11 Supplemental Material

11.1 Experiment Metric: Correlation

In addition to the three metrics presented in the Section 6.1 (trustworthiness, continuity, and hit rate), we also measured correlation as a metric for evaluating the quality of the projections generated by HyperNP. Correlation refers to how similar the pairwise distances are between any two points in the 2D projection and in $\mathbb{R}^n$.

Specifically, we compute correlation as follows:

$$r = pearsonr(\|P_i^{(N)} - P_j^{(N)}\|, \|P_i^{(2)} - P_j^{(2)}\|)\ \forall i, j \in P, i \neq j \tag{1}$$

where $P_i^{(N)}$ refers to the position of point i in $\mathbb{R}^n$ and $P_i^{(2)}$ the position in the projected 2D space. The function *pearsonr* computes the Pearson correlation between the two sets of distances. This approach is seen in Geng et al. [GZZ05] and is conceptually similar to the use of a Shepard diagram[JCC*11] as a measure of goodness-of-fit. Just as we did with the hit rate metric we calculate correlation on a uniform sampling of 10% of the data for computational ease.

11.2 STL10 Experiment

In addition to evaluating HyperNP with the **MNIST**, **FashionMNIST**, and the **GloVe** datasets, we also tested HyperNP with the **STL10** data[CNL11]. **STL10** is an image dataset with 100,000 unlabeled colored images of 96x96 pixel resolution. Each of the images falls into 10 possible classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck).

For our experiment using **STL10**, we first performed preprocessing to reduce the images to 28x28 pixels and transformed them to greyscale. The images were then standardized and projected across nearest neighbor values 2 through 50, with a gap size of 8. Additionally, since the data we are using from STL10 does not contain labels, we cluster the dataset using k-means clustering choosing $k = 10$. This should mimic a real world scenario where access to labeled data is not available.

To evaluate the impact of the sampling algorithm has on the performance of HyperNP, we compared two sampling techniques: uniform sampling and stratified sampling. In particular, in this experiment one HyperNP model is trained using a 20% simple uniform sampling of the data, and a second is trained using a 20% stratified sampling that leverages the cluster labels. We repeat this process four times for 4 different seedings of the sampling. The quantitative results are averaged and displayed in Figure 9.

Figure 9 displays a parallel coordinates plot of the aggregated trust, continuity, correlation, and hit rate scores of the experiment. Hit rate is calculated based on the k-means clustering labels described above. As shown in the figure, there is very little difference between the two sampling methods across all of the metrics explored. The ground truth projections perform roughly 4% better on Trust, and 3% better in terms of hit rate.

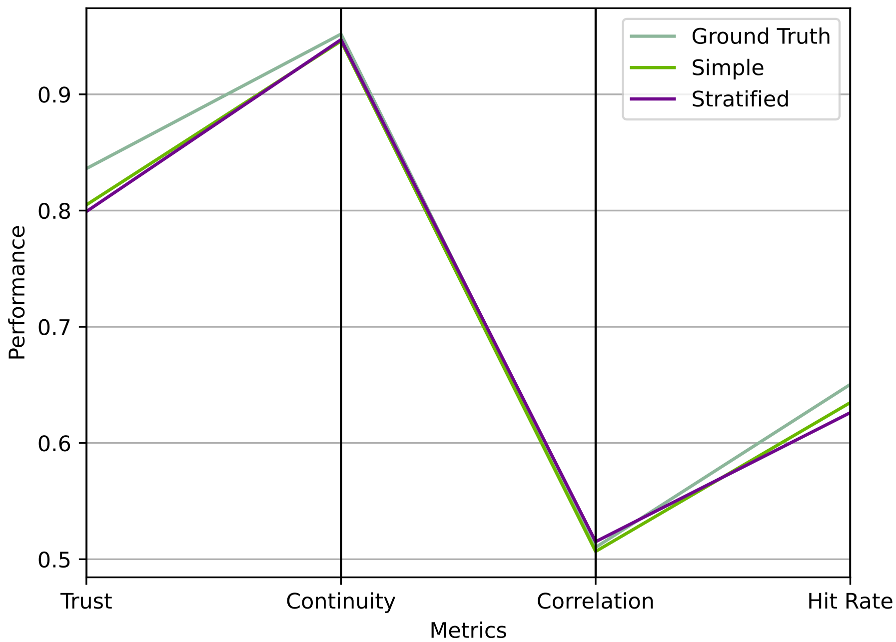


Figure 9: The performance as measured in trust, continuity, correlation, and hit rate. The HyperNP model was trained on 20% of the data, which was sampled either through simple uniform sampling or stratified sampling, and a gap size of 8. The results suggest that the sampling methods have little effect on the performance on HyperNP. See Section 11.2 for more detail.

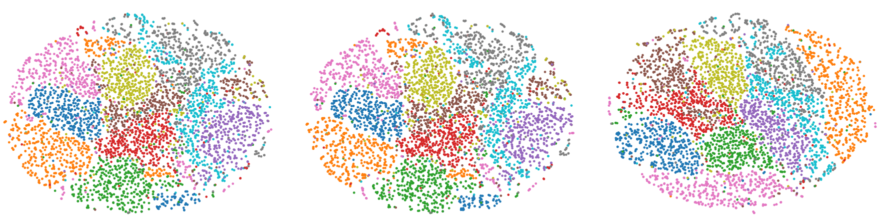
11.3 Projection Instability

Section 3.3 discusses projection instabilities. We refer to the two of classes of instabilities discussed as seeding instabilities and eigenanalysis instabilities. Figures 10 and 11 illustrate seeding instabilities. Image 10(a) shows a UMAP projection of the MNIST dataset using the hyperparameter $\mathbf{h} = 3$ nearest neighbors. Simply re-running UMAP for $\mathbf{h} = 4$, the next possible hyperparameter value, yields image 10(c), in which many of the point clusters move significantly as compared to 10(a), as the color based on point labels shows – see the light blue, dark purple, and green clusters. In contrast, image 10(b) seeds UMAP for $\mathbf{h} = 4$ by the 2D scatterplot in 10(a). The resulting projection is now much closer to 10(a), which is desired. A similar discussion explains Figure 11.

Figure 12 demonstrates eigenanalysis instabilities. Image 12(a) shows Isomap run with $\mathbf{h} = 3$ nearest neighbors on the MNIST dataset. Images 12(b-e) show Isomap on the same dataset for $\mathbf{h} = 4$ where the projections result in different "flipped" orientations. Our heuristic (as described in Section 3.3) used to pick the next projection results in 12(b) – the most similar to the original 12(a) in orientation.

