A DATASET DETAIL

In this study, we use 25 public datasets mostly from the OpenML (Vanschoren et al., 2014) library, including the frequently used datasets in previous studies (Yoon et al., 2020; Ucar et al., 2021; Gorishniy et al., 2021; 2022). Each dataset has exactly one train-validation-test split, so all algorithms use the same splits as the previous studies (Gorishniy et al., 2021; 2022; Rubachev et al., 2022). We summarize the main properties of datasets in Table 5. For each dataset, we use a predefined batch size depending on the number of training samples: 64 when the number of training samples is less than 1000, 128 when the number of training samples is larger than 1000 and less than 5000, 256 when the number of training samples is larger than 5000 and less than 10000, 512 when the number of training samples is larger than 10000 and less than 50000, and 1024 when the number of training samples is larger than 50000.

We regard the feature as categorical when the number of unique values in the training dataset is less than 20 (5 for AL, MNIST, p-MNIST, MI). The categorical variables are fed into the feature tokenizer for FT-Transformer while MLP has no additional operation for them. For MNIST and p-MNIST datasets, we ignore the features that have only one possible value throughout the training dataset.

Table 5: Dataset summary.

Abbr.	Name	# Train	# Validation	# Test	# Num	# Cat	Task type	Batch size
CH	Churn Modeling [1]	6400	1600	2000	4	6	Binclass	256
HI	Higgs Small (Baldi et al., 2014)	62751	15688	19610	24	4	Binclass	1024
AD	Adult (Kohavi et al., 1996)	26048	6513	16281	2	12	Binclass	512
BM	Bank Marketing (Moro et al., 2011)	28934	7234	9043	7	9	Binclass	512
PH	Philippine (Guyon et al., 2019)	3732	933	1167	308	0	Binclass	128
OS	Online Shoppers (Sakar et al., 2019)	7891	1973	2466	8	9	Binclass	256
CS	German Credit dataset [2]	640	160	200	20	0	Binclass	64
PO	Phoneme	3458	865	1081	5	0	Binclass	128
CO	Covertype (Blackard & Dean, 1999)	371847	92962	116203	44	7	Multiclass	1024
OT	Otto Group Products [3]	39601	9901	12376	80	13	Multiclass	512
GE	Gesture Phase	6318	1580	1975	32	0	Multiclass	256
VO	Volkert [4] Guyon et al. (2019)	37318	9330	11662	147	33	Multiclass	512
WQ	Wine Quality (Cortez et al., 2009)	4157	1040	1300	11	0	Multiclass	128
AL	ALOI (Geusebroek et al., 2005)	69120	17280	21600	124	4	Multiclass	1024
HE	Helena (Guyon et al., 2019)	62752	15688	19610	27	0	Multiclass	512
MNIST	Handwritten Digit Images	50000	10000	10000	627	90	Multiclass	512
p-MNIST	Permuted MNIST	50000	10000	10000	627	90	Multiclass	512
CA	California Housing (Pace & Barry, 1997)	13209	3303	4128	8	0	Regression	512
HO	House 16H [5]	14581	3646	4557	16	0	Regression	512
FI	FIFA	12273	3069	3836	28	0	Regression	512
MI	MSLR-WEB10K(Fold 1) (Qin & Liu, 2013)	723412	235259	241521	131	5	Regression	1024
KI	Forward kinetics of an 8 link robot arm [6]	5242	1311	1639	8	0	Regression	256
CPU	Computer Activity Databases [7]	5242	1311	1639	8	0	Regression	256
DIA	Diamonds	34521	8631	10788	9	0	Regression	512
EL	Electricity [8]	24623	6156	7695	7	0	Regression	512

B IMPLEMENTATION DETAILS

We use the optimization strategy for SSL as follows. We do not tune any hyperparameter and the same configuration is applied to all cases.

