DE-VAE: Revealing Uncertainty in Parametric and Inverse Projections with Variational Autoencoders using Differential Entropy

Frederik L. Dennig[*]
University of Konstanz

Daniel A. Keim[†]
University of Konstanz

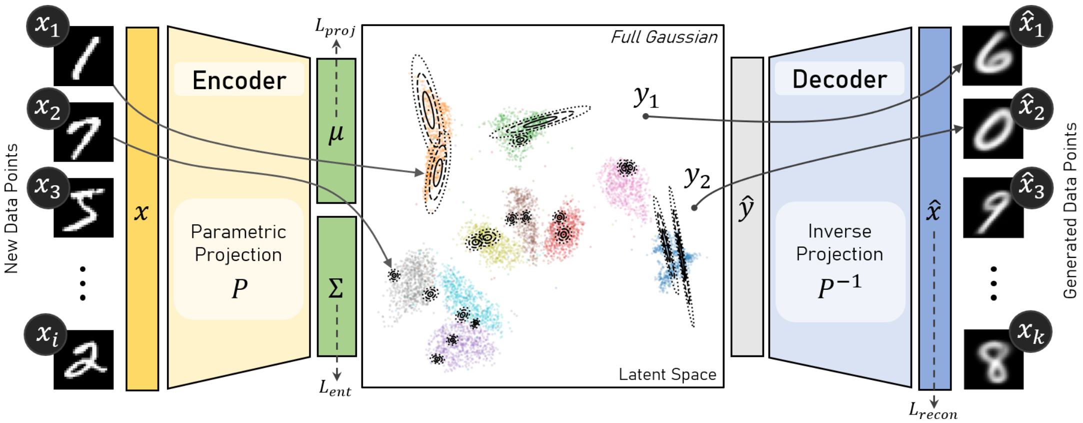

Figure 1: The encoder of a DE-VAE learns a parametric projection P of MNIST, mapping each data point x_i to a full Gaussian $\mathcal{N}(\mu,\Sigma)$, modeling the uncertainty of a UMAP projection. The decoder learns an inverse projection P^{-1}, taking y_k and reconstructing a plausible sample $\hat{x}_k$. P enables uncertainty-aware visualization of the latent space. DE-VAEs optimize the losses: $\mathcal{L}_{\text{recon}}$, ensuring reconstruction; $\mathcal{L}_{\text{proj}}$, aligning μ with points of the projection; and $\mathcal{L}_{\text{ent}}$, maximizing the variance of Σ. To show learned Gaussian distributions, we depict the 1st, 2nd, and 3rd standard deviations as ellipses around two randomly sampled points per class.

Abstract

Recently, autoencoders (AEs) have gained interest for creating parametric and invertible projections of multidimensional data. Parametric projections make it possible to embed new, unseen samples without recalculating the entire projection, while invertible projections allow the synthesis of new data instances. However, existing methods perform poorly when dealing with out-of-distribution samples in either the data or embedding space. Thus, we propose DE-VAE, an uncertainty-aware variational AE using differential entropy (DE) to improve the learned parametric and invertible projections. Given a fixed projection, we train DE-VAE to learn a mapping into 2D space and an inverse mapping back to the original space. We conduct quantitative and qualitative evaluations on four well-known datasets, using UMAP and t-SNE as baseline projection methods. Our findings show that DE-VAE can create parametric and inverse projections with comparable accuracy to other current AE-based approaches while enabling the analysis of embedding uncertainty.

Index Terms: Uncertainty visualization, dimensionality reduction, variational autoencoders, inverse projections.

1 Introduction

Multidimensional projections, also referred to as dimensionality reduction (DR) techniques, are widely used and well-established tools for the visual exploration of high-dimensional data [20]. DR methods are usually used to create 2- or 3-dimensional representations while trying to preserve relationships present in the high-dimensional space, e.g., distances and neighborhoods [27]. In this context, *parametric* projection methods employ a trained model to create a mapping from high- to low-dimensional space, enabling the projection of previously unseen data points, real and synthetic, without the need to recompute the projection [28]. *Inverse* projection methods provide a usually smooth mapping from the projection space to the high-dimensional space, enabling the generation of new data from any position in the projection space [31], e.g., helpful in visualizing and exploring the decision zones of classifiers.