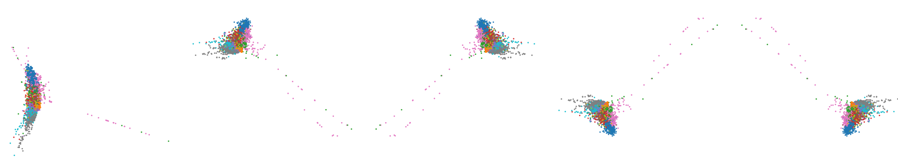
(a) $k = 3$, Random (b) $k = 4$, Previous (c) $k = 4$, Random

Figure 10: Three UMAP projections of the MNIST dataset. (a) uses three nearest neighbors when constructing the kNN graph, while (b) and (c) both use four nearest neighbors. (b) initializes its embedding with positions from (a), while (c) is initialized as usual. Note that (a) and (c) appear differently due to the lack of projection stability, whereas (a) and (b) are more similar because of the initialization.



(a) $k = 3$, Random (b) $k = 4$, Previous (c) $k = 4$, Random

Figure 11: Three t-SNE projections of the MNIST dataset. (a) uses three nearest neighbors when constructing the kNN graph, while (b) and (c) both use four nearest neighbors. (b) initializes its embedding with positions from (a), while (c) is initialized as usual. Note that (a) and (c) appear differently due to the lack of projection stability, whereas (a) and (b) are more similar because of the initialization.

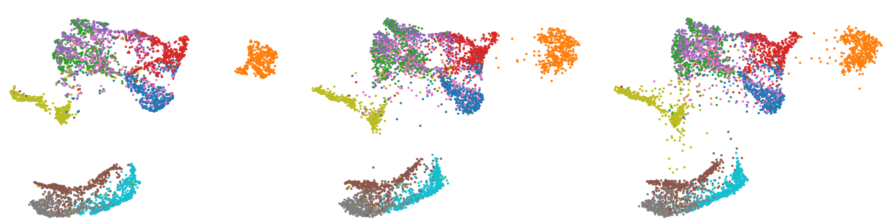


(a) $k = 3$ (b) $k = 4$, Possibility #1 (c) $k = 4$, Possibility #2 (d) $k = 4$, Possibility #3 (e) $k = 4$, Possibility #4

Figure 12: Five Isomap projections of the MNIST dataset. (a) uses $\mathbf{h} = 3$ nearest neighbor, while (b-e) use $\mathbf{h} = 4$. (b-e) show the four possible orientations due to Eigenanalysis instability (see Section 3.3 for discussion) that can result from Isomap. Our technique produces image (b) – the most similar of the four to the original image (a) in orientation.

11.4 Out-Of-Sample Qualitative Performance

In Section 5 we briefly explore the qualitative out-of-sample qualitative performance of HyperNP on t-SNE and Isomap in Figures 2 and 4 respectively. The same idea is demonstrated for UMAP in Figure 13.



(a) Ground Truth (b) Train Predictions (c) Test Predictions

Figure 13: HyperNP approximation of the UMAP projections of the MNIST dataset. From left to right: (a) UMAP projection used for training with 20% data, (b) HyperNP projection of the same data used in (a), (c) HyperNP projection of data instances unseen during training. This result suggests that HyperNP is adequately learning the UMAP projection using just 20% of the data as these three images are visually similar.

11.5 Interpolating Natural Number Hyperparameters

Section 5 also contains a discussion of interpolating between natural number hyperparameters, using the number of neighbors in UMAP as an example (Figure 3). We present a similar figure this time using Isomap in Figure 14.

(a) k=3 (b) k=3.5 (c) k=4

Figure 14: Using HyperNP to explore values of the number of nearest-neighbor hyperparameter, k, in Isomap on the FashionMNIST data set. From left to right: (a) $k = 3.0$, (b) $k = 3.5$, (c) $k = 4.0$. While non-natural values for the parameter k are not valid, we show that HyperNP is able to smoothly interpolate between meaningful values without sacrificing projection quality.

11.6 Composites

This section contains composites of ground truth projections along with HyperNP approximations for all of the combinations of dataset and projection not presented in Section 5. Of particular note are the composites featuring GloVe, which are Tables 6, 8, and 11. These tables demonstrate how HyperNP faithfully reconstructs poor projections. This is discussed in more detail in Section 8.3.

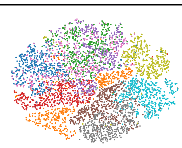
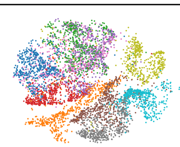
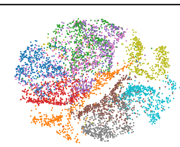
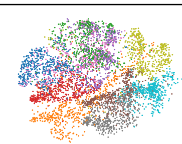
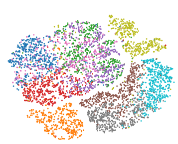
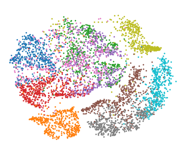
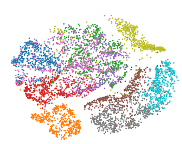
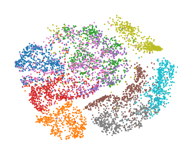
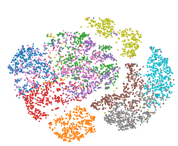
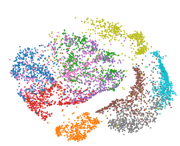
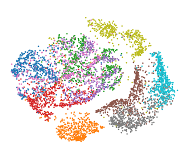
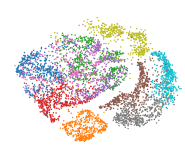
p	t-SNE		HyperNP	
		Gap=2	Gap=4	Gap=8
5				
15				
25				

Table 5: t-SNE and HyperNP projections for three different perplexity values. First column: FashionMNIST projected with t-SNE projections at perplexity values of $p = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

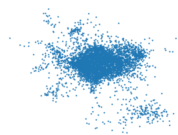
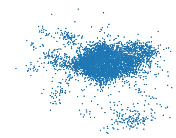
p	t-SNE	HyperNP Gap=2	HyperNP Gap=4	HyperNP Gap=8
5				
15				
25				

Table 6: t-SNE and HyperNP projections for three different perplexity values. First column: GloVe projected with t-SNE projections at perplexity values of $p = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

