SALSA: Semantically-Aware Latent Space Autoencoder

Kathryn E. Kirchoff
University of North Carolina
Department of Computer Science
kat@cs.unc.edu

Travis Maxfield
University of North Carolina
Eshelman School of Pharmacy
tmaxfield@unc.edu

Alexander Tropsha*
University of North Carolina
Eshelman School of Pharmacy
alex_tropsha@unc.edu

Shawn M. Gomez†
University of North Carolina
Eshelman School of Pharmacy
smgomez@gmail.com

Abstract

In learning molecular representations, SMILES strings enable the use of powerful NLP methodologies, such as sequence autoencoders. However, an autoencoder trained solely on SMILES is insufficient to learn molecular representations that are semantically meaningful, which capture structural similarities between molecules. We demonstrate by example that a standard SMILES autoencoder may map structurally similar molecules to distant latent vectors, resulting in an incoherent latent space. To address this shortcoming we propose Semantically-Aware Latent Space Autoencoder (SALSA), a transformer-autoencoder modified with a contrastive objective of mapping structurally similar molecules to nearby vectors in the latent space. We evaluate semantic awareness of SALSA representations by comparing to a naive autoencoder as well as the standard ECFP4. We show empirically that SALSA learns a representation that maintains 1) structural awareness, 2) physicochemical property awareness, 3) biological property awareness, and 4) semantic continuity.

1 Introduction

In drug discovery, the availability of high-quality molecular representations underpins the success of computational tasks such as property prediction, virtual screening, and *de novo* generation. High-quality representations are those that demonstrate effective *semantic awareness*—an attribute that, in general, amounts to mapping similar data instances to similar feature vectors. In the case of chemical data, the similar property principle (SPP) [11] states that structurally (i.e. graphically) similar molecules tend to have similar properties, suggesting structural similarity as a reasonable notion of chemical similarity.

Traditional molecular representations rely on handcrafted features, one such representation being molecular fingerprints. Extended-Connectivity Fingerprints (ECFPs) [22] are among the most prominent fingerprinting methods. ECFPs are in fact designed to encode the graphical structure of chemicals, and thus capture a notion of structural similarity. However, their utility is limited in that they do not provide generative capabilities, nor can they be readily modified to accommodate specific use cases. Recently, advancements in deep learning have given rise to expressive methodologies capable of learning rich and flexible representations. In particular, generative pre-training of sequence autoencoders, especially transformers, has proven to be a highly effective form of representation

*Corresponding author
†Corresponding author

The NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development (AI4D3 2023), New Orleans, LA, USA, 2023.

learning that also naturally results in a generative decoder [17, 18]. These techniques can be leveraged for molecular representation learning provided chemical data are encoded as sequences, such as SMILES strings [32]. Thus, training a SMILES-based autoencoder [7, 31] is one approach for potentially powerful molecular representations.

However, we observe that SMILES-based autoencoders do not adequately learn the structure-based semantics that underlie chemical datasets. As a result, these models may map semantically (i.e., structurally) similar molecules to distant codes in the latent space. This phenomenon is more precisely defined as an instance in which semantically similar molecules (having low GED) are mapped to distant latent representations (having high Euclidean distance); this is illustrated in left panel of Figure 1. Collectively, many of these semantically-naive events induce a disorganized latent space, limiting success in downstream applications.

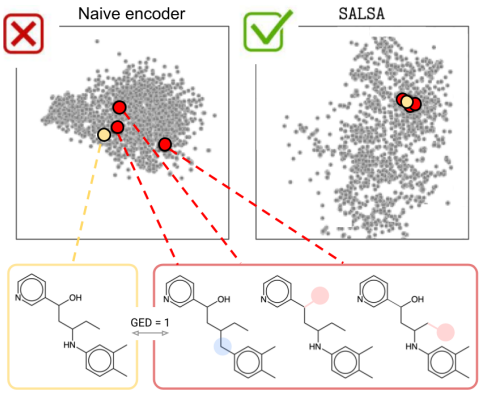

Figure 1: Naive SMILES encoders may map structurally similar molecules to distant codes (Left). Our method, SALSA, learns a semantically aware representation, mapping similar molecules to nearby codes (Right).

We seek to improve the semantic awareness of these SMILES-based representations. Specifically, we aim to achieve structural awareness, such that similar molecular graphs are mapped to similar latent codes (an example of this desired outcome is shown in the right panel of Figure 1). To do so, we propose indirectly injecting information about structural similarity into a SMILES-based autoencoder through a contrastive objective, such that structurally similar molecules are mapped near one another in the latent space. Our proposed model, Semantically-Aware Latent Space Autoencoder (SALSA), is a SMILES-based transformer autoencoder modified with a contrastive task. The contrastive objective is to map structurally similar molecules, separated by a single graph edit, to similar codes in the effected latent space. Our model is trained on a custom dataset comprised of pairs of structurally similar molecules, designed specifically to accommodate our contrastive task. In this way, we are able to learn a semantically meaningful and continuous latent space. Collectively, our results indicate that through this implicit incorporation of structural information, SALSA learns a *general* molecular representation useful in a variety of drug discovery tasks.

Our contributions are as follows:

- We propose a novel framework for learning semantically-aware molecular representations, integrating a contrastive objective into an autoencoder framework, allowing us to implicitly inject semantic (structural) information onto the latent representations.
- We develop a scheme for constructing a chemical dataset suited to contrastive learning of molecular entities, specifically aimed at learning structural similarities between molecules.
- We evaluate the quality of SALSA representations by assessing: (1) structural organization within local neighborhoods, (2) property-based organization and associated data visualization, (3) similarity-based virtual screening, and (4) generative capacity through molecular interpolation.

