Eurographics Conference on Visualization (EuroVis) 2019
M. Gleicher, H. Leitte, and I. Viola
(Guest Editors)

Volume 38 (2019), Number 3

Supplemental Material

1. Literature Survey

As we aimed to draw from a diverse set of research communities, we started the search on arXiv, an interdisciplinary digital archive for preprints in computer science as well as other scientific fields. While our primary goal was to find publications pertinent to latent representations, we noticed that the term is not standard terminology. Rather, researchers referred to it as "manifold", "latent code", "embedding", or simply "z". Thus, we additionally used keywords corresponding to algorithms that produced latent representations, for instance "variational autoencoder". Because of the sheer amount of relevant articles, we only included papers published within the recent 5 years and randomly sampled a subset from the search results. We manually filtered the search results to only include articles published on peer-reviewed venues. We also excluded papers that did not inspect or evaluate the latent representations. After the initial list of publications, we followed backward and forward citation links to obtain additional relevant resources. The additional articles obtained by citation links were not randomly sampled.

With this procedure, we arrived at 78 papers from a variety of publication venues, including 44 papers from ML related conferences and journals, 27 from NLP, and 7 from scientific fields such as Physics and Biology.

1.1. Paper List

In this section, we list all the articles we decided to include in the literature survey. We divide the articles into three categories depending on their publication venues.

Papers published in Machine Learning, Artificial Intelligence, or Computer Vision related venues: [HSSQ17] [PGH*16] [KMRW14] [KWKT15] [ADvdH17] [DTSB15] [DSTB17] [JZS17] [YYSL16] [SGZ*16] [BGS16] [DCF*15] [CKD*15] [GDG*15] [BTN18] [CDH*16] [WG16] [SPT*17] [AFDM17] [LSL*16] [CKS*17] [MZZ*16] [HMP*17] [ZSE17] [CDP*18] [TGLX18] [HNP17] [PWH*17] [KWM*18] [JBJ18] [DTD*18] [YKDFF17] [SRM*16] [NS17] [MSSW16] [WDGH16] [MNG17] [LTWE*17] [YHSBK17] [KPHL17] [RLM*17] [BMK15] [BM17] [BCZ*16]

Papers published in Natural Language Processing or Information Retrieval related venues: [ZZE17] [HZG17] [HG18] [NCMW18] [SSB17] [BVV*16] [CGB*15] [ALL*16] [NSPM14] [JYY*16] [UFDR16] [AM16] [NSV16] [HLJ16] [CR16] [KBdR16] [YS16] [CKP16] [NAM16] [YSD*18] [GA17] [LK17] [ZC17] [NNM*17] [SLJ*17] [Str17] [PGE17]

Papers published in Physics, Astronomy, Biology, and general natural science venues: [RBWT18] [Wet17] [RGS*18] [FSBL17] [WG18] [NBPvdW18] [GBWD*18]

Note that there might be discrepancies between the topic of an article and its publication venue. For example, an article focus on molecular graph, but it appears in the proceedings of an ML conference. In this case, we count it as an ML paper instead of a natural science paper.

1.2. Analysis

We analyzed the surveyed papers using an iterative coding method. The first author coded all data. Throughout the coding process, we iteratively revised our codes according to evolved understanding. The goals and tasks in using and interpreting latent spaces, presented in Section 4 of the main text, are results from this analysis. Figure 1 summarizes the assignment of codes.

2. System Details

In this section, we include concrete math steps for computing attribute vector projection and rendering multiple attribute vectors in a global view. We then briefly describe the visual encodings and available interactions.

2.1. Attribute Vector Projection

Let $\mathbf{X} \in \mathbb{R}^{n \times l}$ denote the data matrix where each row represents a latent vector of a data sample (*e.g.,* an emoji image). To define an attribute vector, the user specifies a starting concept $\{\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, ..., \boldsymbol{x}_m^s\}$ and an ending concept $\{\boldsymbol{x}_1^e, \boldsymbol{x}_2^e, ..., \boldsymbol{x}_k^e\}$, where $\boldsymbol{x}_i$ is a row in $\mathbf{X}$. The attribute vector $\boldsymbol{v}$ is:

$$\boldsymbol{v} = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{x}_i^e - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i^s \quad (1)$$

In the attribute vector projection, the x-axis is the direction of $\boldsymbol{v}$. The y-axis is the first principal component of the remaining dimensions, specifically:

$$\hat{\mathbf{X}} = (\mathbf{X} - \bar{\boldsymbol{x}})\boldsymbol{e}_v^T \boldsymbol{e}_v \quad (2)$$

where $\boldsymbol{e}_v$ is the unit vector of $\boldsymbol{v}$. The first principal component $\boldsymbol{e}_y$ of $\hat{\mathbf{X}}$ is the direction of y-axis.