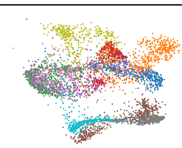
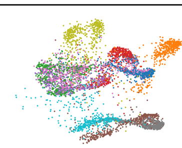
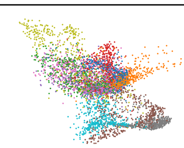
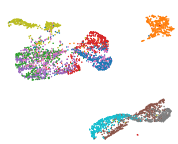
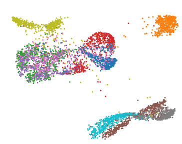
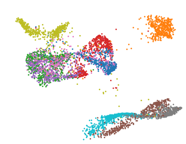
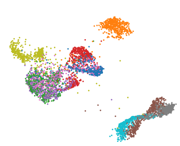
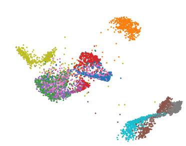
k	UMAP	HyperNP Gap=2	HyperNP Gap=4	HyperNP Gap=8
5				
15				
25				

Table 7: UMAP and HyperNP projections for three different values of the nearest neighbors parameter. First column: FashionMNIST projected with UMAP projections at nearest neighbor values of $k = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

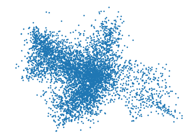
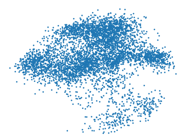
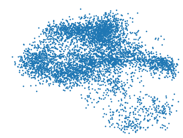
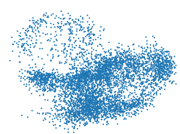
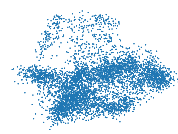
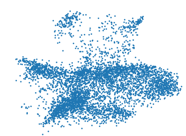
k	UMAP	HyperNP Gap=2	HyperNP Gap=4	HyperNP Gap=8
5				
15				
25				

Table 8: UMAP and HyperNP projections for three different nearest neighbors values. First column: GloVe projected with UMAP projections at nearest neighbor values of $k = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

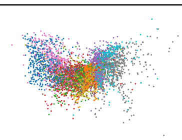
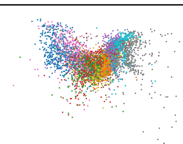
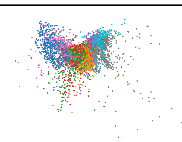
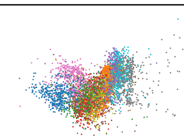
k	Isomap	HyperNP Gap=2	HyperNP Gap=4	HyperNP Gap=8
5				
15				
25				

Table 9: Isomap and HyperNP projections for three different nearest neighbors values. First column: MNIST projected with Isomap projections at nearest neighbor values of $k = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

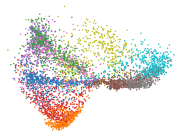
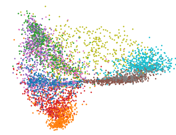
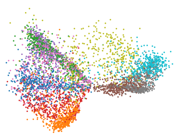
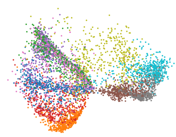
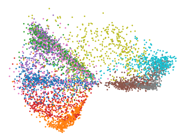
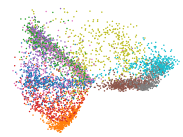
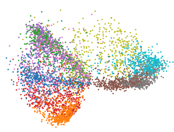
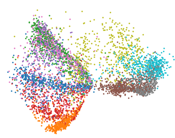
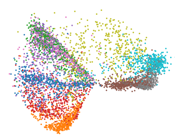
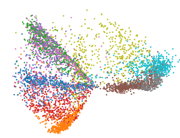
k	Isomap		HyperNP	
		Gap=2	Gap=4	Gap=8
5				
15				
25				

Table 10: Isomap and HyperNP projections for three different nearest neighbors values. First column: FashionMNIST projected with Isomap projections at nearest neighbor values of $k = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

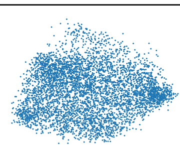
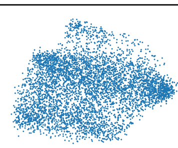
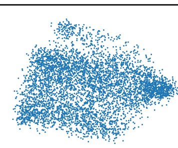
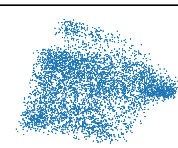
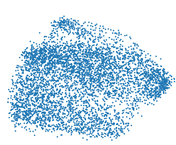
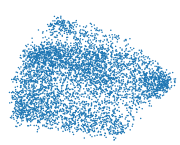
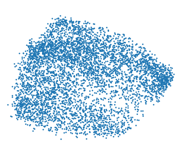
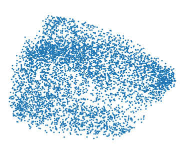
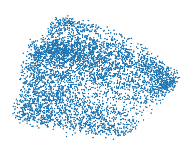
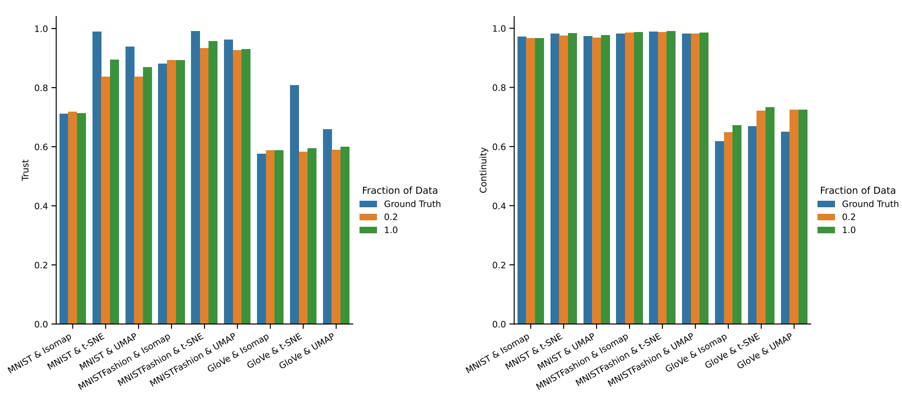
k	Isomap		HyperNP	
		Gap=2	Gap=4	Gap=8
5				
15				
25				

Table 11: Isomap and HyperNP projections for three different nearest neighbors values. First column: GloVe projected with Isomap projections at nearest neighbor values of $k = \{5, 15, 25\}$. The following three columns: HyperNP results when trained with gap values of 2, 4, and 8.