2 Related Work

Sequence representations For our sequence-based (i.e. SMILES-based) representation, we are specifically interested in methods that allow for global representation of sequence inputs. Earlier methods aimed at embedding whole sequences utilized recurrent neural networks (RNNs), including long short-term memory networks (LSTMs), naturally aligned to this objective [4, 23]. However, most state-of-the-art methods are based on the original transformer architecture [27] and do not provide a global representation of the input. Recently, authors have modified the transformer architecture to include a bottleneck (or pooling) layer allowing for a single, fixed-size global embedding of the input [15, 10, 13]. Examples of RNN-based molecular representation models include ChemVAE [6] and AllSMILES VAE [1]. Transformer-based models include ChemBERTa [5], SMILESTransformer [8], and FragNet [24].

Contrastive learning In molecular modeling, both SMILES- and graph-based representations have been explored in the context of contrastive learning. The model proposed by Shrivastava and Kell utilized the normalized temperature-scaled cross entropy (NT-Xent) [25] loss to map enumerated SMILES [3] of identical molecules nearby in the latent space. Regarding graphs, Wang et al. similarly used the NT-Xent loss to maximize the agreement between pairs of augmented graphs ("views") derived from the same molecule; here, each view (i.e. positive sample) is obtained by masking out nodes or edges. The NT-Xent loss, although widely successful, operates solely on positive *pairs*, an issue addressed by Khosla et al. in their formulation of the Supervised Contrastive (SupCon) loss which allows for comparison among an arbitrarily sized *set* (rather than a pair) of positive instances.

3 Methods

Problem Setup Given a molecular dataset, we consider two modes of symbolic representation, the collection of SMILES $\mathcal{D} = \{s_i\}_{i=1}^N$ and the corresponding collection of molecular graphs $G = \{g_i\}_{i=1}^N$. We operate on the SMILES representations directly, learning a mapping from $\mathcal{D} \rightarrow \mathcal{Z} = \{z_i\}_{i=1}^N$. However, we want distances among $\mathcal{Z}$ to be informed by (graph edit) distances in the molecular graph space, G. To achieve this, we implicitly encode structural information via a contrastive objective that operates on similarity relationships defined in G, such that nearby codes in $\mathcal{Z}$ correspond to similar graphs in G.

Transformer Autoencoder We define our autoencoder with a transformer-based encoder and decoder, and an intermediate bottleneck to produce a latent embedding space. Combined with the contrastive component, the general framework is encapsulated in Figure 2. Note that the transformer autoencoder without the contrastive modification constitutes our naive baseline (denoted "Naive") for comparison.

Contrastive Objective As semantic awareness is often defined on some notion of similarity, a contrastive learning approach naturally aligns with our objective as it operates on some definition of similarity between data. For SALSA, we specify a contrastive task operating on pairs of "similar" and "dissimilar" molecules, where "similar" describes any two molecules having a graph edit distance (GED) of one. Here, GED between two molecular graphs is defined as the minimum number of single edits required to make one graph isomorphic to the other.

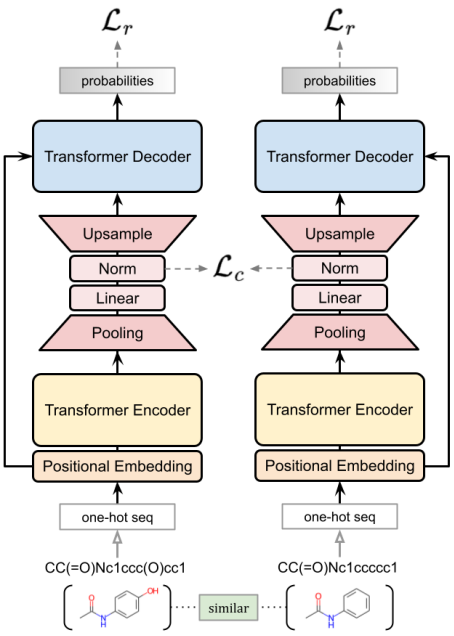

Figure 2: Overview of SALSA architecture. SMILES are input into the encoder, and the reconstruction objective ($\mathcal{L}_r$) is computed from decoder output. For a positive (similar) pair, the contrastive objective ($\mathcal{L}_c$) is to minimize the distance between the latent codes. Note that weights are "shared" between the two networks, i.e. only a single model is trained and used for inference.

This contrastive objective necessitates a dataset of known 1-GED molecular pairs. However, it is computationally infeasible to obtain all pairs of 1-GED molecules systematically from an existing dataset. To sidestep this issue, we propose a pipeline for generating a bespoke dataset of 1-GED molecular pairings suitable to a contrastive learning framework. We accomplish this by defining a set of single graph-edit transformations, or mutations, which are applied to "anchor" molecules in order to obtain similar molecules which we will refer to as "mutants". Our contrastive training set will be the resulting collection of anchors and respective mutants.

3.1 Contrastive Training Set

Anchor compounds We utilize the dataset developed by [16], which contains ∼1,500,000 SMILES sequences from the ChEMBL database (version ChEMBL21) [2]. (For an in-depth description of the curation process, please refer to [16].) We further filter out SMILES with sequence length greater