Now we form an orthogonal basis $\mathbf{W} = \begin{bmatrix} e_v \\ e_y \end{bmatrix}$ for the attribute vector projection. Projecting $\mathbf{X}$ to 2 dimensions is simply $\mathbf{T} = \mathbf{XW}, \mathbf{T} \in \mathbb{R}^{n \times 2}, \mathbf{X} \in \mathbb{R}^{n \times l}, \mathbf{W} \in \mathbb{R}^{l \times 2}$.

		[HSSQ17]	[PGH*16]	[KMRW14]	[KWKT15]	[ADvdH17]	[DTSB15]	[DSTB17]	[JZS17]	[YYSL16]	[SGZ*16]	[BGS16]	[DCF*15]	[CKD*15]	[GDG*15]	[BTN18]	[CDH*16]	[WG16]	[SPT*17]	[AFDM17]	[LSL*16]	[CKS*17]	[MZZ*16]	[HMP*17]	[ZSE17]	[YKDFF17]	[CDP*18]	[TGLX18]	[HNP17]	[PWH*17]	[KWM*18]	[JBJ18]	[DTD*18]	[SRM*16]	[NS17]	[MSSW16]	[WDGH16]	[MNG17]	[LTWE*17]	[YHSBK17]
Venue	ML (M) / NLP (N) / Science (S)	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
Model Type	Generative Models	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Word Embeddings																																							
Use Goals	Improve Downstream Tasks			✓		✓					✓									✓	✓	✓														✓	✓		✓	✓
	Enable Synthesis						✓	✓	✓	✓					✓				✓									✓	✓			✓							✓	✓
	Understand Data																																							
	Unspecified	✓	✓		✓							✓	✓	✓		✓	✓	✓					✓	✓	✓	✓	✓			✓	✓		✓	✓	✓			✓		
Interpretation Goals	Evaluate model	✓	✓		✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓			✓		✓
	Explain model	✓				✓	✓	✓	✓	✓		✓	✓	✓						✓	✓	✓	✓	✓	✓			✓			✓			✓	✓	✓	✓		✓	✓
	Understand data			✓																																				
Tasks	View Reconstruction Examples	✓	✓			✓	✓		✓	✓	✓		✓	✓	✓	✓		✓	✓			✓			✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓		✓
	View Interpolation Results	✓		✓	✓		✓	✓		✓						✓	✓	✓					✓	✓				✓					✓				✓			
	Examine Nearest Neighbors									✓					✓			✓					✓					✓												
	Perform Attribute Vector Arithmetic	✓						✓																																
	Compare Similarities																																							
	Visualize Distribution	✓				✓														✓	✓				✓			✓	✓			✓	✓		✓			✓	✓	✓

		[KPHL17]	[RLM*17]	[RBWT18]	[RGS*18]	[Wet17]	[FSBL17]	[WG18]	[NBPvdW18]	[GBWD*18]	[BVV*16]	[SSB17]	[HG18]	[NCMW18]	[HZG17]	[ZZE17]	[CGB*15]	[ALL*16]	[NSPM14]	[JYY*16]	[UFDR16]	[AM16]	[NSV16]	[HLJ16]	[CR16]	[KBdR16]	[YS16]	[CKP16]	[NAM16]	[YSD*18]	[GA17]	[LK17]	[ZC17]	[NNM*17]	[SLJ*17]	[Str17]	[PGE17]	[BMK15]	[BCZ*16]	[BM17]
Venue	ML (M) / NLP (N) / Science (S)	M	M	S	S	S	S	S	S	S	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	M	M	M
Model Type	Generative Models	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓																								
	Word Embeddings																✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use Goals	Improve Downstream Tasks			✓	✓				✓					✓									✓		✓	✓	✓	✓	✓				✓				✓			
	Enable Synthesis		✓							✓	✓	✓				✓																								
	Understand Data				✓	✓	✓	✓														✓		✓						✓									✓	✓
	Unspecified	✓											✓		✓		✓	✓	✓	✓	✓										✓	✓		✓	✓	✓		✓		
Interpretation Goals	Evaluate model	✓	✓				✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
	Explain model								✓	✓				✓		✓	✓	✓			✓				✓	✓				✓		✓	✓		✓	✓	✓	✓		✓
	Understand data					✓		✓														✓		✓						✓									✓	✓
Tasks	View Reconstruction Examples		✓								✓	✓				✓																								
	View Interpolation Results	✓						✓		✓	✓																													
	Examine Nearest Neighbors									✓									✓					✓	✓	✓				✓	✓				✓	✓	✓			✓
	Perform Attribute Vector Arithmetic							✓									✓	✓	✓	✓					✓							✓		✓		✓		✓	✓	
	Compare Similarities																✓		✓	✓	✓		✓		✓			✓			✓	✓		✓	✓	✓	✓			
	Visualize Distribution	✓				✓	✓	✓		✓			✓			✓				✓	✓			✓						✓					✓				✓	✓

