A Feature Transformations

We report all feature transformations along with their sources and dimensionalities applied to computer vision and NLP datasets in Table 6 and Table 7 respectively. All transformations can be found using their name and the corresponding source[5].

Table 6: Feature transformations for images as features.

Transformation	Source	MNIST	CIFAR10	CIFAR100
Raw	-	✓	✓	✓
PCA (d=32)	scikit-learn	✓	✓	✓
PCA (d=64)	scikit-learn	✓	✓	✓
PCA (d=128)	scikit-learn	✓	✓	✓
NCA (d=64)	scikit-learn	✓	✓	✓
AlexNet(d=4096)	PyTorch-Hub	✓	✓	✓
GoogleNet (d=1024)	PyTorch-Hub	✓	✓	✓
VGG16 (d=4096)	PyTorch-Hub	✓	✓	✓
VGG19 (d=4096)	PyTorch-Hub	✓	✓	✓
ResNet50-V2 (d=2048)	TF-Hub	✓	✓	✓
ResNet101-V2 (d=2048)	TF-Hub	✓	✓	✓
ResNet152-V2 (d=2048)	TF-Hub	✓	✓	✓
InceptionV3 (d=2048)	TF-Hub	✓	✓	✓
EfficientNet-B0 (d=1280)	TF-Hub	✓	✓	✓
EfficientNet-B1 (d=1280)	TF-Hub	✓	✓	✓
EfficientNet-B2 (d=1408)	TF-Hub	✓	✓	✓
EfficientNet-B3 (d=1536)	TF-Hub	✓	✓	✓
EfficientNet-B4 (d=1792)	TF-Hub	✓	✓	✓
EfficientNet-B5 (d=2048)	TF-Hub	✓	✓	✓
EfficientNet-B6 (d=2304)	TF-Hub	✓	✓	✓
EfficientNet-B7 (d=2560)	TF-Hub	✓	✓	✓

Table 7: Feature transformations for natural language as features.

Transformation	Source	IMDB	SST2	YELP
BOW	scikit-learn	✓	✓	✗
BOW-TFIDF	scikit-learn	✓	✓	✗
PCA (d=8)	scikit-learn	✓	✓	✗
PCA (d=16)	scikit-learn	✓	✓	✗
PCA (d=32)	scikit-learn	✓	✓	✗
PCA (d=64)	scikit-learn	✓	✓	✗
PCA (d=128)	scikit-learn	✓	✓	✗
ELMO (d=1024)	TF-Hub	✓	✓	✓
NNLM-EN (d=50)	TF-Hub	✓	✓	✓
NNLM-EN-WITH-NORMALIZATION (d=50)	TF-Hub	✓	✓	✓
NNLM-EN (d=128)	TF-Hub	✓	✓	✓
NNLM-EN–WITH-NORMALIZATION (d=128)	TF-Hub	✓	✓	✓
Universal Sentence Encoder (USE) (d=512)	TF-Hub	✓	✓	✓
BERT-Base (d=678)	PyTorch-Hub	✓	✓	✓

[5]TensorFlow Hub: `https://tfhub.dev`, PyTorch Hub: `https://pytorch.org/hub`, and scikit-learn: `https://scikit-learn.org/`.

B Extended Results

B.1 No Transformation / Raw

To see the impact of choosing the best feature transformation per method, we report the results on the raw features (i.e., pixel values) for the vision datasets in Figure 5 and Table 8, nothing that NLP datasets do not come with any natural *raw* representations. For the method achieving good lower and upper bounds (i.e., kNN and GHP), we see that for both CIFAR variants, they constantly overestimate both the upper and lower bound. For the other methods, selecting the best hyper-parameters leads to non-informative variants.