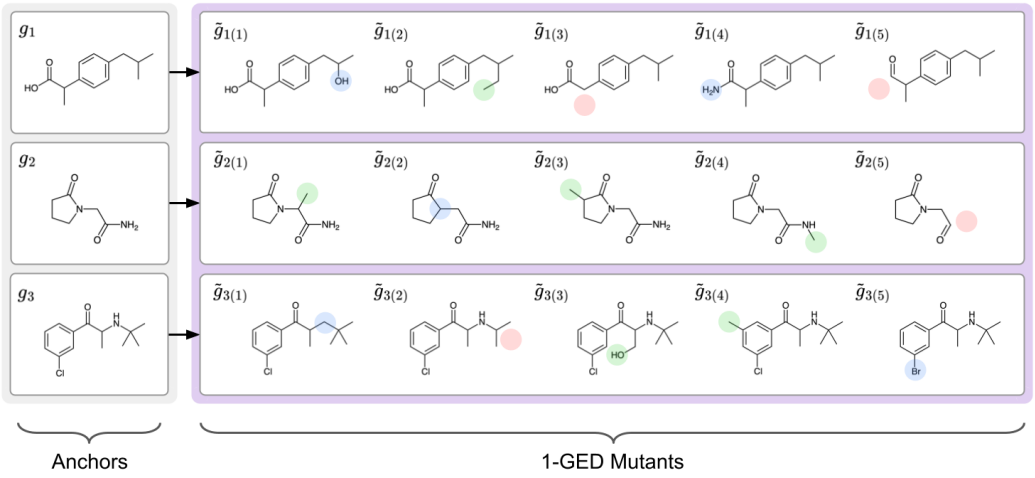

Figure 3: An example batch from the mutated dataset composed of three anchors, g_1, g_2, g_3, and their respective sets of mutants, $P(i) = \{\tilde{g}_{i1}, \tilde{g}_{i2}, \tilde{g}_{i3}, \tilde{g}_{i4}, \tilde{g}_{i5}\}$. True anchor-mutant couplings constitute positive pairs. Correspondingly, negative pairings are defined between anchors and all other molecules in the batch not in that anchor's set $P(i)$. Colored atoms of mutant compounds correspond to single graph edits from anchor to mutant: `add` (green), `replace` (blue), and `remove` (red).

than 110 characters. The remaining compounds, ∼1,250,000 in total, constitute the set of anchor molecules, G, from which we generate 1-GED mutants.

Generation of mutant compounds We define a molecular graph generally as $g = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_0, ..., v_A\}$ is the set of nodes, where each $v_a \in \{\texttt{C}, \texttt{O}, \texttt{N}, \texttt{S}, \texttt{Br}, \texttt{Cl}, \texttt{I}, \texttt{F}, \texttt{P}, \texttt{B}, \$\}$ (atom types), and $\mathcal{E} = \{(v_a, v_b) | v_a, v_b \in \mathcal{V}\}$ is the set of edges (bonds). Note that atom type \$ is a stand-in for any atom type not in the remaining list. We differentiate notation between anchors and mutants with a tilde, i.e. anchor graphs as g and mutant graphs as $\tilde{g}$. Given an anchor, we consider its graph, $g_i \in G$ where i is the index identifying the anchor in G. We obtain a mutated graph, or mutant, by randomly sampling a mutation operator $t(\cdot) \sim \mathcal{T}$ and applying that mutation to the anchor, $t(g_i) = \tilde{g}_{ij}$ where i again corresponds to the original anchor, and j is the index of the mutant graph within the anchors' positive sample set. The set of mutation operators, $\mathcal{T}$, is defined to avoid mutations that would alter the molecular backbone, i.e. breaking or forming rings or making a disconnected graph. Furthermore, we require mutants to be chemically valid molecular graphs, and we normalize all SMILES using the RDKit canonicalization algorithm [19]. We define three mutation operators:

- *Node addition* (`add`): Append a new node, and a corresponding edge, to an existing node.
- *Node substitution* (`replace`): Change the atom type of an existing node.
- *Node deletion* (`remove`): Remove a singly-attached node and its corresponding edge.

For both `add` and `replace`, incoming atom types are drawn from the observed atom type distribution in the original ChEMBL dataset. Examples of all three mutations are shown in Figure 3. We have now defined our curated set of node-level graph transformations, $\mathcal{T} = \{\texttt{add}, \texttt{replace}, \texttt{remove}\}$. For each anchor, g_i, we generate 10 distinct mutants that constitute the "positive" sample set, $P(i)$, for that anchor: $P(i) = \{\tilde{g}_{i1}, \tilde{g}_{i2}, \ldots, \tilde{g}_{i10}\} \in \tilde{G}$. The resulting training set was composed of the anchor compounds and their respective mutant compounds, amounting to ∼14,000,000 total compounds.

Faulty-Positive Filtering The Similar Property Principle (SPP) contends that structurally similar molecules tend to exhibit similar molecular properties [11]. However, some single graph edit mutations may effect great differences in the physicochemical properties between anchor and mutant, thus violating the SPP. We circumvent such phenomena by filtering out mutants that are *too dissimilar* from their respective anchor based on the Mahalanobis distance between the physicochemical properties of an anchor and those of its mutants. Mahalanobis distance between an anchor g_i and mutant $\tilde{g}_{ij}$ is defined as:

$$d_M\left(g_i, \tilde{g}_{ij}\right) = \sqrt{\left(x_i - \tilde{x}_{ij}\right) \Sigma^{-1} \left(x_i - \tilde{x}_{ij}\right)} \tag{1}$$

where x_i and $\tilde{x}_{ij}$ are the physicochemical property vectors for g_i and $\tilde{g}_{ij}$, respectively. Σ is the covariance matrix corresponding to the distribution of physicochemical properties computed over initial anchor set G. We computed physicochemical properties corresponding to the standard collected of RDKit [19] descriptors, and then filtered out descriptors having any invalid property values in order to obtain a real-valued property vector for each molecule (see Supplementary Material 7.2).