Figure 1: A table of reviewed papers and their corresponding goals and tasks in using and interpreting latent spaces.

2.2. Attribute Vectors in a Global View

We support visualizing the attribute vectors globally in any available projection, including t-SNE, UMAP, PCA, and the projection of an attribute vector. Let $\boldsymbol{v}_0 = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i^s$ denote the starting point of $\boldsymbol{v}$ and $\boldsymbol{v}_1 = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{x}_i^e$ the ending point. We are going to project $\mathbf{V} = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$.

For a linear projection such as PCA or our custom attribute vector projection, mapping is straightforward because each attribute vector appear as a line segment. Our goal is to map $\mathbf{V} \in \mathbb{R}^{2\times l}$ to $\mathbf{T} \in \mathbb{R}^{2\times 2}$ and plot a line between $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$. For PCA, we use $\mathbf{T} = \mathbf{VW}$, where $\mathbf{W}$ is the first two eigenvectors. For our custom attribute vector projection, we use $\mathbf{T} = \mathbf{VW}$, where $\mathbf{W}$ is the orthogonal basis described in § 2.1.

For nonlinear projections, the projected attribute vector is no longer a straight line, so we approximate the path using a spline. We first sample a set of control points along the attribute vector $\boldsymbol{v}$:

$$\boldsymbol{c}_{(t)} = \boldsymbol{v}_0 + t\boldsymbol{v}, 0 \leq t \leq 1 \tag{3}$$

We then map the control points $\boldsymbol{c}_{(t)}$ to the corresponding positions $\boldsymbol{c}'_{(t)}$ in the 2D embedding. Finally, we render a Catmull-Rom spline using $\boldsymbol{c}'_{(t)}$ as control points.

UMAP provides native support for the mapping of $\boldsymbol{c}_{(t)}$ to $\boldsymbol{c}'_{(t)}$, but t-SNE does not. We further approximate $\boldsymbol{c}'_{(t)}$ for t-SNE using the following steps. We first find the k nearest neighbor of $\boldsymbol{c}_{(t)}$, where the Euclidean distance (cosine distance for word embeddings) between i-th nearest neighbor and $\boldsymbol{c}_{(t)}$ is w_{ti}. We then map the nearest neighbor to the t-SNE 2D embedding, so $\boldsymbol{n}_{(ti)}$ is the 2D coordinate of the i-th nearest neighbor. Finally, we compute

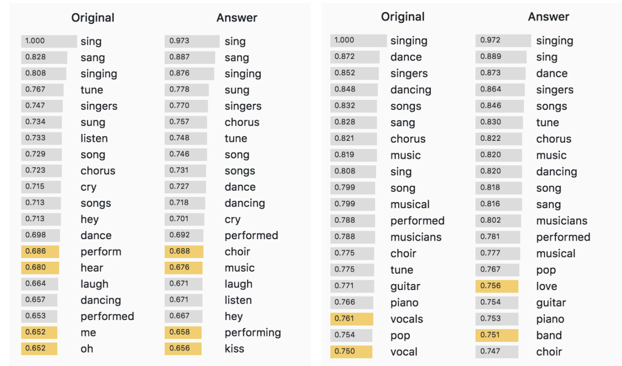
$$\boldsymbol{c}'_{(t)} = \frac{\sum_{i=1}^{k}\frac{1}{w_{ti}}\boldsymbol{n}_{(ti)}}{\sum_{i=1}^{k}\frac{1}{w_{ti}}} \tag{4}$$