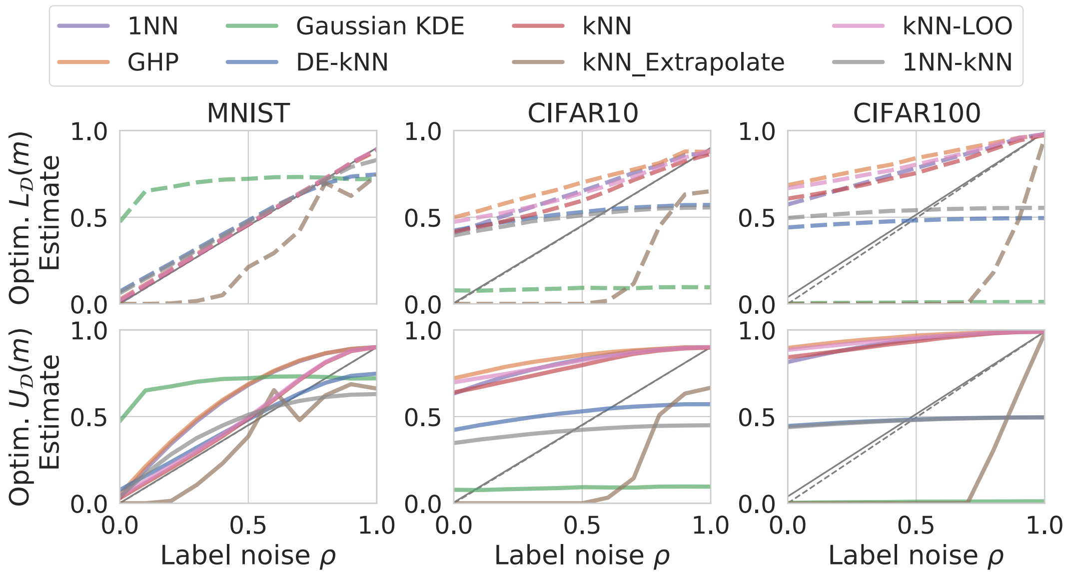


Figure 5: Computer vision datasets on raw features.

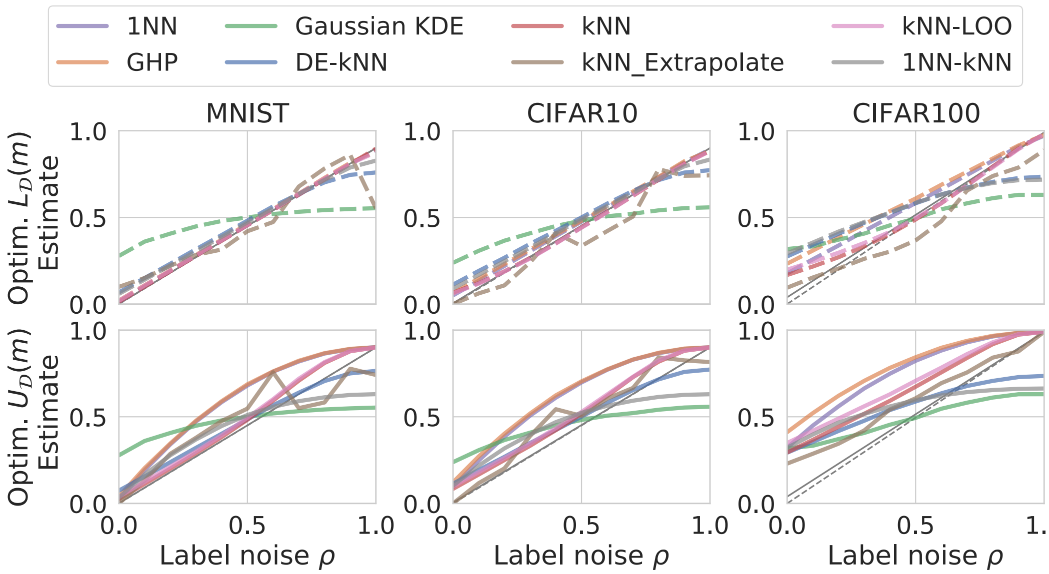
Figure 6: Evolution of the BER for selecting hyper-parameters only on the raw pixels that minimize **(top row, each)** $L_{\mathcal{D}}(m)$ and **(bottom row, each)** $U_{\mathcal{D}}(m)$.