11.7 Aggregate Trustworthiness and Continuity

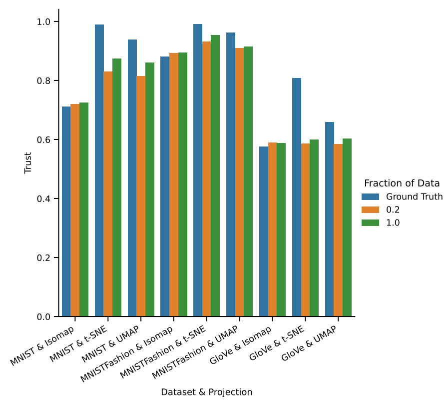
Aggregate trustworthiness and continuity is discussed in Section 6.1, but focuses a gap size of 16, which is the largest gap size tested. The following figures present the metrics for gap sizes of 2, and 8.



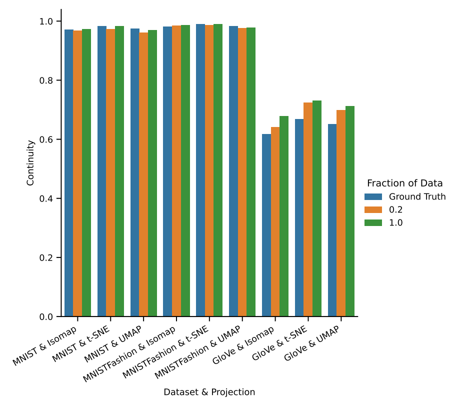
(a) Trustworthiness

(b) Continuity

Figure 15: In this figure we present the trustworthiness and continuity scores associated with the ground truth projections as well as HyperNP trained used a gap size of 2, and data fractions of .2 and 1.0. We have averaged the scores over all of the hyperparameter values from 2 to 50.



(a) Trustworthiness

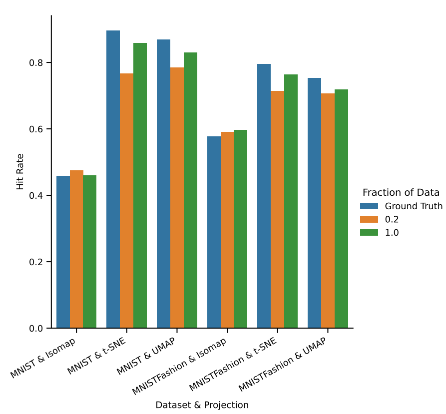


(b) Continuity

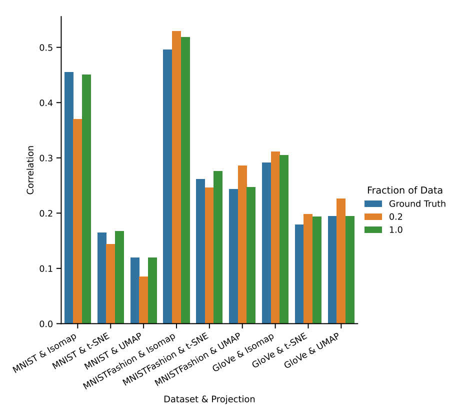
Figure 16: In this figure we present the trustworthiness and continuity scores associated with the ground truth projections as well as HyperNP trained used a gap size of 8, and data fractions of .2 and 1.0. We have averaged the scores over all of the hyperparameter values from 2 to 50.

11.8 Aggregate Hit Rate and Correlation

Aggregate hit rate was discussed in Section 6.1 and correlation was discussed Section 11.1. The following figures present the metrics for gap sizes of 2, 8, and 16.

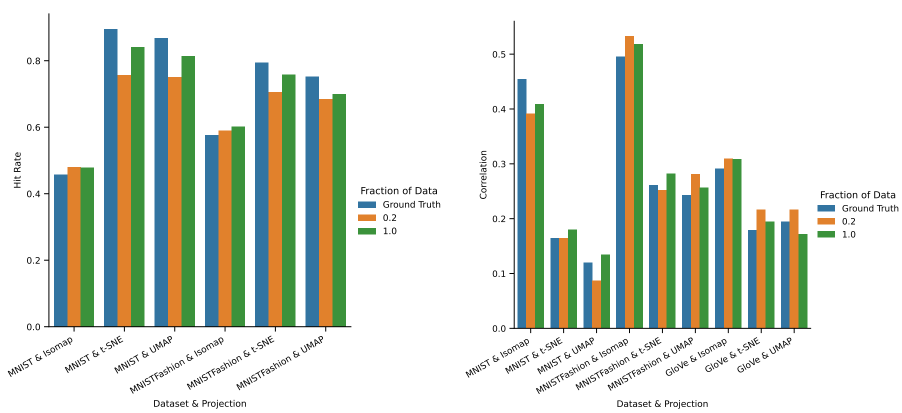


(a) Hit Rate



(b) Correlation

Figure 17: In this figure we present the hit rate and correlation scores associated with the ground truth projections as well as HyperNP trained used a gap size of 2, and data fractions of .2 and 1.0. We have averaged the scores over all of the hyperparameter values from 2 to 50.



(a) Hit Rate

(b) Correlation

Figure 18: In this figure we present the hit rate and correlation scores associated with the ground truth projections as well as HyperNP trained used a gap size of 8, and data fractions of .2 and 1.0. We have averaged the scores over all of the hyperparameter values from 2 to 50.