2.3. Visual Encodings and Interactions

The summary page lists quantitative metrics, and visualizes the distribution on initial latent dimensions for each latent space of generative models (main paper, Figure 1a). In the visualization of initial latent dimensions, the y-axis is a categorical axis, where we stack each initial latent dimension in order. The x-axis indicates the raw values on the latent dimensions. We use a line to represent the distribution on a latent dimension, where the dashed portion goes from the minimum value to the maximum value of the dimension, and the thicker portion represents the inter-quartile range. When showing user-defined metrics (Figure 2), we list the quality scores for all attribute vectors as raw numbers in a table. We additionally color each number using a single-hue sequential color scale (*blue*).

Both the vector space overview (main paper, Figure 3) and the custom projection on attribute vector (main paper, Figure 4) use

Figure 2: Cross-model assessments. We supply quality measures of all attribute vectors in each latent space.



Original		Answer	
1.000	sing	0.973	sing
0.828	sang	0.887	sang
0.808	singing	0.876	singing
0.767	tune	0.778	sung
0.747	singers	0.770	singers
0.734	sung	0.757	chorus
0.733	listen	0.748	tune
0.729	song	0.746	song
0.723	chorus	0.731	songs
0.715	cry	0.727	dance
0.713	songs	0.718	dancing
0.713	hey	0.701	cry
0.698	dance	0.692	performed
0.686	perform	0.688	choir
0.680	hear	0.676	music
0.664	laugh	0.671	laugh
0.657	dancing	0.671	listen
0.653	performed	0.667	hey
0.652	me	0.658	performing
0.652	oh	0.656	kiss

Original		Answer	
1.000	singing	0.972	singing
0.872	dance	0.889	sing
0.852	singers	0.873	dance
0.848	dancing	0.864	singers
0.832	songs	0.846	songs
0.828	sang	0.830	tune
0.821	chorus	0.822	chorus
0.819	music	0.820	music
0.808	sing	0.820	dancing
0.799	song	0.818	song
0.799	musical	0.816	sang
0.788	performed	0.802	musicians
0.788	musicians	0.781	performed
0.775	choir	0.777	musical
0.775	tune	0.767	pop
0.771	guitar	0.756	love
0.766	piano	0.754	guitar
0.761	vocals	0.753	piano
0.754	pop	0.751	band
0.750	vocal	0.747	choir

(a) (b)

Figure 3: (a) Nearest neighbor words of *sing* and those after adding the *present:participle* attribute vector to *sing*. (b) Nearest neighbor words of *singing* and those after subtracting the *present:participle* attribute vector to *singing*.

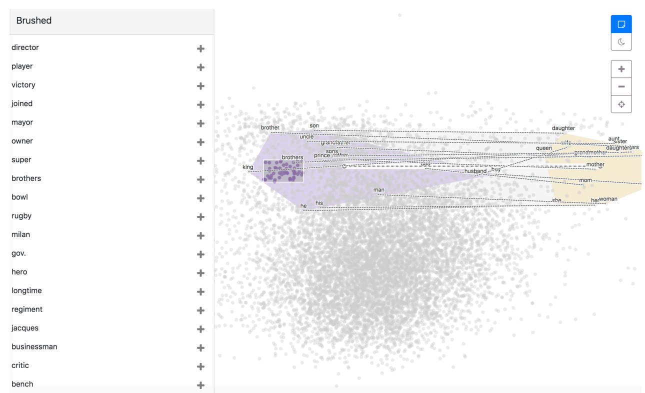


Figure 4: Gender bias in word embeddings. Words are projected onto an attribute vector containing gendered names. Brushing the region around the male concept convex hull reveals words that reflect gender stereotypes.

a scatter plot to visualize data distribution. In the overview page, the x- and y-axis corresponds to the two dimensions resulting from the t-SNE, UMAP or PCA projection. In the custom projection, the x-axis is the attribute vector direction and the y-axis is the first principal component of the remaining dimensions, as described in §2.1. Users can change the y-axis to other principal component, or a categorical axis (main paper, Figure 11). In these scatter plots, each data point is represented by a dot, and the color is configurable. For example, the default color is the mean pixel color of an emoji image, but user can change the color to encode emoji platform or version. When the total number of data points is small enough, we directly render the image or text (main paper, Figure 4).