Table 8: $L_{\mathcal{D}}(m)$ and $U_{\mathcal{D}}(m)$: The optimal values per method on raw pixel values.

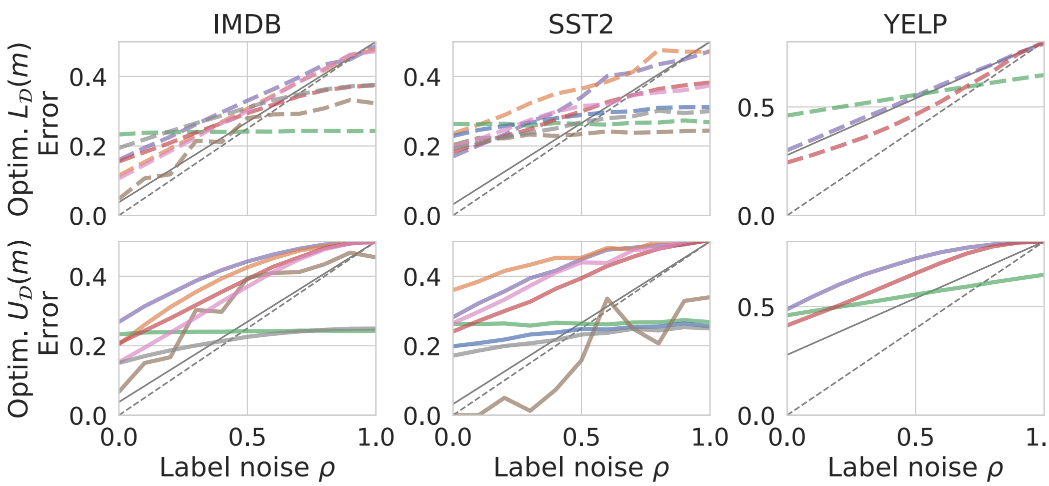
	DE-kNN	KDE	GHP	1NN-kNN	1NN	kNN	kNN-LOO	kNN_Ext
MNIST	0.12	0.64	0.03	0.07	0.02	0.02	0.03	0.50
CIFAR10	0.47	0.83	0.55	0.45	0.46	0.39	0.46	0.69
CIFAR100	0.49	0.98	0.66	0.49	0.54	0.53	0.61	0.71
MNIST	0.12	0.64	0.34	0.22	0.32	0.09	0.11	0.43
CIFAR10	0.47	0.83	0.85	0.48	0.79	0.75	0.80	0.66
CIFAR100	0.49	0.98	0.89	0.49	0.84	0.84	0.88	0.65

B.2 Optimal $L_{\mathcal{D}}(m)$ and $U_{\mathcal{D}}(m)$.

In this section we provide additional visualizations to support the findings in Section 4.2, completing Figure 2. We observe that 1NN, kNN, kNN-LOO and GHP closely follow the evolution of the BER and that for them the main contributions to $L_{\mathcal{D}}(m)$ and $U_{\mathcal{D}}(m)$ come from $L_{\mathcal{D},\triangleright}(m)$ and $U_{\mathcal{D},\triangleright}(m)$. This will be improved as more transformations become available, thus reducing the bias. For DE-kNN, KDE and kNN-Extrapolate we clearly see difficulties in following the evolution of the BER, often visualized in the bottom-right corner, resulting in large $L_{\mathcal{D},\triangleleft}(m)$ and/or $U_{\mathcal{D},\triangleleft}(m)$.



(a) Computer vision datasets.