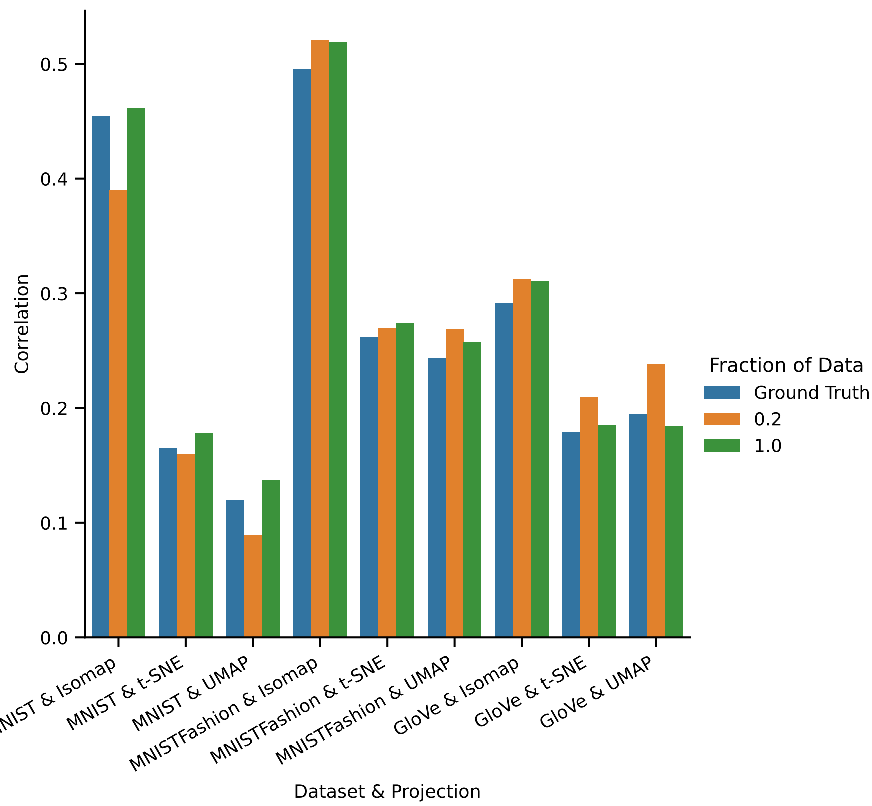
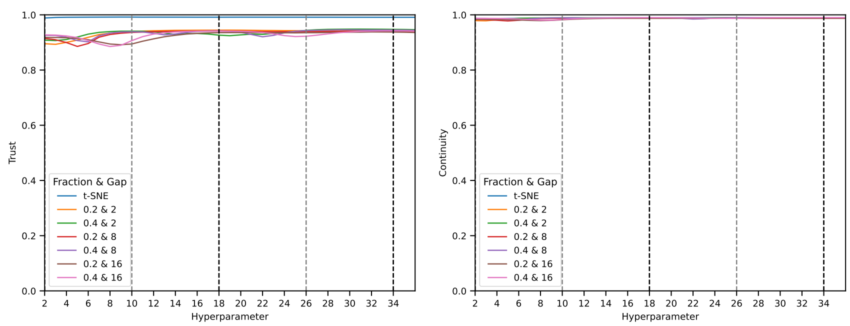


Figure 19: In this figure we present the correlation scores associated with the ground truth projections as well as HyperNP trained used a gap size of 16, and data fractions of .2 and 1.0. We have averaged the scores over all of the hyperparameter values from 2 to 50.

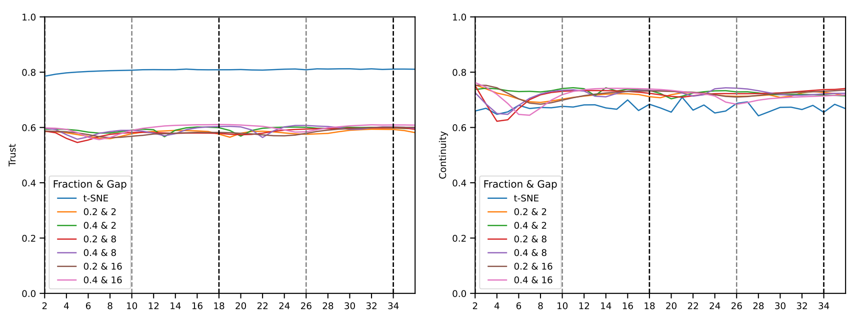
11.9 Non-Aggregate Trustworthiness and Continuity

This subsection contains the non-aggregated trustworthiness and continuity scores for each dataset and projection combination other than t-SNE and MNIST, which appear in Section 6.1.



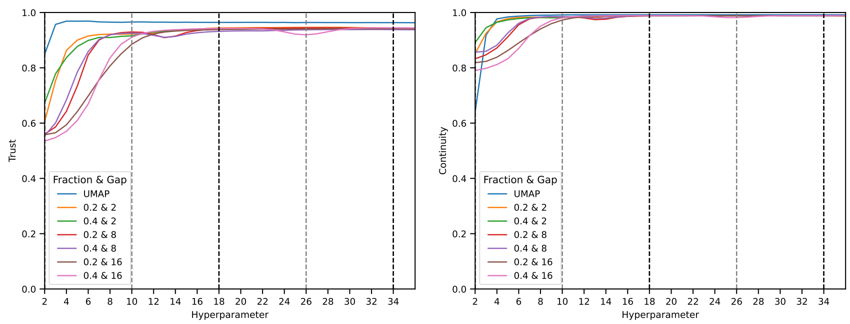
(a) Trustworthiness (b) Continuity

Figure 20: Continuity and trustworthiness for HyperNP trained with t-SNE, on the FashionMNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



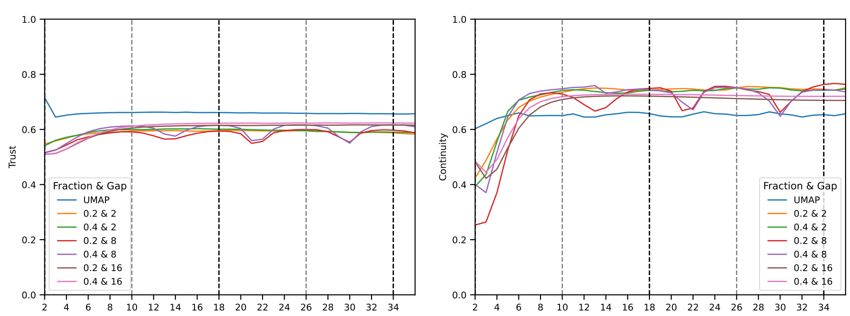
(a) Trustworthiness (b) Continuity

Figure 21: Continuity and trustworthiness for HyperNP trained with t-SNE, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



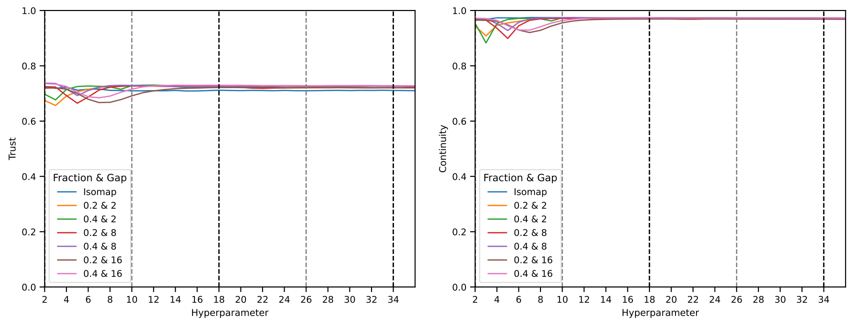
(a) Trustworthiness (b) Continuity

Figure 22: Continuity and trustworthiness for HyperNP trained with UMAP, on the FashionMNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Trustworthiness (b) Continuity