We offer a variety of interactions to make exploration smooth and fluid, including brush, search, filter, zoom, and pan.

We precompute t-SNE projections using several perplexity parameters (5, 10, 30, 50, 100, with 30 being the default value) and allow users to change perplexity in a dropdown menu. Similarly, we allow users to adjust two parameters in generating UMAP projections, including *Neighbor*, which determines the number of neighboring points used in local approximation, and *Distance*, which controls how tightly the embedding compresses points together.

When user asks for the details of a data point, in addition to showing all available meta data (main paper, Figure 1c), we tailor specific information for each data type we support. For abstract vector data where each input datum is a high-dimensional numeric array, we visualize all input dimensions as a heatmap (main paper, Figure 11). Specifically, each cell in the heatmap corresponds to a number in the input array and the color represents the scalar value using a perceptually uniform color scale (*inferno*). From the heatmap, users might gain exploratory understandings on the input data, for example identifying a data quality issue where all input dimensions are zero. For word embeddings, because of the importance of neighborhood information, we list the top-k nearest neighbors in the original embedding space.

3. Case Study Details

In this section, we attach two figures to support claims in the word embedding example, and provide more details on the analysis of cancer transcriptomes.

3.1. Case Study: Word Embeddings

Figure 3 and Figure 4 support the findings on word embeddings in the main text.

3.2. Case Study: Cancer Transcriptomes

We conducted an analysis on high grade serous ovarian cancer subtypes involving vector arithmetic. Here are our steps, using the *immunoreactive* and *mesenchymal* pair as an example:

1. We first compute the centroid of each subtype in z, namely $\overline{\theta}_{immuno}$ and $\overline{\theta}_{mes}$.
2. We run both $\overline{\theta}_{immuno}$ and $\overline{\theta}_{mes}$ through the decoder to reconstruct

the corresponding gene profiles X_{immuno} and X_{mes}. We subtract to get the signed difference $X_{mes-immuno} = X_{mes} - X_{immuno}$.

3. Following [WG18], we next look to threshold for genes 2.5 standard deviations away in both positive and negative tails. However, the distribution is not normal and so we use quantile-based thresholding. We first find the quantiles corresponding to 2.5 standard deviations in a standard normal distribution (0.06% and 99.4%, respectively). We then take the corresponding top and bottom quantiles of $X_{mes-immuno}$ to form the gene list. The genes in the positive tail are associated with the mesenchymal subtype, while the genes in the negative tail are associated with immunoreactive.

The major differences of our approach from [WG18] are:

1. We do not project the vector back to a single latent encoding dimension;
2. We apply the non-linear decoder function instead of directly inspecting decoder weights;
3. We use quantile-based thresholding.

Ours	Theirs	#B	#O	#T	Agreement
Mesenchymal	87 +	19	32	56	59%
Mesenchymal	56 -	1	32	90	3%
Immunoreactive	56 +	15	32	58	46%
Immunoreactive	87 -	0	32	74	0%
Proliferative	79 +	9	32	60	28%
Proliferative	38 -	8	32	134	25%
Differentiated	38 +	1	32	15	3%
Differentiated	79 -	0	32	39	0%

Table 1: Comparison of the overlap between our and their gene lists. #B: the number of overlapping genes in both list. #O: the total number of genes in our list. #T: the total number of genes in their list.

We compare the overlap between our resulting gene list and that of [WG18], as shown in Table 1. The agreement is notably poor, including multiple cases with a null intersection.

References

[ADvdH17] Abbasnejad M. E., Dick A., van den Hengel A.: Infinite variational autoencoder for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 781–790. 1

[AFDM17] Alemi A. A., Fischer I., Dillon J. V., Murphy K.: Deep variational information bottleneck. In *Proceedings of the International Conference on Learning Representations* (2017). 1

[ALL*16] Arora S., Li Y., Liang Y., Ma T., Risteski A.: A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics 4* (2016), 385–399. 1

[AM16] Asgari E., Mofrad M. R.: Comparing fifty natural languages and twelve genetic languages using word embedding language divergence (weld) as a quantitative measure of language distance. In *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP* (2016), pp. 65–74. 1

[BCZ*16] Bolukbasi T., Chang K.-W., Zou J. Y., Saligrama V., Kalai A. T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems* (2016), pp. 4349–4357. 1