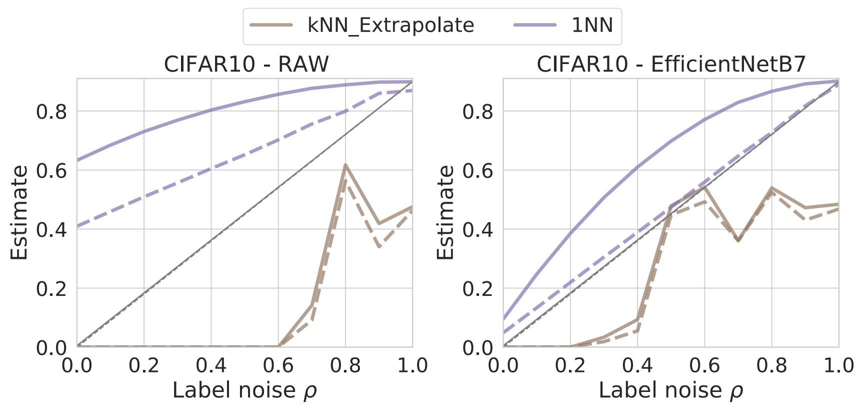
(b) NLP datasets.

Figure 7: Evolution of the BER for selecting hyper-parameters and transformations that minimize **(top row, each)** $L_{\mathcal{D}}(m)$ and **(bottom row, each)** $U_{\mathcal{D}}(m)$.

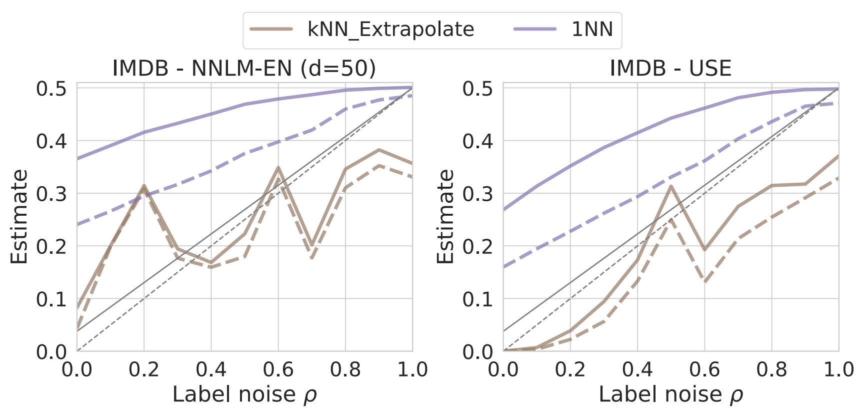
B.3 Further Example Plots

In this section we present example plots produced by `FeeBee`. For simplicity, we focus only on two datasets and two methods. In order to have comparable lower and upper bounds, we fix the transformation in each figure.

We observe that 1NN performs better than kNN-Extrapolate for the lower bound, clearly following the evolution of the BER. We also observe that for both the lower bound is always below upper bound.



(a)



(b)

Figure 8: Example plots of $L_{\mathcal{D}}(m)$ and $U_{\mathcal{D}}(m)$ for **(a)** CIFAR10 over raw and EfficientNetB7, and **(b)** IMDB over NNLM-EN ($d = 50$) and USE.

C Tables

In this section we provide tables with optimal values of scores $L_{\mathcal{D}}(m)$ and $U_{\mathcal{D}}(m)$, for every dataset and every method, reporting the hyper-parameters and transformation that yield that score, together with the upper ($L_{\mathcal{D},\triangleright}, U_{\mathcal{D},\triangleright}$) and lower areas ($L_{\mathcal{D},\triangleleft}, U_{\mathcal{D},\triangleleft}$) that contribute to the scores.

C.1 Optimal $L_{\mathcal{D}}(m)$.

