then $\mathcal{E}^*(u) = \mathcal{E}^*(v)$

This theorem states the condition for SimCLR to encode semantically equivalent symbols into equivalent embeddings.

Proof. Since $u \mathrel{\hat{=}} v$, (i.e. $\forall \rho : p(y|u, \rho) = p(y|v, \rho)$), we can write:

$$p(u|y, \rho) = \frac{p(u, y, \rho)}{p(y, \rho)} = \frac{p(y|u, \rho)p(u, \rho)}{p(y, \rho)} = \frac{p(y|v, \rho)p(u, \rho)}{p(y, \rho)} = \frac{p(y, v, \rho)p(u, \rho)}{p(y, \rho)p(v, \rho)} = \frac{p(v|y, \rho)p(u, \rho)}{p(v, \rho)}$$

Or,

$$\begin{aligned} p(u|y, \rho)p(v, \rho) &= p(v|y, \rho)p(u, \rho) \\ p(u|y, \rho)p(\rho|v)p(v) &= p(v|y, \rho)p(\rho|u)p(u) \\ (*)\ \exp(\mathrm{sim}(\mathcal{E}^*(u), E^*(y, \rho)))p(u)p(\rho|v)p(v) &= \exp(\mathrm{sim}(\mathcal{E}^*(v), E^*(y, \rho)))p(v)p(\rho|u)p(u) \\ \exp(\mathrm{sim}(\mathcal{E}^*(u), E^*(y, \rho)))p(\rho|v) &= \exp(\mathrm{sim}(\mathcal{E}^*(v), E^*(y, \rho)))p(\rho|u) \end{aligned}$$

The step $(*)$ can be concluded by considering Lemma A.1. Take a context-label pair ρ, y sampled with symbols $\sigma_1, \ldots, \sigma_N$, one of which will be u and one of which will be v. Further, let i_u and i_v be the indices of u and v in the set of symbols. By Lemma A.1, we have:

$$p(u|\rho, y) = \exp(\mathrm{sim}(\mathcal{E}^*(u), E^*(y, \rho)))p(u) \left(\frac{p(u|\rho, y)}{p(u)} + \frac{p(v|\rho, y)}{p(v)} + \sum_{\substack{j \\ j \neq i_u, i_v}} \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)} \right)$$

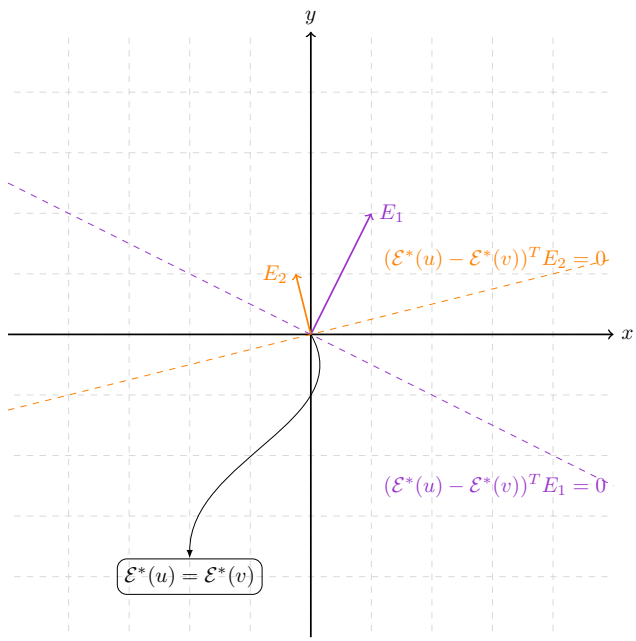


Figure 3:
2−dimensional representation of: $(\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_1 = 0$ (purple) and $(\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_2 = 0$ (orange). E_1 and E_2 have to form a basis of $\mathbb{R}^2$ in order for the solution to be unique.