- Optimizer: AdamW (Loshchilov & Hutter, 2017)
- Learning rate: 1e-4

[1] https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling

[2] https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data

[3] https://www.kaggle.com/c/otto-group-product-classification-challenge/data

[4] https://automl.chalearn.org/data

[5] http://www.ncc.up.pt/~ltorgo/Regression/DataSets.html

[6] http://www.ncc.up.pt/~ltorgo/Regression/DataSets.html

[7] http://www.ncc.up.pt/~ltorgo/Regression/DataSets.html

[8] https://github.com/LeoGrin/tabular-benchmark

- Weight decay: 1e-5
- Epochs: 1000
- Learning rate scheduler: Cosine annealing scheduler (Loshchilov & Hutter, 2016; Goyal et al., 2017)

For the hyperparameters related to SSL, we tried $p_m \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $T \in \{2, 5, 10, 20, 50, 100\}$. When we combine the transformation function and binning methods, to reduce the hyperparameter space, we tried $p_m \in \{0.1, 0.2, 0.3\}$ and $T \in \{2, 10\}$ for MLPs, and $p_m \in \{0.1, 0.2\}$ and $T \in \{2, 10\}$ for FT-Transformers. After SSL, we evaluate the pre-trained representations with the linear head. The linear head is trained with different random seeds 10 times, and the average performance is reported.

For other state-of-the-art models, we directly reference the reported performance in the papers to reduce the ambiguity from the random seeds or the tuning details.

B.1 MLP

For MLPs, we set the architecture when the validation performance is best under the supervised setup with the encoder network f_e after the grid search on the depth (1, 2, 3, 4, 5) and the width (128, 256, 512, 1024). The representation size is determined as identical to the width of MLPs. The following decoder network f_d is defined as symmetric with f_e.

For supervised learning, we use the same configuration of SSL summarized above, except that the learning rate is 0.001 and the number of epochs is 100. We summarize the best setups for all datasets as follows.

Table 6: MLP architectures.

Depth	Datasets	Width	Datasets
1	CH, HI, AD, BM, OS, FI, CS	128	CH, HI, AD, BM, OS, FI, MI, CA
2	MI, CPU, HE, OT, AL	256	CS, HE, KI, PH, HO
3	CA, KI, MNIST, EL	512	CPU, WQ, p-MNIST, DIA
4	WQ, p-MNIST, PH, HO, CO, GE, VO, PO	1024	CO, GE, VO, PO, MNIST, EL, OT, AL
5	DIA		

B.2 FT-Transformer

We do not conduct any hyperparameter tuning for FT-Transformer, and we use the default setup defined in Gorishniy et al. (2021) with the number of blocks as 3. For three large-scale datasets, such as MI, MNIST, and p-MNIST, we set the number of blocks as 1 because of the computational budget. For the representation size, we adopt the value found in MLP cases. For f_d, we use the MLP network whose architecture is the same as Table 6.

B.3 Linear evaluation and Fine-tuning

For linear evaluation, we use the same optimization configuration for SSL except for the learning rate of 0.01 for 100 epochs. For fine-tuning, we use the same setups of the supervised cases.

C Full results

Here, we present the comprehensive results from our manuscript, accompanied by standard deviations derived from 10 repetitions of the experiment.

Table 7: Full results of Table 1. We repeat the evaluation 10 times and the average and the standard deviations are provided.