[BGS16] Burda Y., Grosse R. B., Salakhutdinov R.: Importance weighted autoencoders. In *Proceedings of the International Conference on Learning Representations* (2016). 1

[BM17] Bamler R., Mandt S.: Dynamic word embeddings. In *Proceedings of the International Conference on Machine Learning* (2017), pp. 380–389. 1

[BMK15] Bollegala D., Maehara T., Kawarabayashi K.: Embedding semantic relations into word representations. In *Proceedings of the International Joint Conference on Artificial Intelligence* (2015), pp. 1222–1228. 1

[BTN18] Bouchacourt D., Tomioka R., Nowozin S.: Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), pp. 2095–2102. 1

[BVV*16] Bowman S. R., Vilnis L., Vinyals O., Dai A. M., Józefowicz R., Bengio S.: Generating sentences from a continuous space. In *Proceedings of the Conference on Computational Natural Language Learning* (2016), pp. 10–21. 1

[CDH*16] Chen X., Duan Y., Houthooft R., Schulman J., Sutskever I., Abbeel P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems* (2016), pp. 2172–2180. 1

[CDP*18] Chen L., Dai S., Pu Y., Zhou E., Li C., Su Q., Chen C., Carin L.: Symmetric variational autoencoder and connections to adversarial learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (09–11 Apr 2018), vol. 84, PMLR, pp. 661–669. 1

[CGB*15] Cui Q., Gao B., Bian J., Qiu S., Dai H., Liu T.: Knet: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems 34*, 1 (2015), 4. 1

[CKD*15] Chung J., Kastner K., Dinh L., Goel K., Courville A. C., Bengio Y.: A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems* (2015), pp. 2980–2988. 1

[CKP16] Chiu B., Korhonen A., Pyysalo S.: Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, RepEval@ACL* (2016), pp. 1–6. 1

[CKS*17] Chen X., Kingma D. P., Salimans T., Duan Y., Dhariwal P., Schulman J., Sutskever I., Abbeel P.: Variational lossy autoencoder. In *Proceedings of the International Conference on Learning Representations* (2017). 1

[CR16] Cao K., Rei M.: A joint model for word embedding and word morphology. In *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL* (2016), pp. 18–26. 1

[DCF*15] Denton E. L., Chintala S., Fergus R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems* (2015), pp. 1486–1494. 1

[DSTB17] Dosovitskiy A., Springenberg J. T., Tatarchenko M., Brox T.: Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence 39*, 4 (2017), 692–705. 1

[DTD*18] Dai H., Tian Y., Dai B., Skiena S., Song L.: Syntax-directed variational autoencoder for structured data. In *Proceedings of the International Conference on Learning Representations* (2018). 1

[DTSB15] Dosovitskiy A., Tobias Springenberg J., Brox T.: Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1538–1546. 1

[FSBL17] Frontera-Pons J., Sureau F., Bobin J., Le Floc'h E.: Unsupervised feature-learning for galaxy seds with denoising autoencoders. *Astronomy & Astrophysics 603* (Jul 2017), A60. 1

[GA17] Goldberg Y., Avraham O.: The interplay of semantics and morphology in word embeddings. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics* (2017), pp. 422–426. 1

[GBWD*18] Gómez-Bombarelli R., Wei J. N., Duvenaud D., Hernández-Lobato J. M., Sánchez-Lengeling B., Sheberla D., Aguilera-Iparraguirre J., Hirzel T. D., Adams R. P., Aspuru-Guzik A.: Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science 4*, 2 (2018), 268–276. 1

[GDG*15] Gregor K., Danihelka I., Graves A., Rezende D., Wierstra D.: Draw: A recurrent neural network for image generation. In *Proceedings of the International Conference on Machine Learning* (2015), pp. 1462–1471. 1

[HG18] Hsu W., Glass J. R.: Scalable factorized hierarchical variational autoencoder training. In *Proceedings of the Annual Conference of the International Speech Communication Association* (2018), pp. 1462–1466. 1

[HLJ16] Hamilton W. L., Leskovec J., Jurafsky D.: Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2016). 1

[HMP*17] Higgins I., Matthey L., Pal A., Burgess C., Glorot X., Botvinick M., Mohamed S., Lerchner A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations* (2017). 1

[HNP17] Hadjeres G., Nielsen F., Pachet F.: Glsr-vae: Geodesic latent space regularization for variational autoencoder architectures. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (2017), IEEE, pp. 1–7. 1