A similar equation can be written for v:

$$p(v|\rho, y) = \exp(\text{sim}(\mathcal{E}^*(v), E^*(y, \rho)))p(v) \left(\frac{p(u|\rho, y)}{p(u)} + \frac{p(v|\rho, y)}{p(v)} + \sum_{\substack{j \\ j \neq i_u, i_v}} \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)} \right)$$

Since both summation term are equal, in equation $(*)$ they can be canceled out. Further, me made no assumption from the context-label pair ρ, y. Therefore, by applying the hypothesis $p(\rho|u) = p(\rho|v)$ we can write:

$$\forall \rho, y : \exp(\mathcal{E}^*(u)^T E^*(y, \rho)) = \exp(\mathcal{E}^*(v)^T E^*(y, \rho))$$

Next consider $(\rho_1, y_1), \ldots, (\rho_d, y_d)$ context-label pairs such that $E^*(y_i, \rho_i) = E_i$ form a basis of $\mathbb{R}^d$:

$$\begin{cases} \exp(\mathcal{E}^*(u)^T E_1) = \exp(\mathcal{E}^*(v)^T E_1) \implies (\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_1 = 0 \\ \ldots \\ \exp(\mathcal{E}^*(u)^T E_d) = \exp(\mathcal{E}^*(v)^T E_d) \implies (\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_d = 0 \end{cases}$$

The above system has d equations stating that $\mathcal{E}^*(u) - \mathcal{E}^*(v)$ is perpendicular to every basis vector $E_0, \ldots, E_d$. Thus, Only one solution is possible: $\mathcal{E}^*(u) = \mathcal{E}^*(v)$. Fig. 3 provides a graphical representation of the system when $d = 2$. In this case, we have only two equations whose solutions lie onto a line. The solution is unique when E_1 and E_2 are linearly independent (i.e., form a basis for $\mathbb{R}^2$). □

A.3 THEOREM 4.2

Let us consider the first theorem:

Theorem. *Given the symbols $u, v \in \Sigma$ such that:*

1. $\forall \rho : p(\rho|u) = p(\rho|v)$.
2. *If there are $\rho_1, \dots, \rho_d$ contexts such that $E^*(\rho_i)$ form a basis for $\mathbb{R}^d$*

then $\mathcal{E}^*(u) = \mathcal{E}^*(v)$

This theorem states the condition for our approach to encode conditionally equivalent symbols into equivalent embeddings. For completeness, we provide the complete proof to this theorem. However, this is an almost exact match to the proof of Theorem A.2.

Proof. Let us begin by considering the hypothesis: $p(\rho|u) = p(\rho|v)$:

$$\begin{aligned}
\forall \rho : p(\rho|u) = p(\rho|v) \implies \\
\forall \rho : \frac{p(\rho, u)}{p(u)} = \frac{p(\rho, v)}{p(v)} \implies \\
\forall \rho : \frac{p(u|\rho)p(\rho)}{p(u)} = \frac{p(v|\rho)p(\rho)}{p(v)} \implies \\
(*)\forall \rho : \exp(\mathcal{E}^*(u)^T E^*(\rho)) = \exp(\mathcal{E}^*(v)^T E^*(\rho))
\end{aligned}$$

The step $(*)$ follows, once again, from an application of Lemma A.1. Next consider $\rho_1, \dots, \rho_d$ contexts such that $E^*(\rho_i) = E_i$ form a basis of $\mathbb{R}^d$:

$$\begin{cases}
\exp(\mathcal{E}^*(u)^T E_1) = \exp(\mathcal{E}^*(v)^T E_1) \implies (\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_1 = 0 \\
\dots \\
\exp(\mathcal{E}^*(u)^T E_d) = \exp(\mathcal{E}^*(v)^T E_d) \implies (\mathcal{E}^*(u) - \mathcal{E}^*(v))^T E_d = 0
\end{cases}$$

The above system has d equations stating that $\mathcal{E}^*(u) - \mathcal{E}^*(v)$ is perpendicular to every basis vector $E_0, \dots, E_d$. Thus, Only one solution is possible: $\mathcal{E}^*(u) = \mathcal{E}^*(v)$. Fig. 3 provides a graphical representation of the system when $d = 2$. In this case, we have only two equations whose solutions lie onto a line. The solution is unique when E_1 and E_2 are linearly independent (i.e., form a basis for $\mathbb{R}^2$). □

A.4 Proposition A.2

Now, let us introduce a discuss the proposition for the backward implication of the Corollary 4.3.