(a) Binary classification

Masking	Masking value	Objective(s)	CH	HI	AD	BM	PH	OS	CS	PO
FALSE	-	ValueRecon	0.810±0.001	0.651±0.000	0.837±0.000	0.899±0.000	0.728±0.001	0.883±0.000	0.709±0.003	0.851±0.000
TRUE	Const.	MaskXent	0.807±0.001	0.672±0.000	0.836±0.000	0.899±0.000	0.715±0.000	0.893±0.000	0.708±0.004	0.845±0.000
TRUE	Const.	ValueRecon	0.810±0.000	0.653±0.000	0.839±0.000	0.900±0.000	0.734±0.001	0.884±0.000	0.718±0.002	0.849±0.001
TRUE	Const.	MaskXent+ValueRecon	0.817±0.001	0.669±0.000	0.835±0.000	0.900±0.000	0.724±0.001	0.877±0.000	0.706±0.000	0.837±0.002
TRUE	Random	MaskXent	0.814±0.000	0.681±0.000	0.843±0.000	**0.901**±0.000	0.710±0.000	0.883±0.000	0.706±0.000	0.853±0.000
TRUE	Random	ValueRecon	0.811±0.000	0.661±0.000	0.838±0.000	0.898±0.000	**0.738**±0.001	0.885±0.000	0.714±0.003	0.842±0.000
TRUE	Random	MaskXent+ValueRecon	0.804±0.001	0.647±0.000	0.826±0.000	0.899±0.000	0.715±0.003	0.879±0.001	0.713±0.003	0.861±0.001
FALSE	-	BinXent	0.817±0.001	0.683±0.000	0.845±0.000	**0.901**±0.000	0.732±0.001	0.886±0.000	**0.738**±0.000	0.851±0.001
FALSE	-	BinRecon	**0.823**±0.000	**0.687**±0.000	0.840±0.000	0.900±0.000	0.737±0.000	0.889±0.000	0.724±0.005	**0.865**±0.000
TRUE	Const.	BinRecon	0.820±0.000	0.672±0.000	0.843±0.000	0.899±0.000	0.730±0.000	**0.896**±0.000	0.718±0.004	0.858±0.000
TRUE	Random	BinRecon	0.819±0.001	0.682±0.000	**0.846**±0.000	0.898±0.000	0.735±0.000	0.894±0.000	0.718±0.004	0.858±0.000

(b) Multiclass classification

Masking	Masking value	Objective(s)	CO	OT	GE	VO	WQ	AL	HE	MNIST	p-MNIST
FALSE	-	ValueRecon	0.769±0.000	0.776±0.000	0.527±0.001	0.619±0.000	0.568±0.001	0.931±0.000	0.353±0.000	0.965±0.000	0.928±0.000
TRUE	Const.	MaskXent	0.784±0.000	0.777±0.000	0.518±0.001	0.545±0.000	0.547±0.000	0.909±0.001	0.341±0.000	0.793±0.000	0.554±0.000
TRUE	Const.	ValueRecon	0.783±0.000	0.791±0.000	0.557±0.001	0.622±0.000	0.586±0.001	0.931±0.000	0.354±0.000	0.966±0.000	0.925±0.000
TRUE	Const.	MaskXent+ValueRecon	0.750±0.000	0.774±0.001	0.519±0.005	0.610±0.000	0.571±0.001	0.931±0.000	0.360±0.000	0.941±0.000	0.907±0.000
TRUE	Random	MaskXent	0.763±0.000	0.791±0.000	0.555±0.000	0.549±0.000	0.544±0.001	0.925±0.000	0.336±0.000	0.945±0.000	0.817±0.000
TRUE	Random	ValueRecon	0.761±0.000	0.782±0.000	0.538±0.001	0.625±0.000	0.573±0.001	0.930±0.000	0.357±0.000	0.956±0.000	0.934±0.000
TRUE	Random	MaskXent+ValueRecon	0.769±0.000	0.779±0.001	0.521±0.004	0.564±0.001	0.519±0.004	0.925±0.001	0.353±0.001	0.945±0.000	0.906±0.001
FALSE	-	BinXent	0.742±0.000	0.781±0.000	0.517±0.001	0.600±0.001	0.565±0.001	0.903±0.001	0.354±0.000	0.956±0.000	0.908±0.000
FALSE	-	BinRecon	0.784±0.000	0.783±0.000	0.544±0.001	0.625±0.000	**0.592**±0.001	0.935±0.000	0.357±0.000	0.964±0.000	0.950±0.000
TRUE	Const.	BinRecon	0.812±0.000	0.792±0.000	0.559±0.001	0.647±0.000	0.581±0.001	0.943±0.000	0.359±0.000	0.974±0.000	0.964±0.000
TRUE	Random	BinRecon	**0.814**±0.000	**0.794**±0.000	**0.580**±0.000	**0.655**±0.000	0.574±0.001	**0.949**±0.000	**0.365**±0.000	**0.981**±0.000	**0.971**±0.000