[HSSQ17] Hou X., Shen L., Sun K., Qiu G.: Deep feature consistent variational autoencoder. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision* (2017), pp. 1133–1141. 1

[HZG17] Hsu W.-N., Zhang Y., Glass J.: Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *Automatic Speech Recognition and Understanding Workshop* (2017), IEEE, pp. 16–23. 1

[JBJ18] Jin W., Barzilay R., Jaakkola T.: Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the International Conference on Machine Learning* (10–15 Jul 2018), vol. 80, pp. 2323–2332. 1

[JYY*16] Ji S., Yun H., Yanardag P., Matsushima S., Vishwanathan S. V. N.: Wordrank: Learning word embeddings via robust ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2016), pp. 658–668. 1

[JZS17] Jain U., Zhang Z., Schwing A. G.: Creativity: Generating diverse questions using variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5415–5424. 1

[KBdR16] Kenter T., Borisov A., de Rijke M.: Siamese CBOW: optimizing word embeddings for sentence representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2016). 1

[KMRW14] Kingma D. P., Mohamed S., Rezende D. J., Welling M.: Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems* (2014), pp. 3581–3589. 1

[KPHL17] Kusner M. J., Paige B., Hernández-Lobato J. M.: Grammar variational autoencoder. In *Proceedings of the International Conference on Machine Learning* (06–11 Aug 2017), vol. 70, pp. 1945–1954. 1

[KWKT15] Kulkarni T. D., Whitney W. F., Kohli P., Tenenbaum J.: Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems* (2015), pp. 2539–2547. 1

[KWM*18] Kim Y., Wiseman S., Miller A. C., Sontag D., Rush A. M.: Semi-amortized variational autoencoders. In *Proceedings of the International Conference on Machine Learning* (2018), pp. 2683–2692. 1

[LK17] Lison P., Kutuzov A.: Redefining context windows for word embedding models: An experimental study. In *Proceedings of the Nordic Conference on Computational Linguistics* (2017), pp. 284–288. 1

[LSL*16] Louizos C., Swersky K., Li Y., Welling M., Zemel R.: The variational fair autoencoder. In *Proceedings of the International Conference on Learning Representations* (2016). 1

[LTWE*17] Linh Tran D., Walecki R., Eleftheriadis S., Schuller B., Pantic M., et al.: Deepcoder: Semi-parametric variational autoencoders for automatic facial action coding. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 3190–3199. 1

[MNG17] Mescheder L., Nowozin S., Geiger A.: Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning* (06–11 Aug 2017), vol. 70, pp. 2391–2400. 1

[MSSW16] Maaløe L., Sønderby C. K., Sønderby S. K., Winther O.: Auxiliary deep generative models. In *Proceedings of the International Conference on Machine Learning* (2016), vol. 48, pp. 1445–1454. 1

[MZZ*16] Mathieu M. F., Zhao J. J., Zhao J., Ramesh A., Sprechmann P., LeCun Y.: Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems* (2016), pp. 5040–5048. 1

[NAM16] Nayak N., Angeli G., Manning C. D.: Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, RepEval@ACL* (2016), pp. 19–23. 1

[NBPvdW18] Naul B., Bloom J. S., Pérez F., van der Walt S.: A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy 2*, 2 (2018), 151. 1

[NCMW18] Naradowsky J., Cotterell R., Mielke S. J., Wolf-Sonkin L.: A structured variational autoencoder for contextual morphological inflection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2018), pp. 2631–2640. 1

[NNM*17] Nguyen D. Q., Nguyen D. Q., Modi A., Thater S., Pinkal M.: A mixture model for learning multi-sense word embeddings. In *Proceedings of the Joint Conference on Lexical and Computational Semantics* (2017), pp. 121–127. 1

[NS17] Nalisnick E., Smyth P.: Stick-breaking variational autoencoders. In *Proceedings of the International Conference on Learning Representations* (2017). 1

[NSPM14] Neelakantan A., Shankar J., Passos A., McCallum A.: Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2014), pp. 1059–1069. 1

[NSV16] Nguyen K. A., Schulte im Walde S., Vu N. T.: Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2016). 1

[PGE17] Pinter Y., Guthrie R., Eisenstein J.: Mimicking word embeddings using subword rnns. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2017), pp. 102–112. 1

[PGH*16] Pu Y., Gan Z., Henao R., Yuan X., Li C., Stevens A., Carin L.: Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems* (2016), pp. 2352–2360. 1