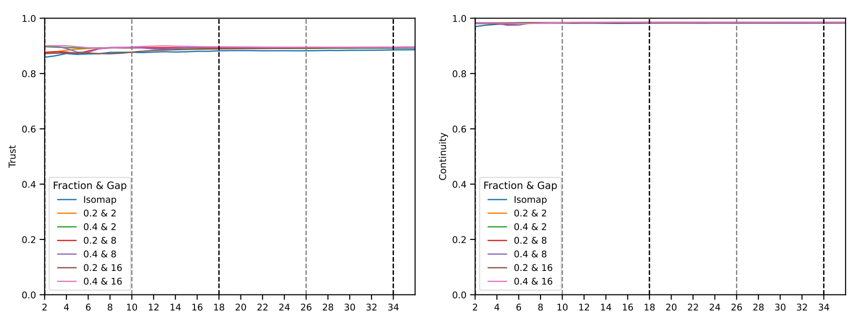
Figure 23: Continuity and trustworthiness for HyperNP trained with UMAP, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Trustworthiness

(b) Continuity

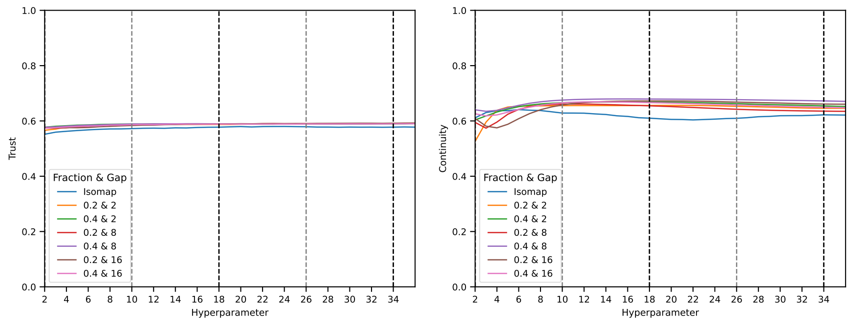
Figure 24: Continuity and trustworthiness for HyperNP trained with Isomap, on the MNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Trustworthiness

(b) Continuity

Figure 25: Continuity and trustworthiness for HyperNP trained with Isomap, on the FashionMNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Trustworthiness (b) Continuity

Figure 26: Continuity and trustworthiness for HyperNP trained with Isomap, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.

11.10 Non-Aggregate Hit Rate and Correlation

This subsection contains the non-aggregated hit rate and correlation scores that did not appear in Section 6.1.

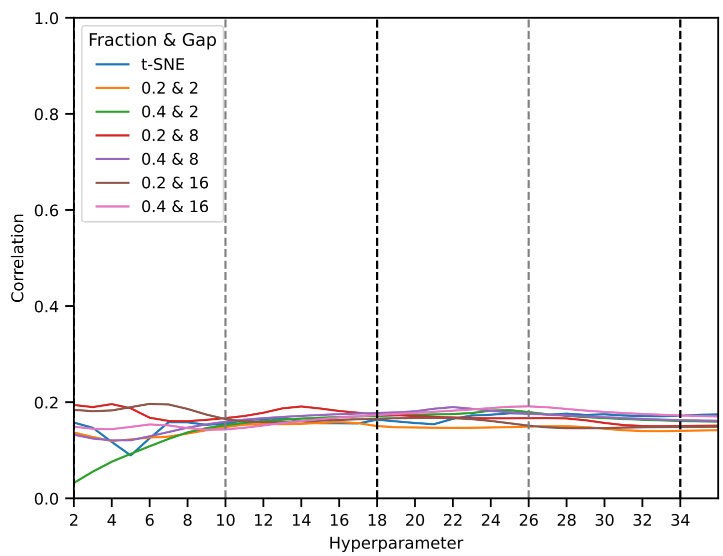
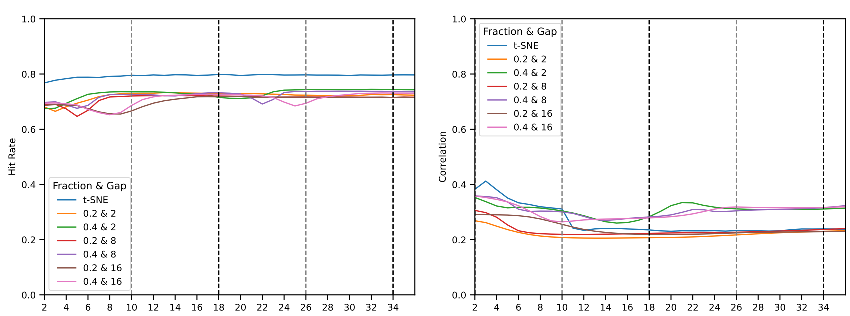


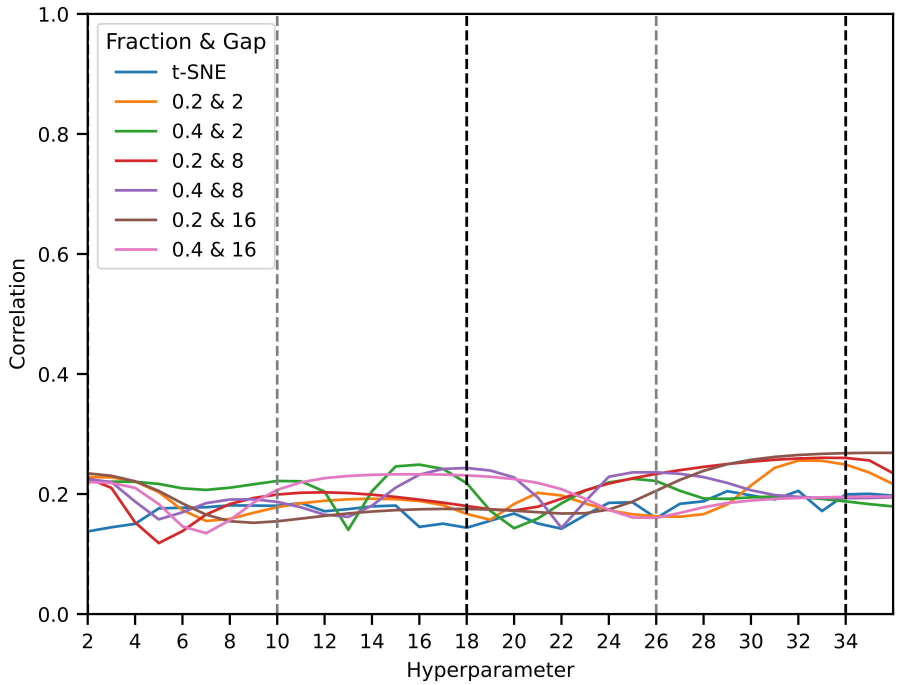
Figure 27: Correlation for HyperNP trained with t-SNE, on the MNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Hit Rate