Table 9: MNIST - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN	dist=cosine, k=2	NCA (d=64)	0.02	0.00	0.01
1NN	dist=cosine	NCA (d=64)	0.02	0.01	0.01
kNN-LOO	dist=cosine, k=2	NCA (d=64)	0.03	0.01	0.02
GHP	default	PCA (d=32)	0.03	0.00	0.02
1NN-kNN	dist=cosine, k=21	PCA (d=32)	0.07	0.02	0.05
DE-kNN	dist=squared_l2, k=15	PCA (d=32)	0.11	0.05	0.06
kNN_Extrapolate	dist=cosine, k=10	PCA (d=32)	0.28	0.15	0.13
Gaussian KDE	B=0.05	VGG16	0.41	0.18	0.23

Table 10: CIFAR10 - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN-LOO	dist=squared_l2, k=3	EfficientNet-B3	0.03	0.01	0.02
kNN	dist=squared_l2, k=3	EfficientNet-B2	0.03	0.01	0.02
1NN	dist=squared_l2	EfficientNet-B7	0.05	0.00	0.05
GHP	default	EfficientNet-B4	0.07	0.00	0.07
1NN-kNN	dist=squared_l2, k=22	EfficientNet-B7	0.10	0.02	0.08
DE-kNN	dist=squared_l2, k=18	EfficientNet-B7	0.14	0.04	0.11
kNN_Extrapolate	dist=squared_l2, k=3	PCA (d=32)	0.34	0.23	0.11
Gaussian KDE	B=0.1	ResNet152-V2	0.36	0.19	0.18

Table 11: CIFAR100 - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN	dist=cosine, k=5	EfficientNet-B5	0.06	0.01	0.05
kNN-LOO	dist=cosine, k=6	EfficientNet-B6	0.07	0.01	0.07
1NN	dist=cosine	EfficientNet-B7	0.14	0.00	0.14
GHP	default	EfficientNet-B7	0.20	0.00	0.20
kNN_Extrapolate	dist=squared_l2, k=1	PCA (d=32)	0.22	0.17	0.05
DE-kNN	dist=squared_l2, k=4	EfficientNet-B7	0.27	0.10	0.18
1NN-kNN	dist=squared_l2, k=12	EfficientNet-B7	0.29	0.10	0.19
Gaussian KDE	B=0.1	VGG19	0.31	0.18	0.14

Table 12: IMDB - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN-LOO	dist=squared_l2, k=1	USE	0.15	0.01	0.14
GHP	default	USE	0.16	0.01	0.16
kNN	dist=cosine, k=9	USE	0.25	0.09	0.16
1NN	dist=squared_l2	USE	0.25	0.01	0.24
kNN_Extrapolate	dist=squared_l2, k=6	PCA (d=16)	0.25	0.19	0.06
1NN-kNN	dist=cosine, k=4	PCA (d=32)	0.31	0.09	0.22
Gaussian KDE	B=0.0025	PCA (d=8)	0.49	0.29	0.20

Table 13: SST2 - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN	dist=cosine, k=10	BOW-TFIDF	0.29	0.08	0.20
1NN	dist=cosine	ELMO	0.32	0.02	0.30
kNN-LOO	dist=cosine, k=7	USE	0.34	0.10	0.24
1NN-kNN	dist=cosine, k=3	ELMO	0.38	0.20	0.18
DE-kNN	dist=squared_l2, k=4	NNLM-EN-NORM (d=128)	0.42	0.17	0.25
GHP	default	USE	0.44	0.01	0.42
kNN_Extrapolate	dist=squared_l2, k=2	PCA (d=8)	0.47	0.29	0.18
Gaussian KDE	B=0.0025	PCA (d=16)	0.49	0.24	0.25

Table 14: YELP - Optimal $L_{\mathcal{D}}(m)$

Method	Variant	Transformation	$L_{\mathcal{D}}(m)$	$L_{\mathcal{D},\triangleleft}(m)$	$L_{\mathcal{D},\triangleright}(m)$
kNN	dist=squared_l2, k=9	USE	0.00	0.00	0.00
1NN	dist=squared_l2	USE	0.03	0.00	0.03
Gaussian KDE	B=0.05	NNLM-EN (d=50)	0.20	0.06	0.14

C.2 Optimal $U_{\mathcal{D}}(m)$.