However, DR is inherently *lossy and prone to distortions*, such as false neighborhoods, or rifts, especially, in the case of non-linear DR methods [27]. To address this, researchers proposed methods to quantify and visualize distortions and gradients between projected points. Recent approaches for parametric and inverse projections, which are usually created using neural networks (NNs) [12] or variational autoencoders (VAEs) [8], did not address or leverage the *uncertainty* inherent in the process. In this work, we propose DE-VAEs, which use *differential entropy* to capture the uncertainty in the parametric projection by modeling it in its latent space. Thus, unlike traditional VAEs, DE-VAEs *model* the uncertainty associated with a sample with a multidimensional Gaussian distribution. In contrast, for the inverse- or generative-case, we *sample* from the modeled distribution to retrieve similar synthetic data items. Since DR for data projection is inherently visual [26], we can gain insight into the sources and effects of uncertainty through uncertainty visualization [5]. Overall, we contribute the following:

(1) We propose DE-VAEs, which leverage *differential entropy* for creating parametric and invertible projections.
(2) We perform an evaluation comparing *isotropic*, *diagonal* and *full Gaussian* distributions quantitatively and qualitatively on four datasets using UMAP and t-SNE.
(3) For *reproducibility*, we provide the analysis, results and source code on OSF and GitHub.

With this work, we aim to enhance high-dimensional data analysis by improving the interpretation of uncertainty in DR methods.

[*] e-mail: frederik.dennig@uni-konstanz.de
[†] e-mail: keim@uni-konstanz.de

2 Related Work

First, we introduce some notation. Let $D = \{x_i\}_{1 \leq i \leq n}$ denote a high-dimensional dataset with d dimensions and n samples, where each $x_i \in \mathbb{R}^d$. A *projection* technique P transforms D into a lower-dimensional set $P(D) = \{P(x_i) \mid x_i \in D\} = \{y_i\}_{1 \leq i \leq n}$, such that $P(D) \subset \mathbb{R}^q$ with $q \ll d$. In our setting, $q = 2$, so $P(D)$ can be displayed as a 2D projection.

A plethora of DR methods exist and are studied in several surveys quantitatively and qualitatively [14, 27]. Generally, there are *linear* and *non-linear* methods which focus on either preserving *global structure* or *local neighborhoods*. Linear projection techniques, such as principal component analysis (PCA) [20], are computationally efficient and excel at maintaining the data's global structure. In contrast, non-linear global methods like multidimensional scaling (MDS) [23, 7] aim to capture more complex relationships while still focusing on the overall structure. However, many modern non-linear approaches prioritize preserving local neighborhood relationships, often sacrificing global structure in the process. Notable examples include t-SNE [30], which forms implicit neighborhoods by computing pairwise similarities in high-dimensional space using a Gaussian kernel centered at each data point, while UMAP [25] explicitly constructs a k-nearest neighbor graph to capture local structure and then optimizes a low-dimensional embedding that preserves the fuzzy topology implied by the graph. AE-based models also belong to the class of non-linear methods; they learn low-dimensional embeddings by training neural networks to reconstruct input data, effectively capturing complex, non-linear feature representations [32].

Parametric Projection Methods: These approaches provide an explicit mapping function from the high-dimensional space to a lower-dimensional embedding, enabling efficient out-of-sample projections for new data points [18]. Van der Maaten [29] employed a feed-forward neural network to create a parametric version of t-SNE. In a similar vein, parametric UMAP [28] utilizes neural networks, including autoencoders, as a parametric alternative to the original non-parametric embedding. By training neural networks to predict 2D coordinates for input data points, Espadoto et al. [13] demonstrated that sufficiently large networks can effectively approximate various non-parametric methods.

Inverse Projections: A function $P^{-1} : \mathbb{R}^q \to \mathbb{R}^d$, that is typically smooth and aims to minimize the error $\sum_{x \in D} \|P^{-1}(P(x)) - x\|^2$, where P is a projection and $\|\cdot\|$ denotes the L_2 norm, is called an inverse projection. There are only a limited number of non-NN methods supporting inversion: PCA [20], LAMP [19, 11], and UMAP [25]. At the same time, an autoencoder, by design, has an encoder (learning P) and a decoder (learning P^{-1}) that reconstructs the original data [3]. Other methods augment a subset of DR methods with inverse transformations, making different assumptions. One of the first techniques used a global interpolation-based method to create synthetic data points [31]. Amorim et al. [1] refined this approach by using a radial basis function for kernel-based interpolation. Blumberg et al. [4] invert MDS by applying data point multilateration based on geometrical properties.

Uncertainty-aware Projection Methods: These techniques support probability distributions modeling uncertainty as their input. Görtler et al. [15] proposed a linear DR technique that generalizes PCA to multivariate probability distributions by incorporating input uncertainty directly into the covariance matrix. UAMDS extends MDS by mapping high-dimensional random vectors from any distribution to a low-dimensional space supporting various stress types [16].