3.2 Modeling Framework

Architecture The core architecture of SALSA is based on the encoder-decoder transformer paradigm proposed by Vaswani et al. The SALSA transformer takes SMILES sequences as input and additionally considers the similarity relationships between those SMILES inputs (denoted either "similar" or "dissimilar"), as determined by their structural similarity. We modify the original transformer architecture by introducing a pooling layer and a subsequent upsampling layer between the encoder and decoder, and in this way impose an autoencoder capable of producing fixed-size latent representations. Specifically, whereas the intermediate output of the transformer encoder is a vector of size $\mathbb{R}^{L\times H}$ for a sequence of length L and hidden dimension size H, SALSA is designed to output a latent vector of fixed size $\mathbb{R}^S$. This is accomplished by first applying a component-wise mean pooling from $\mathbb{R}^{L\times H} \rightarrow \mathbb{R}^H$. We normalize the output of the "Pooling" layer onto the hypersphere embedded in $\mathbb{R}^S$ and then route to the contrastive loss and into the decoder, which, requires an additional dimension reshaping referred to as "Upsample" in Figure 2.

Loss Function We define a compound loss function, composed of (1) a contrastive term defined over a batch of inputs (a set of anchors and their respective mutants) and (2) a reconstruction term defined per input. For our contrastive task, we adapt the supervised contrastive (SupCon) loss [12]:

$$\mathcal{L}_c = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(z_i \cdot z_p/\tau\right)}{\sum_{a\in A(i)} \exp\left(z_i \cdot z_a/\tau\right)}, \tag{2}$$

where τ is the temperature, $A(i)$ is the set of all samples sharing a batch with instance i, with latent code z_i, and $P(i)$ are those elements of $A(i)$ that are similar to i, and I is the set of anchors in the batch, using the terminology of Sec. 3.1.

The autoencoder, operating on SMILES, is trained with a reconstruction loss with causal masking. For a single sequence s_i and its associated latent vector z_i, the loss is:

$$\mathcal{L}_r = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(s_i^{(t)}|z_i, s_i^{(<t)}), \tag{3}$$

where T is the length of the sequence s_i and $p_\theta(s_i^{(t)}|z_i, s_i^{(<t)})$ is the output of the decoder at position t along the sequence. The full reconstruction loss $\mathcal{L}_r$ is the average of all per-sequence losses. The final loss computation is a weighted combination of the two terms,

$$\mathcal{L} = \lambda\mathcal{L}_c + (1-\lambda)\mathcal{L}_r \tag{4}$$

where $0 \leq \lambda \leq 1$ is a hyperparameter that weights the contributions of the contrastive loss and the reconstruction loss, respectively. For our experiments, we train SALSA with $\lambda = 0.5$.

4 Experiments

Hierarchical tasks for semantic awareness According to the Similar Property Principle (SPP), structurally similar molecules tend to have similar properties, whether physicochemical (e.g. molecular weight, hydrophobicity) or biological (e.g. binding affinity) [11]. Through the SPP, we expect that a representation that demonstrates (1) structural awareness will correspondingly demonstrate some degree of (2) physicochemical property awareness as well as, to a lesser extent, (3) biological property awareness. Ordered (1)-(2)-(3), these modes of awareness are effectively ranked by increasing complexity, corresponding to higher orders of semantic awareness. Additionally, we consider the task of semantic continuity, which is notably difficult given the discrete nature of chemical space. To evaluate this, we ask to what extent can SALSA generate meaningful interpolants given similarly structured endpoints? Thus, we ask the following four questions guided by the SPP in the order of increased complexity, each associated with an evaluation task:

(1) **GED-EuD correlation** ☞ Does SALSA exhibit *structural* awareness?
(2) **Data visualization** ☞ Does SALSA encode information about *physicochemical* properties?
(3) **Virtual screening** ☞ Does SALSA generalize to tasks on *biological* properties?
(4) **Molecular interpolation** ☞ Does SALSA exhibit semantic *continuity*?

4.1 Structural Awareness

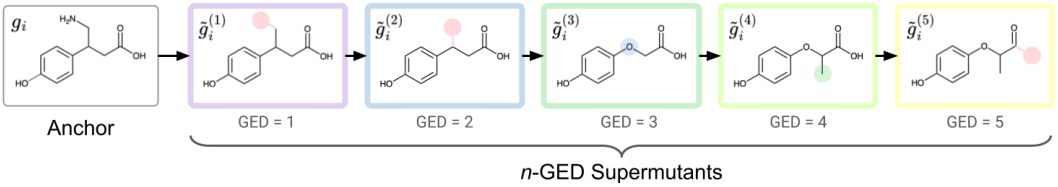

Figure 4: Example of supermutants and their originating anchor, g_i. Supermutants are color-coded according to n-GED (1-GED: purple, 2-GED: blue, 3-GED: green, etc.) from the anchor.

Our initial evaluation task aims to verify that SALSA achieves the explicit goal of structural awareness, such that similar molecular graphs are mapped to similar latent codes (shown in right panel of Figure 1). We aim to correlate the graph edit distance (GED) between molecules against their Euclidean distance (EuD) in latent space, but this necessitates *a priori* knowledge of GED between molecular pairs, the computing of which is computationally infeasible. In lieu of that, we opt to generate our own evaluation set of "supermutants" in a similar fashion to our generation of "mutants" for our training set in Section 3.1.

Supermutant evaluation set We extend our mutation process defined in Section 3.1 to iteratively generate sets of n-GED supermutants where $n \in \{1, 2, 3, 4, 5\}$. For a given anchor, we apply a random node-level mutation (`add`, `replace`, or `remove`) to generate a 1-GED (super)mutant, $\tilde{g}_i^{(1)} = t^{(1)}(g_i)$, to which another random mutation is applied to generate a 2-GED supermutant, $\tilde{g}_i^{(2)} = t^{(2)}(\tilde{g}_i^{(1)})$, and so on. One step in this iterative process may be generalized as:

$$\tilde{g}_i^{(n+1)} = t^{(n+1)}(\tilde{g}_i^{(n)}) \tag{5}$$

where $\tilde{g}_i^{(n+1)}$ is the supermutant, and n is the depth of the mutation path, a reliable proxy for the GED between the anchor and mutant. We draw 5000 random anchors and for each generate n-GED supermutants where $n \in \{1, 2, 3, 4, 5\}$, resulting in 30,000 total compounds. Example of a supermutant set and associated anchor is shown in Figure 4.