Proposition A.2. *Given symbols u and v such that $\forall \rho : p(\rho|u) = p(\rho|v)$. Given optimal embedding function $\mathcal{E}^*$. Then:*

$$\mathcal{E}^*(u) = \mathcal{E}^*(v) \implies u \mathrel{\mathring{=}} v$$

The proof is derived in a simila manner to the Theorem 3.1.

Proof. Take a context-label pair ρ, y sample with symbols $\sigma_1, \dots, \sigma_N$, one of which will be u and one of which will be v. Further, let i_u and i_v be te indices of u and v in the set of symbols. Then, by Lemma A.1, we know that:

$$p(u|\rho, y) = \exp(\text{sim}(\mathcal{E}^*(u), E^*(y, \rho)))p(u) \left(\frac{p(u|\rho, y)}{p(u)} + \frac{p(v|\rho, y)}{p(v)} + \underbrace{\sum_{\substack{j \\ j \neq i_u, i_v}} \frac{p(\sigma_j|\rho, y)}{p(\sigma_j)}}_{\alpha} \right)$$

By isolating the exponential term, we get:

$$\frac{p(u|\rho,y)}{\left(\frac{p(u|\rho,y)}{p(u)}+\frac{p(v|\rho,y)}{p(v)}+\alpha\right)p(u)}=\exp(\text{sim}(\mathcal{E}^*(u),E^*(y,\rho)))$$

Similarly, for v, we have:

$$\frac{p(v|\rho,y)}{\left(\frac{p(u|\rho,y)}{p(v)}+\frac{p(v|\rho,y)}{p(v)}+\alpha\right)p(v)}=\exp(\text{sim}(\mathcal{E}^*(v),E^*(y,\rho)))$$

By hypothesis, we know that $\mathcal{E}^*(u)=\mathcal{E}^*(v)$. Therefore, the exponentials terms are equal. This implies the following:

$$\frac{p(u|\rho,y)}{\left(\frac{p(u|\rho,y)}{p(u)}+\frac{p(v|\rho,y)}{p(v)}+\alpha\right)p(u)}=\frac{p(v|\rho,y)}{\left(\frac{p(u|\rho,y)}{p(v)}+\frac{p(v|\rho,y)}{p(v)}+\alpha\right)p(v)}$$
$$\frac{p(u|\rho,y)}{p(u)}=\frac{p(v|\rho,y)}{p(v)}$$

Note that, no assumption is made on ρ or y. Therefore, using $p(\rho|u)=p(\rho|v)$ concludes the result ($u \mathrel{\mathring{=}} v$).

□

A.5 Embedding Symbol Function Update Rule

Let us consider the infoNCE loss with the scalar product as similarity function:

$$\mathcal{L}(D,\theta)=-\sum_i \log\left(\frac{\exp\left(\mathcal{E}(\sigma_i;\theta)^T E(y_i,\rho_i;\theta)\right)}{\sum_j \exp(\mathcal{E}(\sigma_j;\theta)^T E(y_j,\rho_j;\theta))}\right)$$