3 Differential Entropy in Variational Autoencoders

In general, AEs are a type of NN for unsupervised learning of compressed data representations, i.e., latent representations [3], useful for feature extraction. In contrast, the latent representation of a variational autoencoder (VAE) is a probability distribution [22]. Compared to AEs, VAEs learn a smooth and continuous latent space where nearby points correspond to similar outputs, useful for sampling and interpolation. In recent work, VAEs were trained to jointly create a parametric projection P and an inverse projection P^{-1} [8]. The approach used a modified loss function for VAEs to structure the latent space, allowing for inverting a user-defined projection. However, it did not outperform the more straightforward AE-based method. We provide an alternative approach in this paper.

We show the general idea of a DE-VAE in Fig. 1. A DE-VAE learns to project data items from a ground-truth projection, i.e., a pre-computed projection. In contrast to existing parametric projection methods, DE-VAEs infer Gaussian distributions as there *encoder* output, not just a low-dimensional coordinate. The *decoder* can reconstruct a high-dimensional data point from any 2-dimensional position. More specifically, we target three objectives: (1) Differing from VAEs, we target a latent mean μ alignment such that the *encoder* learns a parametric projection P, and (2) models uncertainty through variance via differential entropy. (3) Similar to VAEs, we target accurate inverse projection P^{-1} or reconstruction of the input data through the *decoder*. Unlike the VAE's latent space, which is shaped by the Kullback-Leibler (D_{KL}) loss term, tying both mean and variance to a *fixed prior* (e.g., $\mathcal{N}(0, I)$), our approach independently controls the mean μ to align it with a projection $P(D)$. Additionally, the encoder infers an associated variance Σ around the projected data point. In contrast to classical variational inference with respect to a prior, our regularization toward an externally chosen target shapes the latent space according to a projection.

In the following, we describe its loss function (see Eq. (1)) in detail. Let $x \in D \subset \mathbb{R}^d$ denoting the original vector, $\hat{x} \in \mathbb{R}^d$ the reconstructed vector output, $\mu \in \mathbb{R}^q$ the latent mean from encoder, $y \in \mathbb{R}^d$ is the target latent vector, i.e. the embedding vector of the projection, and the Gaussian approximate posterior distribution $q(z) := q(z \mid x) = \mathcal{N}(z \mid \mu(x), \Sigma(x))$.

$$\begin{aligned}\mathcal{L}(x, \hat{x}, y, \mu, \Sigma) = \mathcal{L}_{\text{recon}}(x, \hat{x}) + \lambda_{\text{proj}} \cdot \mathcal{L}_{\text{proj}}(y, \mu) \\ + \lambda_{\text{ent}} \cdot \mathcal{L}_{\text{ent}}(\Sigma)\end{aligned} \tag{1}$$

Reconstruction Loss: $\mathcal{L}_{\text{recon}}(x, \hat{x})$ measures how well the model reconstructs the original input x from its latent representation. Minimizing this term ensures that the latent code z retains sufficient information to reproduce x faithfully [22]. We refrain from a concrete choice since it is dataset-dependent, e.g., we use the mean squared error (MSE) and binary cross-entropy (BCE).

Projection Loss: $\mathcal{L}_{\text{proj}}(y, \mu) = \|y - \mu\|^2$ encourages the latent mean μ to align with a target low-dimensional projection y, also called the ground-truth projection. This guides the latent space to reflect the structure of the projection $P(D)$. We use the MSE to measure the numerical difference between predicted and actual values.

Entropy Term: $\mathcal{L}_{\text{ent}}(\Sigma) = -\mathcal{H}[\mathcal{N}(0, \Sigma)]$ promotes higher uncertainty or diversity in the associated latent distribution $\mathcal{N}(\mu, \Sigma)$ by maximizing *differential entropy*. This prevents overly confident or degenerate latent representations and supports smoother, more flexible embeddings. It needs to be carefully balanced, usually with $\mathcal{L}_{rec}$, not to dominate the overall loss (see Eq. (1)).

Weighting terms are standard in loss functions of VAEs [17]. We introduce λ_{proj}, which controls the strength with which the mean values of the latent space distributions will match the ground-truth projection, and λ_{ent}, which, with higher values increases the variance of each of the learned Gaussian distributions. λ_{ent} is the more sensitive of the two and can overpower all other loss components. We discuss the choice of λ_{proj} and λ_{ent} in Sec. 5.