GED-EuD Correlation For this task, we are primarily interested in SALSA's performance relative to the Naive counterpart, but also compare against normalized ECFP4 as it provides a reasonable standard for structural awareness. We compute the Spearman correlation coefficient (ρ) between graph edit distance (GED) and Euclidean distance (EuD) in each representation space. To better visualize these trends, we plot the anchor-supermutant EuD distributions for each n-GED in Figure 5. Results show that SALSA substantially improves the GED-EuD correlation ($\rho = 0.868 \pm 0.223$) compared to the Naive baseline ($\rho = 0.560 \pm 0.518$), and is comparable to ECFP4 ($\rho = 0.876 \pm 0.199$). Of note is the wide variation of the Naive encoder, further revealed in Figure 5. The slight bimodal distribution of the Naive encoder may be interpreted as single graph edits inducing changes to SMILES strings that are either small (the left mode) or large (the right mode).

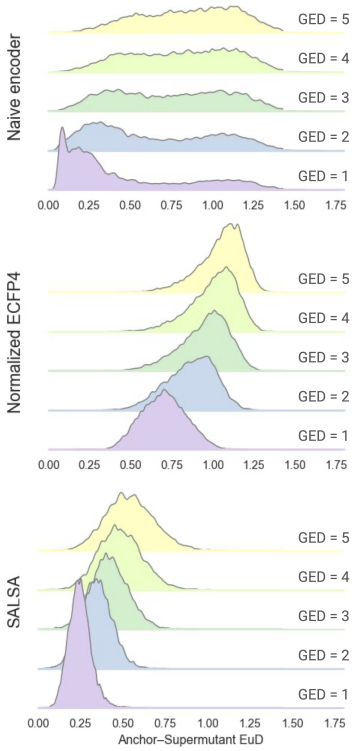

Figure 5: Anchor–supermutant EuDs for SALSA, Naive, and ECFP4, shown per n-GED. Color coding is the same as in Figure 4.

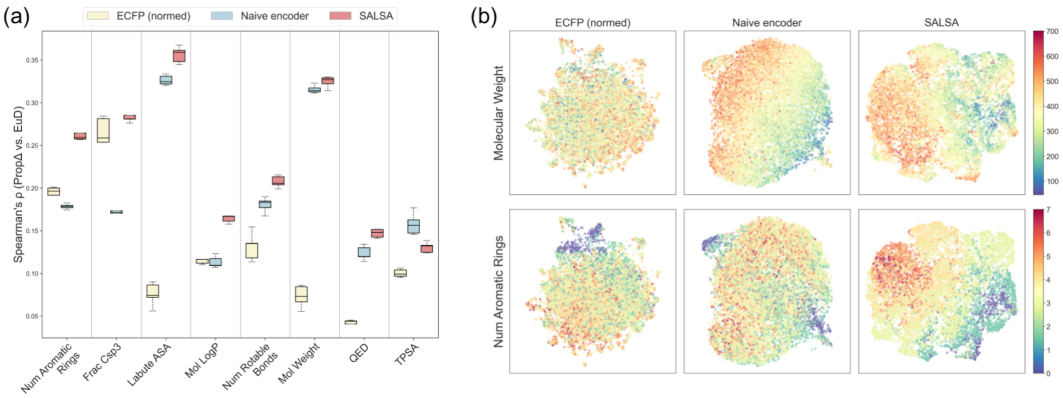

Figure 6: (a) Spearman's ρ correlations between property difference (PropΔ) and Euclidean distance (EuD). We consider eight physicochemical properties and three representations: ECFP4, Naive encoder, and `SALSA`. (b) UMAP embedding of 10,000 random compounds color-coded according to molecular weight and number of aromatic rings for ECFP4, Naive encoder, and `SALSA` representations.

4.2 Physicochemical Property Awareness

In practice, drug discovery campaigns are largely "human-in-the-loop" operations, that, although facilitated by computational methods, still require input from medicinal chemists. For example, medicinal chemists are often asked to identify the most promising compounds from some chemical dataset. This task may amount to visualizing the chemical dataset via some two-dimensional embedding, generally color-coded by some physicochemical property to facilitate more intuitive exploration. Obtaining an embedding effective for this task necessitates that the underlying molecular representation captures information about physicochemical properties.

Property-EuD correlation We investigate the extent to which molecular representations capture information about physicochemical properties by evaluating correlations between property difference (PropΔ) and Euclidean distance (EuD) in representation space; we compare `SALSA`, Naive encoder, and ECFP representations. First, we encode a sample of 1000 molecules into latent representations and obtain all pairwise EuDs. Second, we calculate eight physicochemical properties and compute all pairwise PropΔs. From here, we are able to calculate Spearman's rank correlation coefficient (ρ) between PropΔs and EuDs. Results are shown in Figure 6(a). In addition, we perform UMAP reduction on a sample of 10,000 compounds and color-code the resulting embeddings by molecular weight or number of aromatic rings shown in Figure 6(b). Considering the correlation results, we find that `SALSA` achieves the highest correlation among models for nine out of 10 properties. We note that `SALSA` demonstrated semantic awareness in the structure-based task and maintains semantic awareness in the higher order property-based task. In contrast, ECFP4 demonstrates structural awareness comparable to `SALSA` but does not capture property awareness as effectively, thus, showing less *comprehensive* semantic awareness.