Table 15: MNIST - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
kNN	dist=cosine, k=10	NCA (d=64)	0.09	0.00	0.09
kNN-LOO	dist=cosine, k=10	NCA (d=64)	0.11	0.00	0.11
DE-kNN	dist=squared_l2, k=16	PCA (d=32)	0.11	0.04	0.07
1NN-kNN	dist=cosine, k=3	NCA (d=64)	0.22	0.12	0.09
1NN	dist=cosine	NCA (d=64)	0.32	0.00	0.32
GHP	default	PCA (d=32)	0.33	0.00	0.33
kNN_Extrapolate	dist=cosine, k=10	PCA (d=128)	0.37	0.16	0.21
Gaussian KDE	B=0.05	VGG16	0.41	0.18	0.23

Table 16: CIFAR10 - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
DE-kNN	dist=squared_l2, k=18	EfficientNet-B7	0.15	0.04	0.11
kNN	dist=cosine, k=10	EfficientNet-B7	0.15	0.00	0.15
kNN-LOO	dist=cosine, k=10	EfficientNet-B4	0.17	0.00	0.17
kNN_Extrapolate	dist=squared_l2, k=3	PCA (d=32)	0.20	0.04	0.16
1NN-kNN	dist=squared_l2, k=3	EfficientNet-B7	0.26	0.12	0.14
Gaussian KDE	B=0.1	ResNet152-V2	0.36	0.19	0.18
1NN	dist=cosine	EfficientNet-B7	0.37	0.00	0.37
GHP	default	EfficientNet-B4	0.39	0.00	0.39

Table 17: CIFAR100 - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
kNN_Extrapolate	dist=cosine, k=7	PCA (d=32)	0.21	0.02	0.19
DE-kNN	dist=squared_l2, k=4	EfficientNet-B6	0.28	0.09	0.19
Gaussian KDE	B=0.1	VGG19	0.31	0.18	0.14
kNN	dist=cosine, k=10	EfficientNet-B7	0.31	0.00	0.31
1NN-kNN	dist=squared_l2, k=3	EfficientNet-B7	0.36	0.14	0.22
kNN-LOO	dist=cosine, k=10	EfficientNet-B6	0.37	0.00	0.37
1NN	dist=cosine	EfficientNet-B7	0.49	0.00	0.49
GHP	default	EfficientNet-B7	0.55	0.00	0.55

Table 18: IMDB - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
kNN_Extrapolate	dist=squared_l2, k=6	PCA (d=16)	0.28	0.03	0.25
kNN-LOO	dist=squared_l2, k=9	USE	0.34	0.00	0.34
1NN-kNN	dist=cosine, k=2	PCA (d=32)	0.39	0.29	0.10
kNN	dist=squared_l2, k=9	USE	0.43	0.00	0.43
Gaussian KDE	B=0.0025	PCA (d=8)	0.49	0.29	0.20
GHP	default	USE	0.51	0.00	0.51
1NN	dist=squared_l2	USE	0.60	0.00	0.60

Table 19: SST2 - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
1NN-kNN	dist=cosine, k=2	NNLM-EN (d=128)	0.42	0.29	0.13
DE-kNN	dist=squared_l2, k=2	USE	0.43	0.27	0.17
kNN	dist=cosine, k=10	BOW-TFIDF	0.48	0.00	0.48
Gaussian KDE	B=0.0025	PCA (d=16)	0.49	0.24	0.25
kNN-LOO	dist=cosine, k=10	USE	0.57	0.00	0.57
kNN_Extrapolate	dist=squared_l2, k=2	USE	0.61	0.49	0.12
1NN	dist=cosine	ELMO	0.63	0.00	0.63
GHP	default	ELMO	0.73	0.00	0.73

Table 20: YELP - Optimal $U_{\mathcal{D}}(m)$

Method	Variant	Transformation	$U_{\mathcal{D}}(m)$	$U_{\mathcal{D},\triangleleft}(m)$	$U_{\mathcal{D},\triangleright}(m)$
Gaussian KDE	B=0.05	NNLM-EN (d=50)	0.20	0.06	0.14
kNN	dist=cosine, k=10	USE	0.25	0.00	0.24
1NN	dist=squared_l2	USE	0.38	0.00	0.38