(c) Regression

Masking	Masking value	Objective(s)	CA	HO	FI	MI	KI	CPU	DIA	EL
FALSE	-	ValueRecon	0.749±0.000	4.241±0.001	13900.720± 0.816	0.784±0.000	0.163±0.000	3.876±0.002	1016.641±0.191	0.399±0.001
TRUE	Const.	MaskXent	0.709±0.000	4.548±0.000	13473.750± 1.371	0.788±0.000	0.185±0.000	4.475±0.033	1259.744±1.066	0.396±0.000
TRUE	Const.	ValueRecon	0.693±0.000	4.086±0.000	13518.683± 0.936	0.778±0.000	0.160±0.000	3.728±0.003	952.444±0.130	0.394±0.000
TRUE	Const.	MaskXent+ValueRecon	0.700±0.001	4.157±0.045	13915.875±18.078	0.775±0.000	0.174±0.001	5.644±0.078	2797.034±191.324	0.398±0.001
TRUE	Random	MaskXent	0.677±0.000	4.297±0.000	13826.641± 0.624	0.782±0.000	0.176±0.000	3.951±0.001	1358.135±0.191	0.388±0.000
TRUE	Random	ValueRecon	0.713±0.000	4.127±0.000	13668.988± 1.262	0.777±0.000	0.162±0.000	3.760±0.002	986.306±0.359	0.396±0.000
TRUE	Random	MaskXent+ValueRecon	0.701±0.003	4.136±0.028	14107.645±29.125	0.780±0.001	0.166±0.001	4.506±0.034	1917.875±123.359	0.397±0.008
FALSE	-	BinXent	0.690±0.000	4.116±0.001	**13038.762**± 1.618	0.776±0.000	0.170±0.000	3.717±0.006	1207.923±2.552	0.383±0.000
FALSE	-	BinRecon	0.622±0.000	3.766±0.000	13453.309± 0.832	**0.767**±0.000	**0.158**±0.000	3.208±0.001	897.645±0.182	0.370±0.000
TRUE	Const.	BinRecon	0.634±0.000	3.765±0.000	13208.133± 1.960	0.773±0.000	**0.158**±0.000	**3.156**±0.002	957.801±0.070	0.371±0.000
TRUE	Random	BinRecon	**0.619**±0.000	**3.703**±0.000	13075.474± 0.800	0.773±0.000	0.160±0.000	3.183±0.001	**870.283**±0.093	**0.368**±0.000

Table 8: Full results of Table 2. We repeat the evaluation 10 times and the average and the standard deviations are provided.

(a) Binary classification

Method	CH	HI	AD	BM	PH	OS	CS	PO
MLP(Init.)	0.796±0.003	0.622±0.004	0.820±0.002	0.892±0.001	0.683±0.011	0.873±0.002	0.726±0.016	0.771±0.007
MLP(Supervised)	0.836±0.003	0.713±0.002	0.849±0.001	0.902±0.001	0.724±0.009	0.895±0.002	0.666±0.026	0.894±0.005
MLP(Ours)	0.823±0.000	0.687±0.000	0.846±0.000	0.901±0.000	0.736±0.001	0.896±0.000	0.738±0.000	0.865±0.000
MLP(Ours+FT)	0.841±0.002	0.716±0.001	0.854±0.001	0.905±0.001	0.738±0.009	0.895±0.002	0.686±0.006	0.892±0.004
FTT(Init.)	0.818±0.004	0.636±0.007	0.828±0.004	0.890±0.002	0.694±0.008	0.866±0.003	0.698±0.022	0.851±0.006
FTT(Supervised)	0.824±0.005	0.701±0.003	0.837±0.003	0.903±0.002	0.724±0.018	0.884±0.002	0.676±0.028	0.895±0.006
FTT(Ours)	0.836±0.000	0.670±0.000	0.853±0.000	0.899±0.000	0.725±0.000	0.887±0.000	0.725±0.000	0.867±0.001
FTT(Ours+FT)	0.835±0.004	0.700±0.002	0.839±0.002	0.903±0.003	0.734±0.004	0.882±0.004	0.688±0.022	0.898±0.007