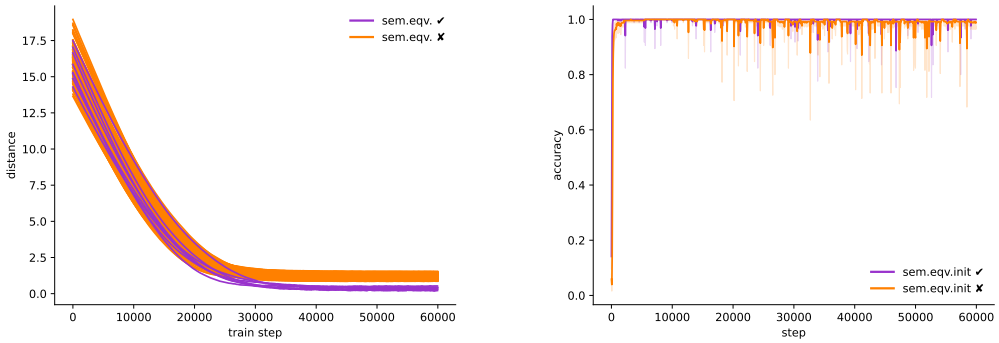
Now, let us assume that $\mathcal{E}$ maps each symbol, w_i, to a corresponding parameter vector, denoted as $\overline{w}_i$. For simplicity, let us call $E(y_j,\rho_j;\theta)=\overline{c}_j$. Next, we differentiate the infoNCE loss wrt. $\overline{w}_i$.

$$\begin{aligned}\frac{\partial\mathcal{L}(D,\theta)}{\partial\overline{\sigma}_k}=&\frac{\partial}{\partial\overline{\sigma}_k}\left(-\sum_i\log\left(\frac{e^{\overline{\sigma}_i^T\overline{c}_j}}{\sum_j e^{\overline{\sigma}_i^T\overline{c}_j}}\right)\right)\\ =&-\sum_i\delta_{ik}\overline{c}_i-\frac{1}{\sum_j e^{\overline{\sigma}_j^T\overline{c}_j}}\sum_j e^{\overline{\sigma}_i^T\overline{c}_j}\delta_{ik}\overline{c}_j\\ =&-\overline{c}_k+\sum_j\underbrace{\frac{e^{\overline{\sigma}_k^T\overline{c}_j}}{\sum_t e^{\overline{\sigma}_k^T\overline{c}_t}}}_{0\le\alpha_j\le1}\overline{c}_j=-\overline{c}_k+\sum_j\alpha_j\overline{c}_j\end{aligned}$$

Thus, the update rule with learning rate η:

$$\overline{\sigma}_k\leftarrow\overline{\sigma}_k+\eta(\overline{c}_k-\sum_j\alpha_j c_j)$$

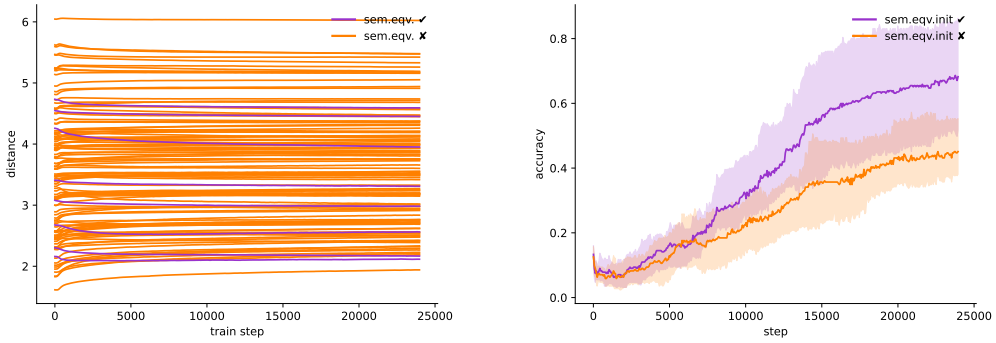
Since $\sum_j\alpha_j=1$, the update on $\overline{\sigma}_k$ pushes towards $\overline{c}_k$ and away from the weighted sum of all $\overline{c}_j$.