3.1 Latent Gaussian Covariance Structures

In the context of VAEs, $q(z)$ represents the *approximate posterior distribution* over the latent variables z, given the observed data x, i.e, $q(z) \equiv q_\phi(z \mid x) = \mathcal{N}(z \mid \mu(x), \Sigma(x))$ with model parameters

	None	Isotropic	Diagonal	Full
Parametric projection: Average projection loss $\mathcal{L}_{proj}$ (lower is better)				
HAR (UMAP)	.675 ± .685	.525 ± .257	.447 ± 0.166	**.404** ± **.079**
MNIST (UMAP)	.402 ± .064	**.400** ± **.077**	.418 ± .162	.428 ± 0.07
Fashion-MNIST (UMAP)	**.263** ± **.034**	.293 ± .069	.306 ± .059	.298 ± 0.05
KMNIST (t-SNE)	**30.4** ± **4.68**	32.3 ± 3.98	31.6 ± 6.23	33.0 ± 5.27
Inverse Projection: Average reconstruction loss $\mathcal{L}_{recon}$ (lower is better)				
HAR (UMAP)	272 ± 2.41	272 ± 2.42	272 ± 2.45	273 ± 4.73
MNIST (UMAP)	**127** ± **0.85**	135 ± 1.01	136 ± 1.64	138 ± 1.28
Fashion-MNIST (UMAP)	**246** ± **1.93**	252 ± 2.36	251 ± 1.62	253 ± 1.53
KMNIST (t-SNE)	**239** ± **1.64**	246 ± 1.83	245 ± 2.52	248 ± 3.02
Number of training epochs until validation loss convergence (lower is better)				
HAR (UMAP)	41.3 ± 13.1	42.3 ± 12.3	41.2 ± 10.9	**40.3** ± **9.41**
MNIST (UMAP)	32.8 ± 6.54	31.0 ± 4.49	32.8 ± 6.54	**30.3** ± **6.44**
Fashion-MNIST (UMAP)	29.7 ± 7.50	29.0 ± 11.0	32.8 ± 13.9	**28.4** ± **8.46**
KMNIST (t-SNE)	46.3 ± 9.84	**38.5** ± **5.75**	45.9 ± 11.4	39.0 ± 9.65

Table 1: Average losses and standard deviations (after ±) of the parametric and inverse projections on test data for 10 runs each, as well as average number of training epochs to compare training time.

ϕ. In the following, we briefly describe three different Gaussian distributions and how they impact the structure of the latent space (see supplementary material for mathematical deductions).

Isotropic Gaussian: The covariance matrix assumes the same variance across all latent dimensions (see Fig. 2), expressed as $\Sigma = \sigma^2 I$, where I is the identity matrix. This simplification implies that latent variables are identically distributed with equal uncertainty. The differential entropy reduces to a form proportional to the logarithm of the shared variance, which directly contributes to the entropy term in the loss function. The entropy of an isotropic Gaussian $\mathcal{N}(\mu, \sigma^2 I)$ is defined as $H[\mathcal{N}(\mu, \sigma^2 I)] = \frac{d}{2}(1 + \log 2\pi + \log \sigma^2)$.

Diagonal Gaussian: In this case, each latent dimension has its own variance, so $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$. This enables modeling of different uncertainty per dimension while maintaining independence between dimensions, i.e., axis-aligned (see Fig. 2). The entropy term in the loss sums contributions from each dimension's variance, reflecting dimension-specific uncertainty in the latent space. The entropy of a diagonal Gaussian is defined as $\mathcal{H}[\mathcal{N}(\mu, diag(\sigma_1^2, \ldots, \sigma_q^2))] = \frac{1}{2}\sum_i(1 + \log 2\pi + \log \sigma_i^2)$.

Full Gaussian: This is the most general form with covariance $\Sigma = LL^\top$, where L is a lower-triangular matrix obtained via the *Cholesky decomposition*, which captures correlations between latent dimensions, allowing for more flexible uncertainty modeling (see Fig. 2). The differential entropy is computed using the log-determinant of Σ, which in turn is derived from L, encouraging expressive latent representations through the covariance structure. The differential entropy of a full Gaussian is defined as $\mathcal{H}[\mathcal{N}(\mu, L)] = \frac{1}{2}d(1 + \log 2\pi) + \sum \log L_{ii}$.