4.3 Biological Property Awareness

We next consider the aspect of biological property awareness, illustrated through performance on a virtual screening benchmark task, to further demonstrate `SALSA`'s capability in the scope of real-world cheminformatics. Virtual screening is a drug discovery task that involves selecting compounds from a candidate pool most likely to be active against a given protein target, given some prescribed notion of molecular similarity. This task essentially assesses the biological property awareness for a given molecular representation, as sufficiently semantically aware representations should result more accurate retrieval of active compounds.

RDKit Virtual Screening Benchmark We utilize the RDKit benchmarking platform [20, 21], which evaluates a model's virtual screening capabilities against 69 protein targets. For each protein target, there is a dataset composed of a small number of "actives" against the protein and a large number of decoy (inactive) compounds. Given a protein target, the objective is to retrieve active compounds from the collective decoy-actives pool given a fixed number ($n = 20$) of query molecules. We compare

Method	Modality	Dimensionality	AUROC
ECFP4 (normed)	Handcrafted	2048	0.62 ± 0.10
RDKit descriptors	Handcrafted	202	0.63 ± 0.03
Hu et al.	Graph	300	0.67 ± 0.10
iMolCLR	Graph	256	0.57 ± 0.09
ChemBERTa	SMILES	768	0.68 ± 0.12
Naive autoencoder	SMILES	32	0.57 ± 0.07
SALSA	**SMILES**	**32**	$\mathbf{0.73 \pm 0.10}$

Table 1: Performance on RDKit VS benchmark. We compare, as usual, against the Naive SMILES encoder and ECFP4. Additionally, we evaluate RDKit descriptors [19] and a variety of deep learning-based methods: Hu et al. [9], iMolCLR [29], and ChemBERTa [5].

SALSA not only to the Naive counterpart and ECFP4, but also to a variety of recent deep learning methods, both SMILES- and graph-based. We show the resulting overall AUROC for each method in Table 1. SALSA demonstrates superior performance relative to ECFP4 and the Naive autoencoder, and is further competitive against the additionally included deep learning-based methods. The results on this biologically-relevant task further indicate SALSA's *comprehensive* semantic awareness.

4.4 Semantic continuity: Molecular Interpolations

We investigate SALSA's ability to generate reasonable molecular interpolations between pairs of endpoint molecules, as higher quality interpolations suggest better semantic continuity in the latent space [23]. To get interpolations, we choose pairs of "endpoint" molecules, calculate the spherical linear interpolation (*slerp*) midpoint [33] between them, and then decode out interpolant molecules from the midpoint code. Figure 7(a) shows a case study of the three most common interpolants for a pair of molecules, for both the SALSA decoder and the Naive decoder. Qualitatively, we can discern that SALSA generates interpolants that are more structurally similar to the endpoints.

We then quantify SALSA's interpolation capability more comprehensively. To this end, we consider five classes of compounds, and for each class, choose a representative set of five molecules. We take all pairwise combinations within each class and determine the most common midpoint interpolants for each pair. Then, to determine "reasonableness" of interpolants, we calculate the Tanimoto distance—a common measure of chemical similarity—between each interpolant and either of their endpoint molecules. Tanimoto distance, d_T, is defined as

$$d_T(b_m, b_e) = \frac{|b_m \cup b_e| - |b_m \cap b_e|}{|b_m \cup b_e|} \in [0, 1] \tag{6}$$

where b_m is the ECFP4 of the midpoint interpolant and b_e is the ECFP of either endpoint molecule. Resulting endpoint-midpoint Tanimoto distances are shown in Figure 7(b). SALSA generates interpolants that, on average, have a lower Tanimoto distance (therefore, are more similar) to their

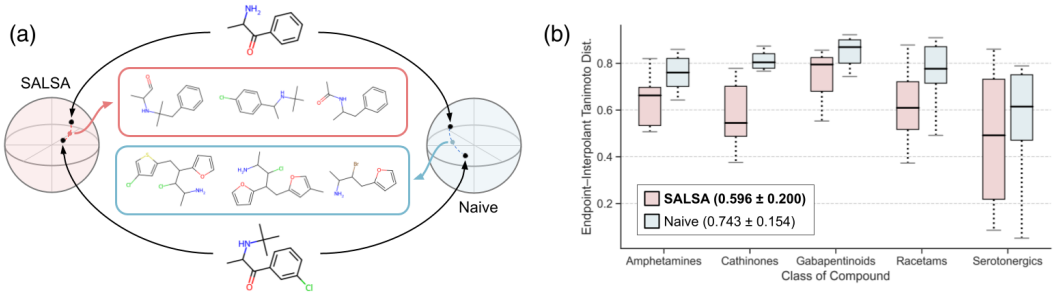

Figure 7: (a) Three most common midpoint interpolants between *cathinone* (top) and *bupropion* (bottom), generated from either SALSA or Naive space. (b) Endpoint-interpolant Tanimoto distances, for each compound class, computed from either SALSA or Naive representations (lower is better).

endpoints. This is indicative of improved semantic continuity in the SALSA space relative to the Naive space, further adding to the *comprehensive* semantic awareness of the SALSA representation.

5 Discussion

Beyond the scope of cheminformatics, we look to provide additional insight as to how SALSA's methodological basis relates to a larger body of deep learning research. An interesting perspective from which to view our work is as a cousin to denoising adversarial autoencoders (DAAEs) [14], particularly as applied to text or sequence data [23]. The goal of the latter work, much like ours, is to coerce a sequence autoencoder to embed related sequences near one another. For the purposes of their DAAE, the natural data space metric is most closely related to a Levenshtein distance. While we seek to respect a different data space metric through SALSA, based on graph similarity, our goals are very much aligned with Shen et al. We opt for an objective function that, although distinct from that of Shen et al., we argue conceptually accomplishes a similar goal, nonetheless, to that of the DAAE objective.