(a) Symbol embedding distances

(b) Validation accuracy

Figure 4: Embedding distances and validation accuracy for bigger architectures



(a) Symbol embedding distances

(b) Validation accuracy

Figure 5: Embedding distances and validation accuracy for architecture without weight decay

A.6 Experiments

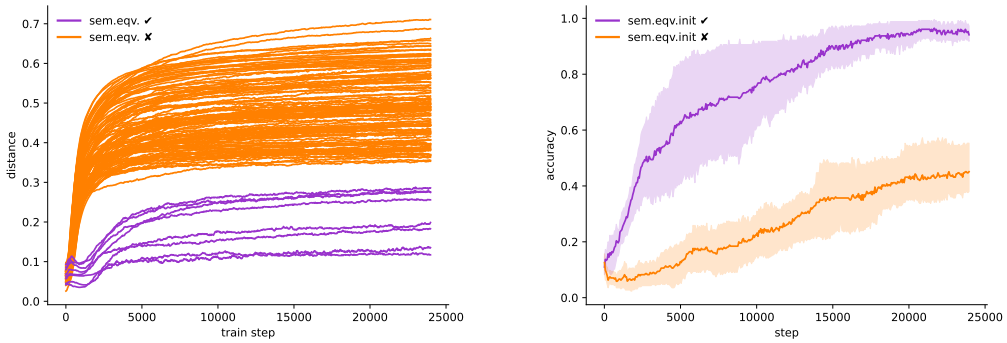
A.6.1 Bigger Embeddings

One of the hypotheses in Theorems 3.1 and 4.2 concerns the existence of context-label pairs that form a basis for $\mathbb{R}^d$, where d is the embedding dimension. Consequently, when the embedding dimension is sufficiently large, there is no guarantee that the model will encode semantically equivalent symbols in close proximity. However, even when the embedding dimension is increased to 128 (which is 16 times larger than required by the theorems), the model still represents semantically equivalent symbols near each other (see Figure 4a).

In Figure 4b, we report the validation accuracy for a model initialized with pre-trained representations (shown in purple) and the same model with randomly initialized symbol representations (shown in orange). Both models perform equally well, suggesting that if the model's capacity is sufficiently large, it can effectively compensate for differences in initialization.

A.7 No Weight Decay

It is instructive to introduce a failure case. In this experiment, we disable weight decay during training, removing the regularization that encourages the model to use low-magnitude weights. The results are shown in Figure 5a. Without weight decay, the model fails to encode semantically equivalent symbols (shown in purple) close to each other within a reasonable time. However, a trend toward pushing semantically equivalent symbols closer together is still observable. This results in



(a) Symbol embedding distances

(b) Validation accuracy

Figure 6: Embedding distances and validation accuracy for architecture without weight decay but with low embedding initialization

semantic equivalence relationships that are only marginally better than those produced by a randomly initialized model.

The effect of poor semantic encoding is visualized in Figure 5b. The model initialized with pre-trained embeddings (shown in purple) performs only slightly better than the randomly initialized model (shown in orange).

Furthermore, we speculate that weight decay may have additional effects on the model's capacity to encode semantic equivalence relationships. As weight decay pushes all embeddings toward zero, the model is discouraged from encoding semantically different symbols too close to each other, as doing so would eventually make it unable to distinguish between them. This limitation does not apply to semantically equivalent symbols; if they become indistinguishable, the model's performance would still be correct.

A.8 No Weight Decay & Low Weight Initialization

The failure caused by the absence of weight decay can be mitigated by employing a lower initialization scheme. In this experiment, we initialize the embeddings using a normal distribution with a mean of 0 and a standard deviation of 0.02. The results are shown in Figure 6a. Under this scheme, the model is able to encode semantically equivalent symbols (shown in purple) close to each other within a reasonable time. Additionally, the model initialized with pre-trained embeddings (shown in purple) performs better than the randomly initialized model (shown in orange), as seen in Figure 6b.