[PWH*17] Pu Y., Wang W., Henao R., Chen L., Gan Z., Li C., Carin L.: Adversarial symmetric variational autoencoder. In *Advances in Neural Information Processing Systems* (2017), pp. 4330–4339. 1

[RBWT18] RIBEIRO J. M. L., BRAVO P., WANG Y., TIWARY P.: Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *The Journal of Chemical Physics 149*, 7 (2018), 072301. 1

[RGS*18] ROCCHETTO A., GRANT E., STRELCHUK S., CARLEO G., SEVERINI S.: Learning hard quantum distributions with variational autoencoders. *npj Quantum Information 4*, 1 (2018), 28. 1

[RLM*17] RAVANBAKHSH S., LANUSSE F., MANDELBAUM R., SCHNEIDER J. G., PÓCZOS B.: Enabling dark energy science with deep generative models of galaxy images. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2017), pp. 1488–1494. 1

[SGZ*16] SALIMANS T., GOODFELLOW I., ZAREMBA W., CHEUNG V., RADFORD A., CHEN X.: Improved techniques for training gans. In *Advances in Neural Information Processing Systems* (2016), pp. 2234–2242. 1

[SLJ*17] SHI B., LAM W., JAMEEL S., SCHOCKAERT S., LAI K. P.: Jointly learning word embeddings and latent topics. In *Proceedings of the International Conference on Research and Development in Information Retrieval* (2017), pp. 375–384. 1

[SPT*17] SHRIVASTAVA A., PFISTER T., TUZEL O., SUSSKIND J., WANG W., WEBB R.: Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), vol. 2, p. 5. 1

[SRM*16] SØNDERBY C. K., RAIKO T., MAALØE L., SØNDERBY S. K., WINTHER O.: Ladder variational autoencoders. In *Advances in Neural Information Processing Systems* (2016), pp. 3738–3746. 1

[SSB17] SEMENIUTA S., SEVERYN A., BARTH E.: A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2017), pp. 627–637. 1

[Str17] STRATOS K.: Reconstruction of word embeddings from subword parameters. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP* (2017), pp. 130–135. 1

[TGLX18] TAN Q., GAO L., LAI Y., XIA S.: Variational autoencoders for deforming 3d mesh models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 5841–5850. 1

[UFDR16] UPADHYAY S., FARUQUI M., DYER C., ROTH D.: Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2016). 1

[WDGH16] WALKER J., DOERSCH C., GUPTA A., HEBERT M.: An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision* (2016), Springer, pp. 835–851. 1

[Wet17] WETZEL S. J.: Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Physical Review E 96*, 2 (2017), 022140. 1

[WG16] WANG X., GUPTA A.: Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision* (2016), Springer, pp. 318–335. 1

[WG18] WAY G. P., GREENE C. S.: Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. In *Proceedings of Pacific Symposium on Biocomputing* (2018), vol. 23, pp. 80–91. 1, 4

[YHSBK17] YANG Z., HU Z., SALAKHUTDINOV R., BERG-KIRKPATRICK T.: Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the International Conference on Machine Learning* (2017), vol. 70, pp. 3881–3890. 1

[YKDFF17] YEUNG S., KANNAN A., DAUPHIN Y., FEI-FEI L.: Tackling over-pruning in variational autoencoders. *ICML 2017 Workshop on Principled Approaches to Deep Learning* (2017). 1

[YS16] YAGHOOBZADEH Y., SCHÜTZE H.: Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2016). 1

[YSD*18] YAO Z., SUN Y., DING W., RAO N., XIONG H.: Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the International Conference on Web Search and Data Mining* (2018), pp. 673–681. 1

[YYSL16] YAN X., YANG J., SOHN K., LEE H.: Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision* (2016), Springer, pp. 776–791. 1

[ZC17] ZAMANI H., CROFT W. B.: Relevance-based word embedding. In *Proceedings of the International Conference on Research and Development in Information Retrieval* (2017), pp. 505–514. 1

[ZSE17] ZHAO S., SONG J., ERMON S.: Learning hierarchical features from deep generative models. In *Proceedings of the International Conference on Machine Learning* (2017), pp. 4091–4099. 1

[ZZE17] ZHAO T., ZHAO R., ESKENAZI M.: Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the Association for Computational Linguistics* (2017), vol. 1, pp. 654–664. 1