Our dual objective function for SALSA combines a *reconstruction* loss, as well as a *contrastive* loss, which we claim acts similarly to the dual objective of the DAAE, combining a *denoising* technique and an *adversarial* loss. To support this claim, we refer to the work of [28], wherein it was demonstrated that the contrastive loss, when restricted to latent vectors on the unit sphere and given the limit of infinite negative samples, simplifies into two components: an *alignment* loss and a *uniformity* loss. The alignment loss acts to align the latent representation of positive pairs, while the uniformity loss encourages the distribution of all latent vectors to be uniformly distributed on the unit sphere. Each of these losses has a conceptual counterpart in the DAAE, where the alignment loss acts similarly to the denoising objective and the uniformity loss acts like the adversarial component. In presenting this methodological comparison, we hope to provide a more general context for the techniques explored in SALSA, outside applications to molecular modeling.

6 Conclusion

In this work, we proposed SALSA, a framework for learning semantically aware molecular representations. Specifically, we trained a SMILES-autoencoder with a contrastive objective that learned structural similarities between molecules. We showed that our resulting SALSA representations maintained 1) structural awareness, 2) physicochemical property awareness, 3) biological property awareness, and 4) semantic continuity. Collectively, our results demonstrate that SALSA has potential use in a variety of drug discovery tasks.

References

[1] Zaccary Alperstein, Artem Cherkasov, and Jason Tyler Rolfe. *All SMILES Variational Autoencoder*. 2019. arXiv: 1905.13343 [cs.LG].

[2] A Patrícia Bento et al. "The ChEMBL bioactivity database: an update". en. In: *Nucleic Acids Res.* 42.Database issue (Jan. 2014), pp. D1083–90.

[3] Esben Jannik Bjerrum. *SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules*. 2017. arXiv: 1703.07076 [cs.LG].

[4] Samuel R. Bowman et al. "Generating Sentences from a Continuous Space". In: arXiv:1511.06349 (May 2016). arXiv:1511.06349 [cs]. DOI: 10.48550/arXiv.1511.06349. URL: http://arxiv.org/abs/1511.06349.

[5] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. "ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction". In: (Oct. 2020). arXiv: 2010.09885 [cs.LG].

[6] Rafael Gómez-Bombarelli et al. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". In: *ACS Central Science* 4.2 (2018). PMID: 29532027, pp. 268–276. DOI: 10.1021/acscentsci.7b00572. eprint: https://doi.org/10.1021/acscentsci.7b00572. URL: https://doi.org/10.1021/acscentsci.7b00572.

[7] Suhail Haroon, Hafsath C.A., and Jereesh A.S. "Generative Pre-trained Transformer (GPT) based model with relative attention for de novo drug design". In: *Computational Biology and Chemistry* 106 (2023), p. 107911. ISSN: 1476-9271. DOI: https://doi.org/10.1016/j.compbiolchem.2023.107911. URL: https://www.sciencedirect.com/science/article/pii/S1476927123001020.

[8] Shion Honda, Shoi Shi, and Hiroki R. Ueda. *SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery*. 2019. arXiv: 1911.04738 [cs.LG].

[9] Weihua Hu et al. "Strategies for Pre-training Graph Neural Networks". In: (May 2019). arXiv: 1905.12265 [cs.LG].

[10] Junyan Jiang et al. "Transformer VAE: A Hierarchical Model for Structure-Aware and Interpretable Music Representation Learning". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 516–520. DOI: 10.1109/ICASSP40776.2020.9054554.

[11] Mark A Johnson, Gerald M Maggiora, et al. "Concepts and applications of molecular similarity". In: *(No Title)* (1990).

[12] Prannay Khosla et al. "Supervised Contrastive Learning". In: (Apr. 2020). arXiv: 2004.11362 [cs.LG].

[13] Chunyuan Li et al. "Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4678–4699. DOI: 10.18653/v1/2020.emnlp-main.378. URL: https://aclanthology.org/2020.emnlp-main.378.

[14] Alireza Makhzani et al. "Adversarial Autoencoders". In: *CoRR* abs/1511.05644 (2015). arXiv: 1511.05644. URL: http://arxiv.org/abs/1511.05644.

[15] Ivan Montero, Nikolaos Pappas, and Noah A. Smith. "Sentence Bottleneck Autoencoders from Transformer Language Models". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1822–1831. DOI: 10.18653/v1/2021.emnlp-main.137. URL: https://aclanthology.org/2021.emnlp-main.137.

[16] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. "Deep reinforcement learning for de novo drug design". en. In: *Sci Adv* 4.7 (July 2018), eaap7885.

[17] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[18] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[19] RDKit. *RDKit: Open-source cheminformatics*. 2023. URL: https://www.rdkit.org.

[20] Sereina Riniker, Nikolas Fechner, and Gregory A Landrum. "Heterogeneous classifier fusion for ligand-based virtual screening: or, how decision making by committee can be a good thing". en. In: *J. Chem. Inf. Model.* 53.11 (Nov. 2013), pp. 2829–2836.

[21] Sereina Riniker and Gregory A Landrum. "Open-source platform to benchmark fingerprints for ligand-based virtual screening". en. In: *J. Cheminform.* 5.1 (May 2013), p. 26.

[22] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". en. In: *J. Chem. Inf. Model.* 50.5 (May 2010), pp. 742–754.

[23] Tianxiao Shen et al. "Educating text autoencoders: Latent representation guidance via denoising". In: *International conference on machine learning*. PMLR. 2020, pp. 8719–8729.