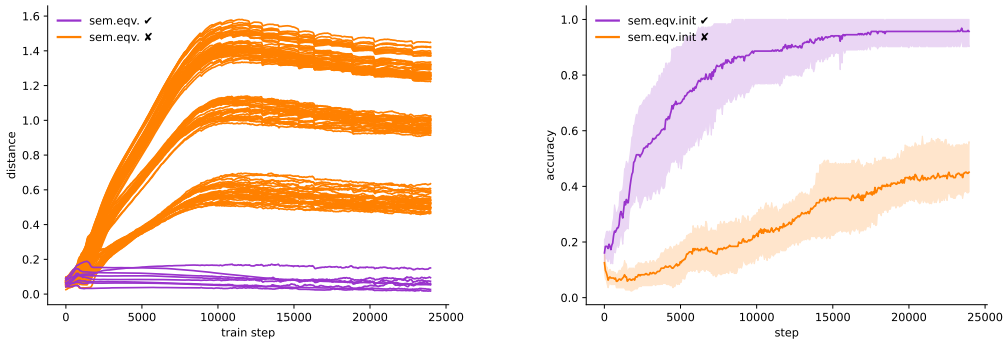
Since all the embeddings are initialized with low-magnitude values, the distance between symbols starts small, compared to the standard PyTorch initialization, which uses a normal distribution with a mean of 0 and a standard deviation of 1 [1], as shown in Figure 2a.

A.9 Weight Decay & Sligthly Low Weight Initialization

Another scenario worth visualizing is the case where we employ both weight decay and low-weight initialization. The results are shown in Figure 7a. In this setup, the model successfully encodes semantically equivalent symbols (shown in purple) close to each other within a reasonable time. Consequently, the model initialized with pre-trained embeddings (shown in purple) outperforms the randomly initialized model (shown in orange), as seen in Figure 7b.

Interestingly, despite the presence of weight decay, the distances between semantically different embeddings end up being larger compared to the previous experiment, which did not use weight

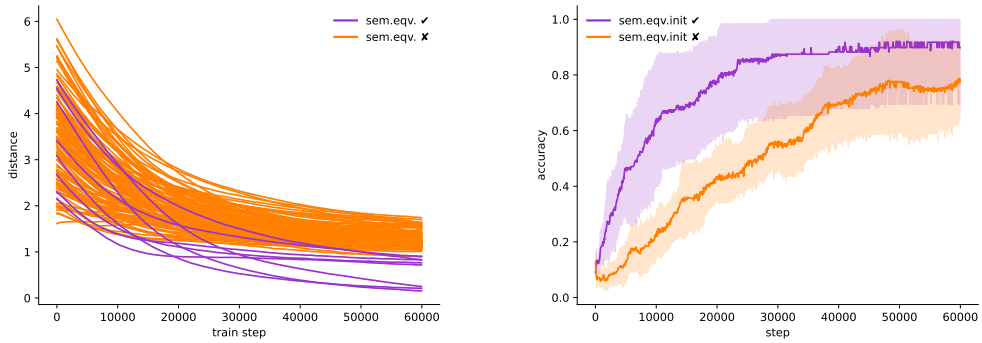
[1] `https://github.com/pytorch/pytorch/blob/v2.4.0/torch/nn/modules/sparse.py#L14`



(a) Symbol embedding distances

(b) Validation accuracy

Figure 7: Embedding distances and validation accuracy for architecture with weight decay but with low embedding initialization



(a) Symbol embedding distances

(b) Validation accuracy

Figure 8: Embedding distances and validation accuracy for a slightly misaligned contrastive and classification datasets