4 Evaluation

We compare our approach using four datasets, characterized by the number of samples **n**, dimensions **d**, intrinsic dimensionality $\rho_{\mathbf{d}}$, sparsity $\gamma_{\mathbf{n}}$, and **Type**. We use *HAR* [2] (**n**: 735, **d**: 561, $\rho_{\mathbf{d}}$: 120, $\gamma_{\mathbf{n}}$: 0.0%, **Type**: Sensor Data), *MNIST* [24] (**n**: 70000, **d**: 784, $\rho_{\mathbf{d}}$: 330, $\gamma_{\mathbf{n}}$: 82.9%, **Type**: Images), *Fashion-MNIST* [33] (**n**: 3000, **d**: 784, $\rho_{\mathbf{d}}$: 187, $\gamma_{\mathbf{n}}$: 50.2%, **Type**: Images), and *KMNIST* [6] (**n**: 70000, **d**: 784, $\rho_{\mathbf{d}}$: 238, $\gamma_{\mathbf{n}}$: 66.7%, **Type**: Images). We evaluate our approach quantitatively by comparing reconstruction loss and projection loss, as well as runtime measurements such as the number of training epochs and training time, using UMAP [25] and t-SNE [30] with standard parameters as the ground-truth projections. We also compare the approaches qualitatively by visually inspecting the latent space and reconstructions. **None** refers to an AE without a variational component [8], predicting only a latent coordinate (equivalent to μ), and serving as the current competitor in all evaluations. Further details are provided in the supplementary material. Replication data is available on OSF and DaRUS [9].

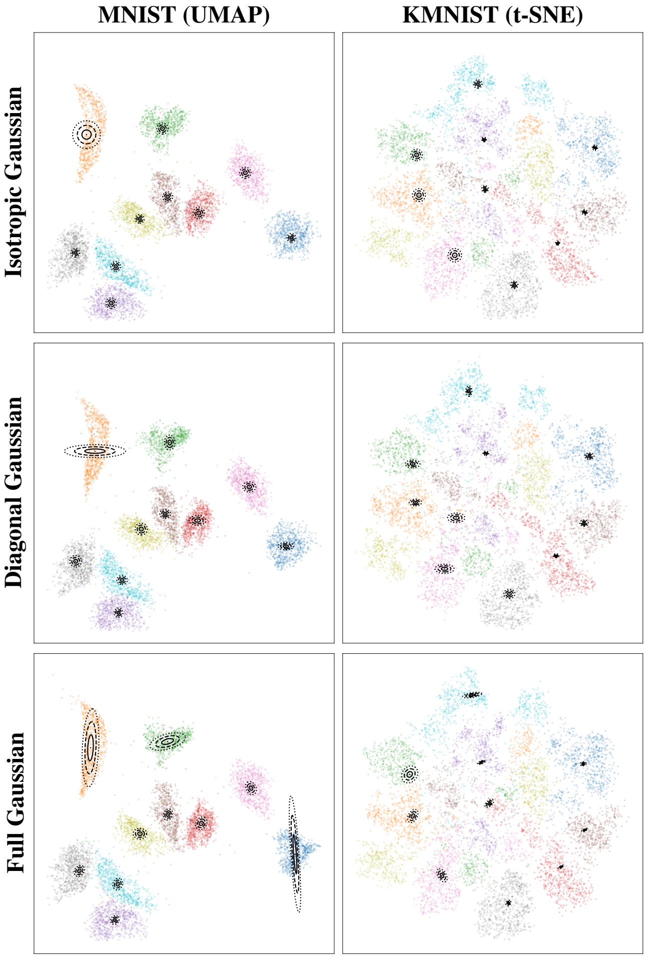

Figure 2: We show the encoder output μ for all test data of the learned UMAP projection of MNIST and the t-SNE projection of KMNIST. The uncertainty modeling, expressed by three Gaussian distribution types, allows for varying levels of expressiveness through their differing degrees of freedom. These are shown as 1st, 2nd, and 3rd standard deviations, depicted as ellipses around the class medoids.

4.1 Data Processing and Training

We use the *Adam optimizer* [21] with default parameters. We set the learning rate to 0.001 and the batch size to 64, which are typical values. We set the number of training epochs to a maximum of 100, allowing training to stop early if the validation loss does not improve for 5 consecutive epochs. We determined the topologies of our AEs and the weighting factors of the loss function λ_{proj} and λ_{ent} through multiple runs, using the validation loss to track the impact of different parameter configurations. For *HAR*, we set $\lambda_{\text{proj}} = 5$ and $\lambda_{\text{ent}} = 0.001$. For *MNIST*, we set $\lambda_{\text{proj}} = 20$ and $\lambda_{\text{ent}} = 5$. For *Fashion-MNIST*, we set $\lambda_{\text{proj}} = 20$ and $\lambda_{\text{ent}} = 4$. Finally, for *KMNIST*, we set $\lambda_{\text{proj}} = 20$ and $\lambda_{\text{ent}} = 3$. For *HAR*, we used the vector data *as-is* while using the MSE as the reconstruction loss. For *MNIST*, *Fashion-MNIST*, and *KMNIST*, we converted the pixel values into the interval $[0, 1]$, enabling us to use binary cross-entropy (BCE) as the reconstruction loss, $\mathcal{L}_{recon}$. In all cases, we used the projection methods with their default parameters without any additional post-processing. For all details, refer to the source code.