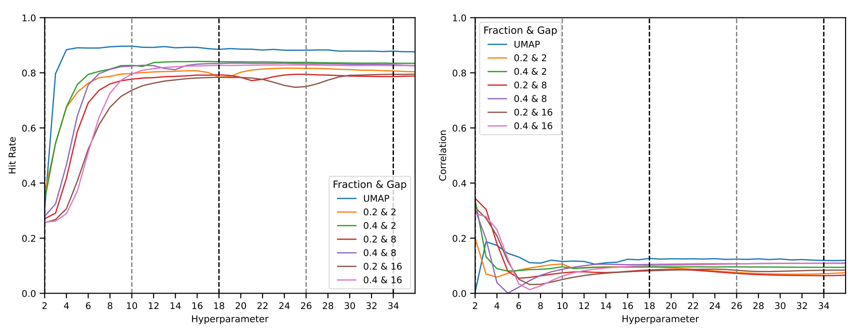
(b) Correlation

Figure 28: Hit rate and correlation for HyperNP trained with t-SNE, on the FashionMNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Correlation

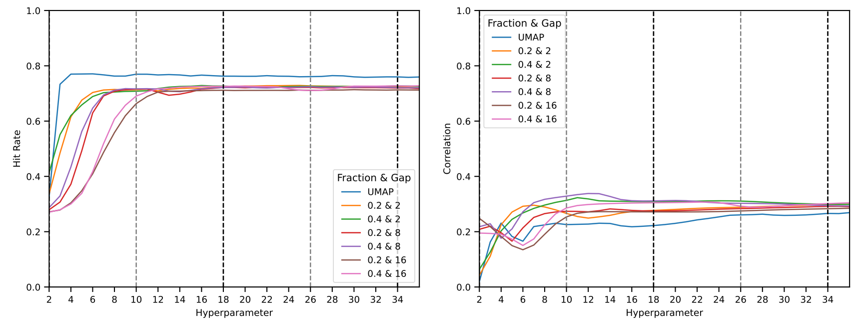
Figure 29: Correlation for HyperNP trained with t-SNE, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Hit Rate

(b) Correlation

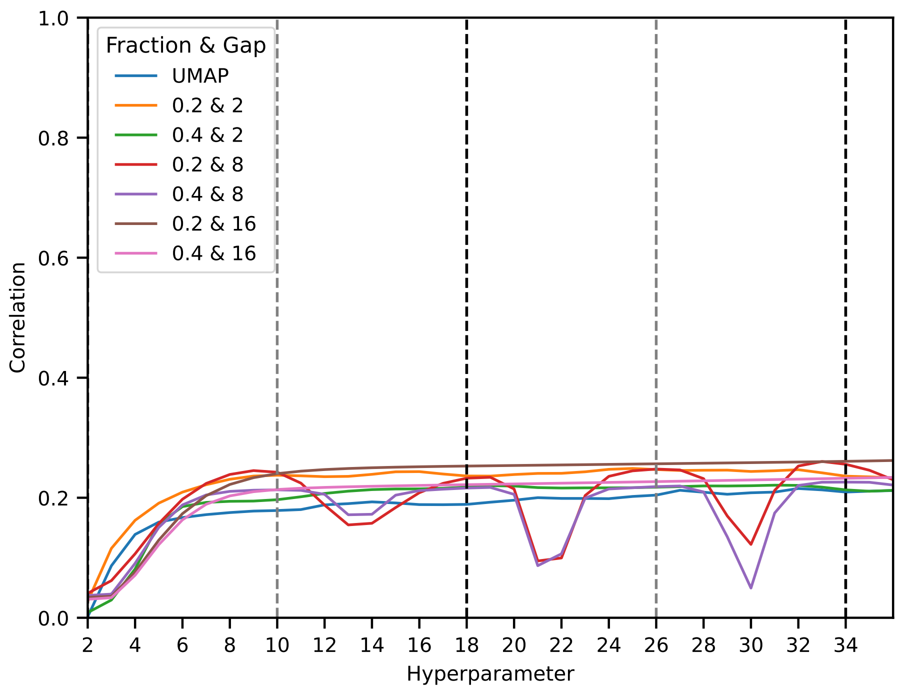
Figure 30: Hit rate and correlation for HyperNP trained with UMAP, on the MNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Hit Rate

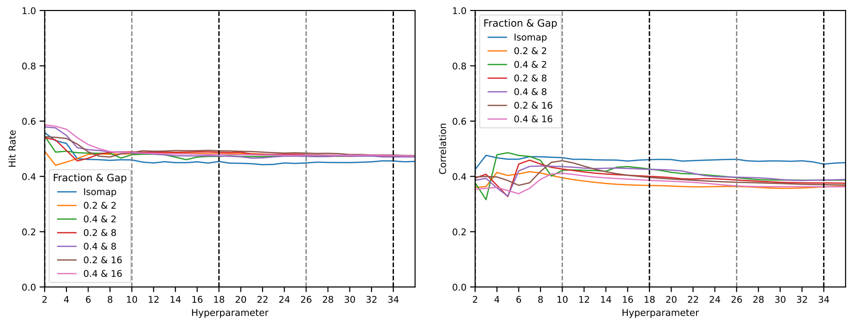
(b) Correlation

Figure 31: Hit rate and correlation for HyperNP trained with UMAP, on the FashionMNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



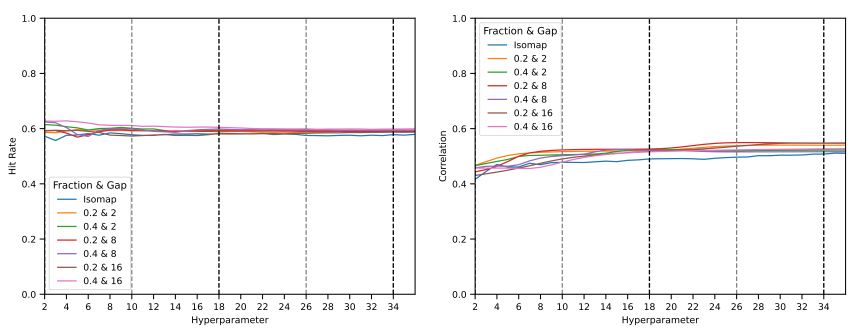
(a) Correlation

Figure 32: Correlation for HyperNP trained with UMAP, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