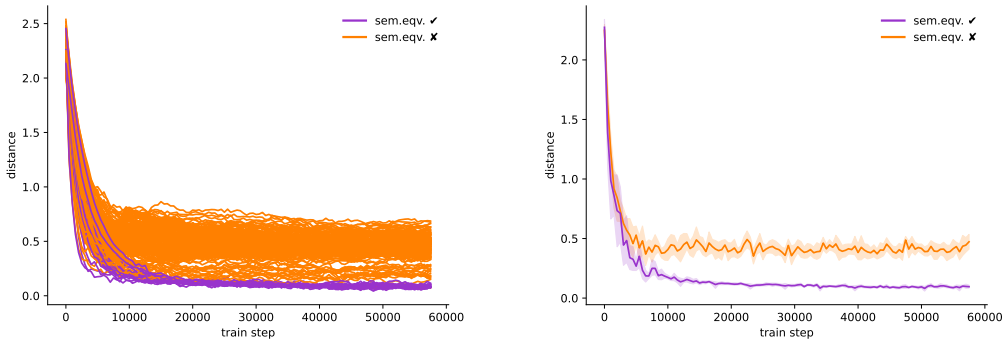
decay. Additionally, the distances between semantically different embeddings appear divided into three distinct bands. Unfortunately, we do not yet have a clear explanation for this behavior.

A.10 Slightly Misaligned Contrastive Dataset

The proposed theory suggests that when a contrastive dataset and a classification dataset are distributionally aligned (see Definition 4.1), training on the contrastive dataset should produce embeddings that are semantically meaningful for the target classification task. But what happens if the distributional alignment is imperfect? In this experiment, we introduce a slight misalignment between the contrastive and classification datasets.

The misalignment is created by assigning a fixed probability to all correct contexts of a particular symbol. Specifically, we assign a probability of 0.9 to one chosen context, while the remaining probability is evenly distributed among all other contexts.

The results are presented in Figure 8a and 8b. The model is still able to position semantically equivalent symbols (shown in purple) close to each other in a reasonable amount of time. Furthermore, the model initialized with pre-trained embeddings (shown in purple) outperforms the randomly initialized model (shown in orange), as seen in Figure 8b. However, compared to the perfectly aligned case (see Figures 2a and 2b), the model takes longer to bring semantically equivalent symbols closer together. As a result, the generated embeddings are less effective for the classification task.



(a) Symbol embedding distances between 40 symbols-

(b) Symbol embedding distances averaged between semantically equivalent and semantically different groups.

Figure 9: Embedding distances for the Word2Vec task. The train steps corresponds to 1% of the Bookcorpus dataset.

A.11 Word Embeddings

Until now, we have exclusively experimented with algorithmic data. However, real-world data are far from being as clean as the data we have used so far. In this experiment, we aim to train a Word2Vec-like model Mikolov et al. (2013) on a small text corpus. Specifically, we will use the BookCorpus dataset Zhu et al. (2015) as our source. To access this dataset[2], we will utilize Hugging Face's datasets library Lhoest et al. (2021). Our objective is to observe whether the patterns identified in algorithmic data are replicated in this real-world context.

We will artificially inject semantic equivalences into the dataset by replacing certain words with fabricated synonyms. For instance, a word such as good may be replaced with equal probability by good_1 or good_2 within the dataset. Specifically, we selected 100 words and replaced each with one of the fabricated synonyms.

To generate symbol-context pairs, we will sample a sentence and randomly mask one word. The masked word will serve as the symbol, while the remainder of the sentence will act as the context.

We will train a Transformer encoder-only model[3] to generate context-embedding. The symbol embedding will be generated by a simple embedding layer. [4] As previously, we will use the InfoNCE loss van den Oord et al. (2018) to match the context-embedding with the correct symbol-embedding, while pushing away the incorrect symbol-embeddings.

During training, we expect the embeddings of semantically equivalent symbols to converge, while embeddings of unrelated symbols should remain far apart. This behavior is confirmed in Figure 9. Specifically, we observe that the embeddings of semantically equivalent symbols (shown in purple) converge to a small distance, whereas the embeddings of semantically different symbols (shown in orange) remain distant. Notably, the distances between semantically different symbols are more spread out compared to the algorithmic data. This occurs because some words are inherently semantically similar, resulting in closer embeddings.

[2] https://huggingface.co/datasets/bookcorpus/bookcorpus
[3] https://pytorch.org/docs/stable/generated/torch.nn.Transformer.html
[4] https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html