4.2 Quantitative Comparison

To evaluate the quality of the resulting parametric and inverse projections, we report $\mathcal{L}_{\text{proj}}$ and $\mathcal{L}_{\text{recon}}$ on the test samples of an 80/10/10 train-validation-test split for each method. For *MNIST*, *Fashion*-

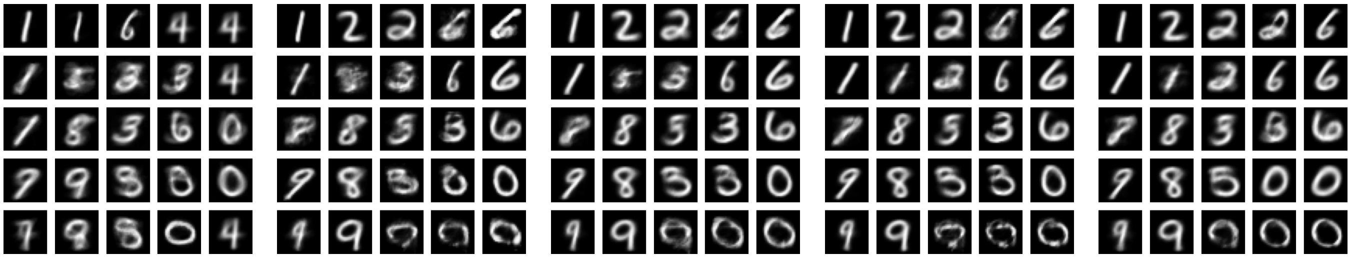
(a) UMAP (inverse) (b) None (AE) (c) Isotropic Gaussian (d) Diagonal Gaussian (e) Full Gaussian

Figure 3: For each approach (a-e), we inverse project 25 samples from an evenly spaced 5×5 grid on a UMAP projection of the *MNIST* dataset and show the results of the inverse projections (P^{-1}). Our models (c-e) show higher quality and more diverse outputs.

MNIST, and *KMNIST*, $\mathcal{L}_{\text{recon}}$ is calculated using BCE, and MSE for *HAR*. In all cases, lower values indicate better performance. $\mathcal{L}_{\text{proj}}$ is always the MSE. We train 10 AEs for each method using different random seeds for model initialization and batch randomization to mitigate outlying measurements. The average losses and the associated standard deviations are reported in Tab. 1.

The average projection loss evaluates how accurately the model's latent space preserves structure from the projection. For *HAR*, there is an improvement as the expressiveness of the covariance increases. The **Full** model achieves the best score, followed by **Diagonal**. The difference between **None** and the others is notable, suggesting that modeling latent uncertainty improves alignment with projections. For *MNIST*, the differences are very small. **Isotropic** slightly outperforms the others. However, all methods perform comparably and the standard deviations overlap. For *Fashion-MNIST* and *KMNIST*, **None** performs best, with increasing projection loss for the other models, indicating over-regularization from uncertainty modeling.

The average reconstruction loss measures how well the decoder reconstructs inputs from latent samples. For *HAR*, all methods perform identically, with negligible variation. For *MNIST*, the plain AE performs best, with increasing loss from **Isotropic** to **Full**. This suggests that more expressive latent noise modeling slightly hurts reconstructions, possibly due to increased uncertainty. Also for *Fashion-MNIST* and *KMNIST*, **None** performs best, though the differences are small. **None** or simple latent distributions are better for reconstruction. This indicates that more complex models introduce uncertainty that can degrade decoder performance.

Comparing the number of training epochs, the **Full** models converge marginally faster, but the training time is longer overall.

4.3 Qualitative Comparison

We visually compare the encoder output created by our three models (see Fig. 2), depicting the 1st, 2nd, and 3rd standard deviations as ellipses around the class medoids, representing the learned Gaussians [16]. As the quantitative analysis shows similar performance across models for the *MNIST* and *KMNIST* datasets for parametric projection, we can confirm that the methods do not differ significantly in the overall structure. All clusters associated with each digit or character are visible. We can also confirm that the choice of Gaussian distribution (i.e., isotropic, diagonal, or full) affects the learned distribution, as indicated by the rotated and strongly elliptical distributions, though less so for *KMNIST*.