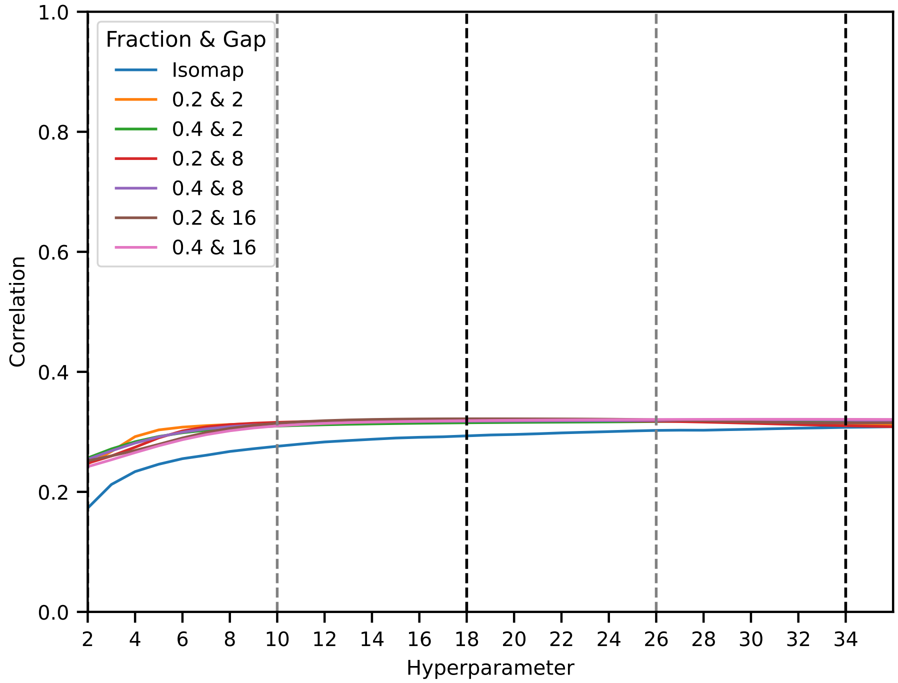
(a) Hit Rate (b) Correlation

Figure 33: Hit rate and correlation for HyperNP trained with Isomap, on the MNIST dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Hit Rate (b) Correlation

Figure 34: Hit rate and correlation for HyperNP trained with Isomap, on the Fashion dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.



(a) Correlation

Figure 35: Correlation for HyperNP trained with Isomap, on the GloVe dataset, for different hyperparameter values. Light gray vertical lines show parameter values used for training with gap $g = 8$; dark gray ones show parameter values used for training with $g = 16$.

11.11 Model Architecture

The architecture of HyperNP's neural network is discussed in Section 3.4 of the main text. Figure 36 illustrates the final configuration of the model chosen.

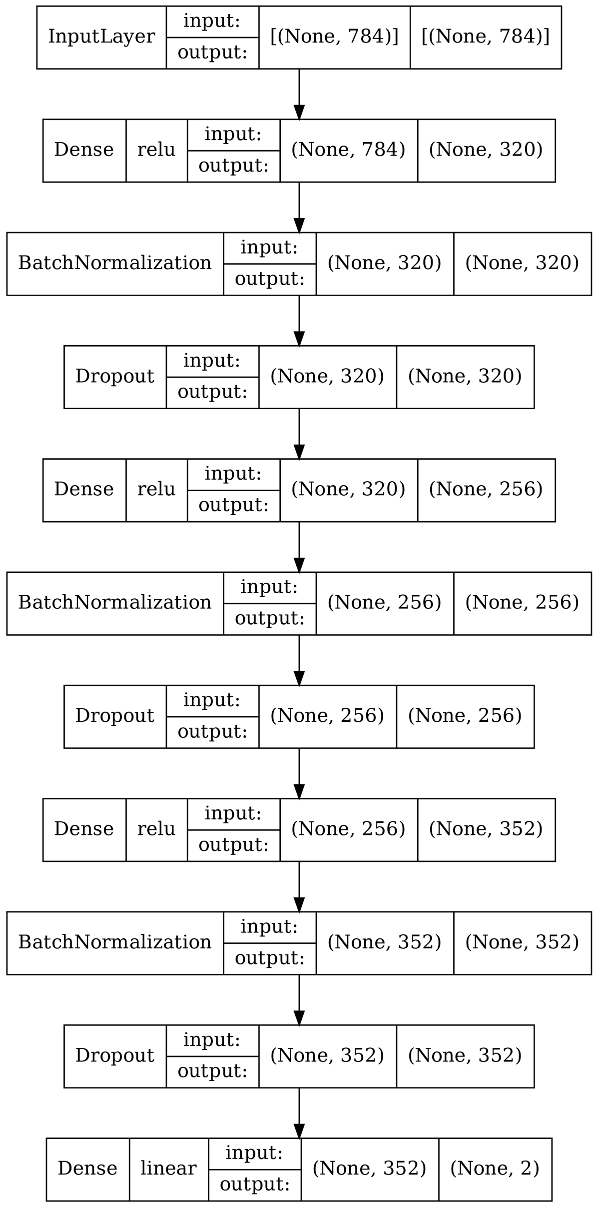

InputLayer
input:
output:
[(None, 784)]
[(None, 784)]
Dense
relu
input:
output:
(None, 784)
(None, 320)
BatchNormalization
input:
output:
(None, 320)
(None, 320)
Dropout
input:
output:
(None, 320)
(None, 320)
Dense
relu
input:
output:
(None, 320)
(None, 256)
BatchNormalization
input:
output:
(None, 256)
(None, 256)
Dropout
input:
output:
(None, 256)
(None, 256)
Dense
relu
input:
output:
(None, 256)
(None, 352)
BatchNormalization
input:
output:
(None, 352)
(None, 352)
Dropout
input:
output:
(None, 352)
(None, 352)
Dense
linear
input:
output:
(None, 352)
(None, 2)

Figure 36

References

[CNL11] Coates A., Ng A., Lee H.: An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (11–13 Apr 2011), vol. 15 of *Proceedings of Machine Learning Research*, pp. 215–223. URL: `https://proceedings.mlr.press/v15/coates11a.html`.

[GZZ05] Geng X., Zhan D.-C., Zhou Z.-H.: Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35*, 6 (2005), 1098–1107. `doi:10.1109/TSMCB.2005.850151`.

[JCC*11] Joia P., Coimbra D., Cuminato J. A., Paulovich F. V., Nonato L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. `doi:10.1109/TVCG.2011.220`.