(b) Multiclass classification

Method	CO	OT	GE	VO	WQ	AL	HE	MNIST	p-MNIST
MLP(Init.)	0.729±0.003	0.766±0.001	0.467±0.007	0.547±0.004	0.540±0.011	0.897±0.002	0.311±0.002	0.896±0.002	0.731±0.010
MLP(Supervised)	0.968±0.000	0.810±0.002	0.659±0.007	0.694±0.002	0.623±0.006	0.960±0.001	0.378±0.002	0.983±0.001	0.978±0.001
MLP(Ours)	0.814±0.000	0.794±0.000	0.580±0.000	0.655±0.000	0.592±0.001	0.949±0.000	0.365±0.000	0.981±0.000	0.971±0.000
MLP(Ours+FT)	0.969±0.000	0.814±0.002	0.675±0.009	0.724±0.002	0.630±0.006	0.963±0.001	0.385±0.002	0.986±0.000	0.982±0.001
FTT(Init.)	0.730±0.002	0.705±0.008	0.509±0.007	0.544±0.005	0.569±0.007	0.762±0.009	0.311±0.004	0.550±0.025	0.480±0.022
FTT(Supervised)	0.970±0.000	0.794±0.004	0.664±0.008	0.704±0.005	0.617±0.008	0.960±0.001	0.338±0.003	0.966±0.002	0.960±0.002
FTT(Ours)	0.762±0.000	0.780±0.000	0.554±0.000	0.614±0.000	0.562±0.001	0.930±0.001	0.364±0.000	0.931±0.000	0.883±0.000
FTT(Ours+FT)	0.971±0.000	0.793±0.002	0.698±0.006	0.720±0.003	0.623±0.006	0.961±0.001	0.342±0.001	0.978±0.001	0.966±0.001

(c) Regression

Method	CA	HO	FI	MI	KI	CPU	DIA	EL
MLP(Init.)	0.854±0.039	4.700±0.055	14241.610±241.419	0.781±0.002	0.192±0.003	5.564±0.205	1291.373±46.713	0.400±0.001
MLP(Supervised)	0.513±0.002	3.146±0.036	10086.080± 65.915	0.754±0.000	0.071±0.001	2.793±0.013	562.153± 2.598	0.354±0.001
MLP(Ours)	0.619±0.000	3.703±0.000	13038.762± 1.618	0.767±0.000	0.158±0.000	3.156±0.002	870.283± 0.093	0.368±0.000
MLP(Ours+FT)	0.502±0.002	3.026±0.037	9963.609± 23.173	0.753±0.000	0.071±0.001	2.801±0.016	550.717± 3.171	0.350±0.001
FTT(Init.)	0.690±0.021	4.107±0.039	16128.694±383.004	0.791±0.004	0.181±0.004	5.205±0.324	1021.200±30.271	0.394±0.002
FTT(Supervised)	0.487±0.007	3.319±0.067	10206.127±347.223	0.752±0.000	0.072±0.001	2.780±0.049	540.904± 3.627	0.350±0.001
FTT(Ours)	0.549±0.000	3.570±0.000	14557.626± 1.390	0.770±0.000	0.153±0.000	3.645±0.000	865.654± 0.005	0.371±0.000
FTT(Ours+FT)	0.477±0.002	3.173±0.024	9936.115±226.549	0.752±0.001	0.072±0.001	2.792±0.038	542.962± 3.207	0.343±0.001