We also compare the reconstruction quality of the built-in inverse projection learned by the decoder (see Fig. 3). We can observe that all approaches create clearer outputs than the inverse of UMAP (see Fig. 3 (a)). Our DE-VAEs (see Fig. 3 (c)–(e)) also produce clearer outputs compared to the **None** (see Fig. 3 (b)) that was recently proposed [8]. The comparison between DE-VAEs is not that clear. Given that the **Full** model is most expressive, via the full Gaussian, we can observe that the additional degrees of freedom do not impact reconstruction quality, at least for out-of-distribution samples.

5 Discussion and Limitations

In principle, other spread measures exist beyond differential entropy. However, differential entropy provides an interpretable and differentiable measure of uncertainty that applies across different Gaussian families. We acknowledge that an evaluation focusing on UMAP and t-SNE limits the generalizability of this work. We used UMAP because it provides a built-in P^{-1}. However, UMAP and t-SNE create strong cluster patterns with few to no outliers.

Entropy-based Loss: While our loss shares similarities with that of a standard VAE, it does not fully satisfy the Evidence Lower Bound (ELBO). Our approach replaces the D_{KL} with a differential entropy term, which encourages spread in the latent space but does not match a specific prior. Thus, our model can reconstruct inputs and preserve the projection, but does not support generative sampling from a prior. This may limit its ability to generate new data points from latent samples without additional regularization.

Loss Weights λ_{proj} and λ_{ent}: We determined λ_{proj} and λ_{ent} experimentally by tracking the individual loss terms. We noticed that the other loss terms can be overpowered by a high λ_{ent}, which can cause the model to degenerate. Thus, it is key to balance λ_{ent} with the reconstruction loss, such that $\mathcal{L}_{\text{ent}}$ is still converging in the training process. As a possible solution, we suggest annealing λ_{ent} to maximize the entropy of the latent Gaussians. Generally, we found that models are less sensitive to high λ_{proj}.

Future Work: We aim to broaden the evaluation to include additional projection techniques, such as MDS. We also intend to compare our AE-based models to established parametric and inverse projection methods. Using standard datasets and an 80/10/10 train-validation-test split, we plan to assess the robustness of each method with respect to training set size and intrinsic dimensionality. The differential entropy of the VAE serves as a form of implicit regularization for the latent space, which should yield a smooth inverse projection, but this effect has not yet been evaluated. The method can be extended to support layout enrichment [10] by combining the Gaussian distributions into a Gaussian mixture model.

6 Conclusion

We proposed DE-VAEs, three VAEs with differential entropy-based structuring of the latent space with isotropic, diagonal, and full Gaussians, for generating *parametric* and *invertible* multidimensional data projections. The learned distributions differ notably across the models, thus providing a method to model uncertainty. Qualitative evaluations with UMAP showed that DE-VAEs can outperform the other tested models regarding reconstruction quality for out-of-distribution samples. Quantitatively, they did not consistently outperform competing approaches. The parameterization of the loss function weights remains dependent on the dataset and projection.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (Project A03).

References

[1] E. Amorim, E. V. Brazil, J. Mena-Chalco, L. Velho, L. G. Nonato, F. Samavati, and M. C. Sousa. Facing the high-dimensions: Inverse projection with radial basis functions. *Comput. Graph.*, 48:35–47, 2015. doi: 10.1016/J.CAG.2015.02.009 2

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient Assist. Living Home Care*, pp. 216–223. Springer, 2012. 3

[3] D. Bank, N. Koenigstein, and R. Giryes. *Autoencoders*, pp. 353–374. Springer, 2023. doi: 10.1007/978-3-031-24628-9_16 2

[4] D. Blumberg, Y. Wang, A. Telea, D. A. Keim, and F. L. Dennig. MultiInv: Inverting multidimensional scaling projections and computing decision maps by multilateration. *Comput. Graph.*, 129:104234, 2025. doi: 10.1016/j.cag.2025.104234 2

[5] G. Bonneau, H. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. Overview and state-of-the-art of uncertainty visualization. In *Scientific Visualization*, pp. 3–27. Springer, 2014. doi: 10.1007/978-1-4471-6497-5_1 1