[24] Aditya Divyakant Shrivastava and Douglas B Kell. "FragNet, a Contrastive Learning-Based Transformer Model for Clustering, Interpreting, Visualizing, and Navigating Chemical Space". en. In: *Molecules* 26.7 (Apr. 2021).

[25] Kihyuk Sohn. "Improved deep metric learning with multi-class N-pair loss objective". In: *Advances in neural information processing systems*. 2016, pp. 1857–1865.

[26] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[27] Ashish Vaswani et al. "Attention is all you need". In: (June 2017). arXiv: 1706.03762 [cs.CL].

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

[29] Yuyang Wang et al. "Improving Molecular Contrastive Learning via Faulty Negative Mitigation and Decomposed Fragment Contrast". en. In: *J. Chem. Inf. Model.* 62.11 (June 2022), pp. 2713–2725.

[30] Yuyang Wang et al. "MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks". In: (Feb. 2021). arXiv: 2102.10056 [cs.LG].

[31] Lai Wei et al. "Probabilistic generative transformer language models for generative design of molecules". In: *Journal of Cheminformatics* 15.1 (Sept. 2023), p. 88. ISSN: 1758-2946. DOI: 10.1186/s13321-023-00759-z.

[32] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *J. Chem. Inf. Comput. Sci.* 28.1 (Feb. 1988), pp. 31–36.

[33] Tom White. "Sampling Generative Networks". In: (Sept. 2016). arXiv: 1609.04468 [cs.NE].

7 Supplementary Material

7.1 Implementation Details

We use $l = 8$ layers for both the encoder and the decoder with a hidden dimension of size $h = 512$, and $m = 8$ heads in the multi-head attention blocks. Our main results are of models trained with $S = 32$ latent dimensions, although we also investigated reduced latent dimensions, $S \in \{16, 8, 4, 2\}$. For the contrastive loss, we set temperature $\tau = 0.7$, following Khosla et al.

7.2 Selected physicochemical properties for Mahalanobis distance

BalabanJ, BertzCT, EState_VSA1, EState_VSA10, EState_VSA2, EState_VSA3, EState_VSA4, EState_VSA5, EState_VSA6, EState_VSA7, EState_VSA8, EState_VSA9, ExactMolWt, FractionCSP3, HallKierAlpha, HeavyAtomCount, Ipc, Kappa1, Kappa2, Kappa3, LabuteASA, MaxAbsEStateIndex, MaxEStateIndex, MinEStateIndex, MolLogP, MolMR, MolWt NumAliphaticCarbocycles, NumAliphaticHeterocycles, NumAliphaticRings, NumAromaticCarbocycles, NumAromaticHeterocycles, NumAromaticRings, NumHAcceptors, NumHDonors, NumHeteroatoms, NumRotatableBonds, NumSaturatedCarbocycles, NumSaturatedHeterocycles, NumSaturatedRings, NumValenceElectrons, PEOE_VSA1, PEOE_VSA10, PEOE_VSA11, PEOE_VSA12, PEOE_VSA13, PEOE_VSA14, PEOE_VSA2, PEOE_VSA3, PEOE_VSA4, PEOE_VSA5, PEOE_VSA6, PEOE_VSA7, PEOE_VSA8, PEOE_VSA9, SMR_VSA1, SMR_VSA10, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA5, SMR_VSA6, SMR_VSA7, SMR_VSA9, SlogP_VSA1, SlogP_VSA10, SlogP_VSA11, SlogP_VSA12, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA6, SlogP_VSA7, SlogP_VSA8, TPSA, VSA_EState1, VSA_EState10, VSA_EState2, VSA_EState3, VSA_EState4, VSA_EState5, VSA_EState6, VSA_EState7, VSA_EState8, VSA_EState9

7.3 Additional Figures

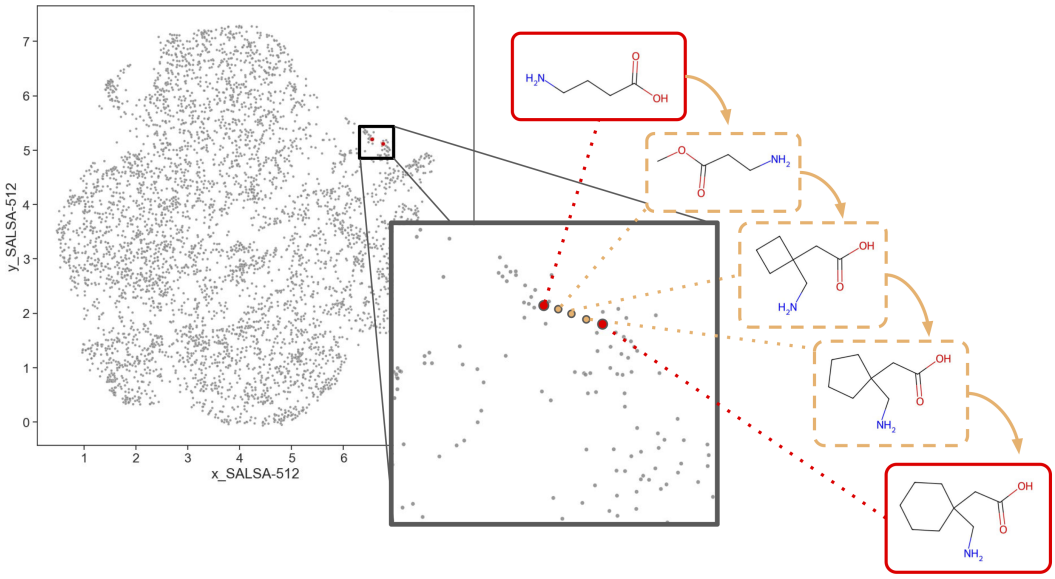

Figure 8: Example of incremental step-wise interpolations in SALSA latent space.