D ADDITIONAL RESULTS FOR DISCUSSION

D.1 COMPARING THE BINNING METHOD BETWEEN THE QUANTILES AND THE EQUAL-WIDTH

We found that the grouping is critical for implementing the binning task successfully. Instead of quantile-based binning in our method, we also can manipulate equal-width binning. Here, we experiment with which method can be more beneficial for binning between the quantile and fixed size. We test the same candidates for the number of bins for equal-width binning, and we compare the test performance when the validation performance is the best with the quantile-based ones.

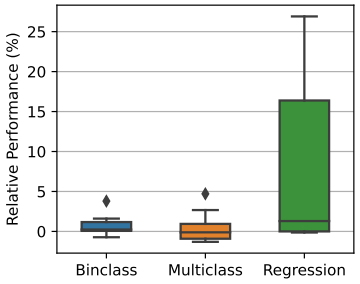

Figure 5: Relative performance when we change the binning method to the equal-width from the quantiles. When the values are positive, the quantile-based binning is better than the equal-width binning. When the values are negative, vice versa. In particular, for regression tasks, the quantile-based binning is much better than the equal-width binning.

The results are described in Figure 5. Among 25 datasets, equal-width binning showed better performance for three datasets (PH, HE, MNIST) to the extent of 0.6% at the maximum, and two binning methods showed comparable performance for two datasets (OT, AL). For the other 20 datasets, quantile-based binning showed better performance. In particular, for regression tasks, we found that the performance degrades 27% as the maximum when we change the binning method from quantile to fixed size. Finally, we conclude that quantile-based binning consistently results in good representations across various datasets.

D.2 BIN INFORMATION IS NOT USABLE UNLESS IT IS PROVIDED AS A PRETEXT TASK

Table 9: Binning regression task performance on various SSL methods. We provide the relative error with the baseline of the BinRecon case. For all cases, the error is increased by at least 38%. As a result, the binning indices are achievable from the raw inputs but not usable in the resulting representations when we do not explicitly provide as the pretext targets.

Masking	Masking value	Objective(s)	Relative error increase (%)
False	-	BinRecon	*(Baseline)* 0
False	-	ValueRecon	49.579
True	Const.	MaskXent	82.922
True	Const.	ValueRecon	38.444
True	Const.	MaskXent+ValueRecon	68.344
True	Random	MaskXent	111.708
True	Random	ValueRecon	38.135
True	Random	MaskXent+ValueRecon	60.016
Average			66.285

E ADDITIONAL RESULTS FOR REBUTTAL

Dear Reviewers,

We appreciate the significance of including the additional results. We aim to incorporate these experimental results in either the main text or the supplementary materials of the final manuscript. Due to page limitations, we might need to implement some modifications to include all the necessary content efficiently. Please refer to this section for an overview of the preliminary results.

Thank you for your attention to these details and for your valuable feedback.

E.1 ADDITIONAL RESULTS WITH OTHER BACKBONE ENCODERS

Our method is model-agnostic, allowing flexibility in choosing backbone encoders. So, we have expanded our experiments to include the ResNet (Gorishniy et al., 2021) and T2G-Former (Yan et al., 2023) model, in addition to our initial tests with MLPs and FT-Transformers. We chose the ResNet due to its distinctly different architectural design compared to MLPs and FT-Transformers, and T2G-Former due to its superior performance.

We provide the partial results in Table 10 in the same format as Table 2 in the manuscript. All the cases are repeated ten times, and the average performance is provided. We found that the binning task is effective with ResNet and T2G-Former backbone as well. These results further validate the adaptability and effectiveness of our method across diverse DNN architectures.