[6] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018. 3

[7] J. de Leeuw and P. Mair. Multidimensional scaling using majorization: SMACOF in R. *J. Stat. Softw.*, 31(3):1—30, 2009. doi: 10.18637/jss.v031.i03 2

[8] F. L. Dennig, N. Geyer, D. Blumberg, Y. Metz, and D. A. Keim. Evaluating autoencoders for parametric and invertible multidimensional projections. In *16th Int. EuroVis Workshop Vis. Anal. (EuroVA)*, 2025. doi: 10.2312/eurova.20251099 1, 2, 3, 4

[9] F. L. Dennig and D. Keim. Replication Data for: "DE-VAE: Revealing Uncertainty in Parametric and Inverse Projections with Variational Autoencoders using Differential Entropy", 2025. doi: 10.18419/DARUS-5258 3

[10] F. L. Dennig, M. Miller, D. A. Keim, and M. El-Assady. FS/DS: A theoretical framework for the dual analysis of feature space and data space. *IEEE Trans. Vis. Comput. Graph.*, 30(8):5165–5182, 2024. doi: 10.1109/TVCG.2023.3288356 4

[11] E. P. dos Santos Amorim, E. V. Brazil, J. D. II, P. Joia, L. G. Nonato, and M. C. Sousa. iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In *7th IEEE Conf. Vis. Anal. Sci. Technol.*, pp. 53–62, 2012. doi: 10.1109/VAST.2012.6400489 2

[12] M. Espadoto, G. Appleby, A. Suh, D. Cashman, M. Li, C. Scheidegger, E. W. Anderson, R. Chang, and A. C. Telea. UnProjection: Leveraging inverse-projections for visual analytics of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 29(2):1559–1572, 2021. doi: 10.1109/TVCG.2021.3125576 1

[13] M. Espadoto, N. S. T. Hirata, and A. C. Telea. Deep learning multidimensional projections. *Inf. Vis.*, 19(3):247–269, 2020. doi: 10.1177/1473871620909485 2

[14] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Trans. Vis. Comput. Graph.*, 27(3):2153–2173, 2019. doi: 10.1109/TVCG.2019.2944182 2

[15] J. Görtler, T. Spinner, D. Streeb, D. Weiskopf, and O. Deussen. Uncertainty-aware principal component analysis. *IEEE Trans. Vis. Comput. Graph.*, 26(1):822–831, 2020. doi: 10.1109/TVCG.2019.2934812 2

[16] D. Hägele, T. Krake, and D. Weiskopf. Uncertainty-aware multidimensional scaling. *IEEE Trans. Vis. Comput. Graph.*, 29(1):23–32, 2023. doi: 10.1109/TVCG.2022.3209420 2, 4

[17] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *5th Int. Conf. Learn. Represent.*, 2017. 2

[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647 2

[19] P. Joia, D. B. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2563–2571, 2011. doi: 10.1109/TVCG.2011.220 2

[20] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986. doi: 10.1007/978-1-4757-1904-8 1, 2

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd Int. Conf. Learn. Represent.*, 2015. 3

[22] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *2nd Int. Conf. Learn. Represent.*, 2014. 2

[23] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964. doi: 10.1007/BF02289565 2

[24] Y. LeCun, C. Cortes, and C. Burges. MNIST handwritten digit database, 1998. 3

[25] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018. doi: 10.48550/arXiv.1802.03426 2, 3

[26] Q. Q. Ngo, F. L. Dennig, D. A. Keim, and M. Sedlmair. Machine learning meets visualization - Experiences and lessons learned. *it Inf. Technol.*, 64(4-5):169–180, 2022. doi: 10.1515/ITIT-2022-0034 1

[27] L. G. Nonato and M. Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Trans. Vis. Comput. Graph.*, 25(8):2650–2673, 2019. doi: 10.1109/TVCG.2018.2846735 1, 2

[28] T. Sainburg, L. McInnes, and T. Q. Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Comput.*, 33(11):2881–2907, 2021. doi: 10.1162/NECO_A_01434 1, 2

[29] L. van der Maaten. Learning a parametric embedding by preserving local structure. In *12th Int. Conf. Artif. Intell. Stat.*, vol. 5, pp. 384–391, 2009. 2

[30] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(86):2579–2605, 2008. 2, 3

[31] J. J. van Wijk and C. W. A. M. van Overveld. Preset based interaction with high dimensional parameter spaces. In *Data Visualization: The State of the Art*, pp. 391–406, 2003. 1, 2

[32] Y. Wang, H. Yao, and S. Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016. doi: 10.1016/j.neucom.2015.08.104 2

[33] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. 3