Table 10: Comparison with supervised baselines. We compare the downstream task performance under several scenarios: (1) Baseline-1, where the encoder is randomly initialized; (2) Baseline-2, where the encoder is trained by optimizing the supervised loss; (3) Ours-1, where the encoder is trained based on the binning task only; and (4) Ours-2, where the encoder is fine-tuned after the pre-training on binning tasks. (Notation: ↑ corresponds to accuracy, ↓ corresponds to RMSE)

Training method	CH ↑	AD ↑	PH ↑	OS ↑	CO ↑	OT ↑	GE ↑	VO ↑	HE ↑	MNIST ↑	CA ↓	HO ↓	FI ↓	EL ↓
Encoder = MLP														
Baseline-1	0.796	0.820	0.683	0.873	0.729	0.766	0.467	0.547	0.311	0.896	0.854	4.700	14241.610	0.400
Baseline-2	0.836	0.849	0.724	0.895	0.968	0.810	0.659	0.694	0.378	0.983	0.513	3.146	10086.080	0.354
Ours-1	0.823	0.846	0.736	**0.896**	0.814	0.794	0.580	0.655	0.365	0.981	0.619	3.703	13038.762	0.368
Ours-2	**0.841**	**0.854**	**0.738**	0.895	**0.969**	**0.814**	**0.675**	**0.724**	**0.385**	**0.986**	**0.502**	**3.026**	**9963.609**	**0.350**
Encoder = FT-Transformer														
Baseline-1	0.818	0.828	0.694	0.866	0.730	0.705	0.509	0.544	0.311	0.550	0.690	4.107	16128.694	0.394
Baseline-2	0.824	0.837	0.724	0.884	0.970	**0.794**	0.664	0.704	0.338	0.966	0.487	3.319	10206.127	0.350
Ours-1	**0.836**	**0.853**	0.725	**0.887**	0.762	0.780	0.554	0.614	**0.364**	0.931	0.549	3.570	14557.626	0.371
Ours-2	0.834	0.839	**0.734**	0.882	**0.971**	0.793	**0.698**	**0.720**	0.342	**0.978**	**0.477**	**3.173**	**9936.115**	**0.343**
Encoder = ResNet														
Baseline-1	0.792	0.757	0.580	0.839	0.505	0.348	0.396	0.342	0.116	0.234	1.071	5.126	19854.911	0.473
Baseline-2	0.822	0.843	0.714	0.888	0.729	0.740	0.489	0.547	0.228	0.821	0.688	3.931	10591.441	0.409
Ours-1	0.791	0.806	**0.716**	0.838	0.602	0.576	0.430	0.478	0.175	0.452	1.001	5.045	19276.360	0.455
Ours-2	**0.838**	**0.844**	0.715	**0.890**	**0.733**	**0.743**	**0.492**	**0.552**	**0.239**	**0.824**	**0.651**	**3.929**	**9843.289**	**0.408**
Encoder = T2G-Former														
Baseline-1	0.820	0.830	0.702	0.870	0.733	0.716	0.512	0.553	0.317	0.660	0.683	4.071	15991.264	0.393
Baseline-2	0.827	0.844	0.734	0.882	0.966	0.810	0.682	0.719	0.351	0.985	0.492	3.296	10551.509	0.348
Ours-1	**0.834**	**0.853**	0.730	**0.891**	0.764	0.775	0.556	0.617	**0.363**	0.951	0.570	3.540	15138.672	0.375
Ours-2	0.832	0.847	**0.747**	0.883	**0.968**	**0.816**	**0.713**	**0.728**	0.353	**0.985**	**0.479**	**3.257**	**10497.660**	**0.342**

E.2 COMPARISON WITH ADDITIONAL SUPERVISED METHODS PRE-TRAINED WITH OTHER PRETEXT TASKS

We will provide the additional results in Table 2 with the baselines, fine-tuned from the pre-trained weights on the other pretext tasks listed in Table 1. In this case, the hyperparameters are fixed, and only the pretraining objectives are different, so we can attribute the performance gain of our method to changing the pretext task as reconstructing the bin indices instead of the raw values. As shown in Table 11, the binning loss outperforms the others in most cases. Please note that even in cases where it did not yield the best performance, the binning loss demonstrated comparable results to the best-performing methods. Only the binning task consistently performs well among the four pretext tasks for all datasets.

Table 11: Comparison with additional supervised methods pre-trained with other pretext tasks. In this case, the models are pretrained with the pretraining loss denoted in the first column; then they are fine-tuned in a supervised fashion. All the architecture and optimization-related hyperparameters are fixed for each dataset.

Pretraining loss	CH ↑	AD ↑	PH ↑	PO ↑	CO ↑	OT ↑	GE ↑	VO ↑	HE ↑	MNIST ↑	CA ↓	HO ↓	FI ↓	CPU ↓
Encoder = MLP														
MaskXent	0.841	0.851	0.737	0.894	0.970	0.811	0.678	0.720	0.385	0.984	**0.499**	3.086	10183.670	2.850
ValueRecon	0.837	0.849	0.719	0.890	0.969	0.810	0.669	0.707	0.381	0.984	0.511	3.119	10374.178	2.825
MaskXent+ValueRecon	0.836	0.848	0.726	0.889	0.968	0.809	0.655	0.697	0.382	0.983	0.547	3.361	11842.398	2.871
BinRecon	**0.842**	**0.854**	**0.738**	**0.897**	**0.970**	**0.814**	**0.680**	**0.724**	**0.386**	**0.986**	0.500	**3.026**	**9963.609**	**2.791**
Encoder = FT-Transformer														
MaskXent	0.821	0.838	0.726	0.900	0.970	**0.797**	0.682	0.690	0.335	0.973	0.479	3.196	10417.374	2.857
ValueRecon	0.821	**0.840**	0.725	0.894	0.958	0.781	0.646	0.684	0.331	**0.979**	0.483	3.301	10527.885	2.769
MaskXent+ValueRecon	0.824	0.839	0.735	0.899	0.960	0.785	0.658	0.689	0.331	**0.979**	0.486	3.257	10708.780	2.762
BinRecon	**0.835**	0.839	**0.736**	**0.902**	**0.971**	0.794	**0.698**	**0.720**	**0.342**	**0.979**	**0.477**	**3.173**	**9936.115**	**2.753**

E.3 VISUALIZATION ANALYSIS

In the manuscript, we provide some discussion based on the experimental results related to the benefits of binning as a pretext task in Section 6. In summary, we found (1) grouping similar values is the most critical factor for the successful implementation of binning (Section 6.1), and (2) bin information is not usable unless it is provided as a pretext task (Section 6.3). To enhance the interpretability of our analysis, we provide the visualization results.

To show that the binning loss can effectively encourage grouping factor, i.e., to make the similar values (i.e., in the same bins) closer to each other and separated from other values, we visualize the representation vectors after SSL with the different pretext task of ValueRecon(Figure 6 Left) and BinRecon(Figure 6 Reft) in the case of HO dataset. Because the representation vectors are high-dimensional, we implement PCA for better interpretability. In Figure 6, we denote the samples in the smallest bin as gray and the samples in the largest bin as orange. As a result, we found that the representation vectors are grouped only when we utilize the BinRecon loss. (The discussion will be added in Section 6 with a more detailed explanation.)

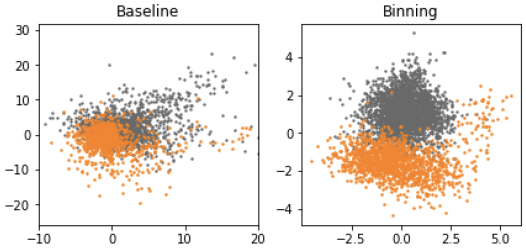

Figure 6: Visualization analysis