One Model for All Columns: Unified Learning for Tabular Data

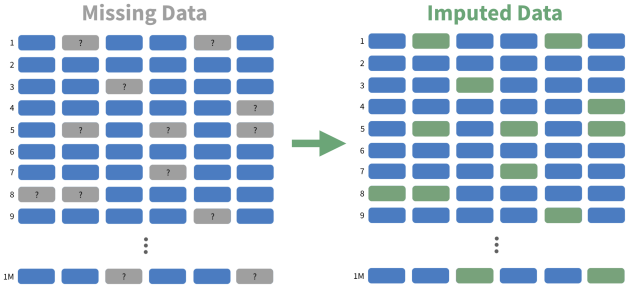
DLCP2026: The 10th International Conference on Deep Learning in Computational Physics

Kirill Katsuba · Roman Degtiarev · Nikita Sevriukov · Aziz Temirkhanov · Mikhail Hushchyn

HSE University

Task

- A system is observed as a table: each row is one state, each column is one measured variable.
- The goal is to solve several prediction tasks.
- We want a model of inter-feature dependencies that can answer different conditional queries.



Predict missing values by exploiting dependencies between observed columns.

Problem Setting

Data

$$\mathcal{D} = \{x^{(n)}\}_{n=1}^{M}, \qquad x = (x_1, \ldots, x_d)$$

Columns can be numerical or categorical. Each row is one observed state of the system.

Goal

Learn one model of inter-feature dependencies:

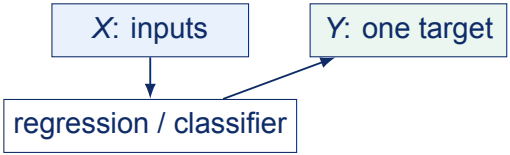
$$p_\theta(x) \quad \text{or} \quad p_\theta(x_U \mid x_O),$$

where O is any observed subset and U is any target subset.

Column-wise evaluation: $x_U = x_j, \quad x_O = x_{\setminus j}, \quad p_\theta(x_j \mid x_{\setminus j})$

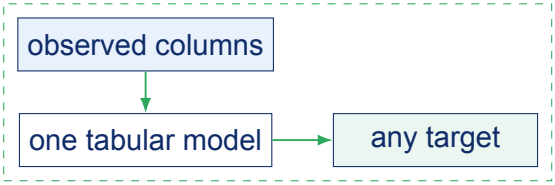
From Target-Specific to Unified Prediction

Conventional pipeline



- Split columns into fixed *X* and *Y*.
- Train a supervised model for this target.
- Repeat when *Y* changes.

Unified pipeline



- Treat target columns as missing values.
- Condition on any observed subset.
- Reuse the same model without retraining.

Compared Approaches

- **Mean/Mode**: fill each column separately with its empirical mean or mode.
- **Linear/Logistic**: one supervised model per target column.
- **CatBoost**: strong gradient boosting baseline, also trained separately for each target column.
- **TabImputer**: our discriminative imputation model; regression heads for numerical columns and classification heads for categorical columns.
- **Diffusion models**: generative tabular models used for conditional imputation.

TabImputer

- Encode numerical and categorical features.
- Add a mask token for missing columns.
- Use one shared backbone for all targets.
- Decode each column with a type-specific head.

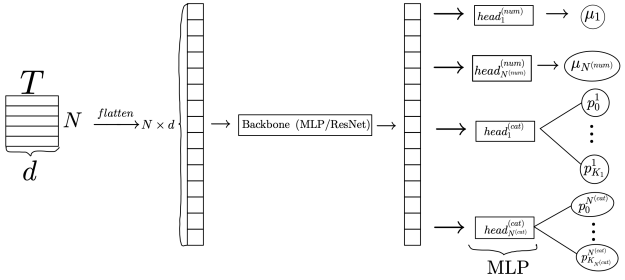


Diagram notation: *T* is the encoded table, *N* is the number of columns, and *d* is the token dimension.

Prediction heads

Numerical columns: regression.
Categorical columns: classification.

TabImputer Heads

Shared representation

$$h = f_\theta(x_{\text{obs}}, m)$$

The mask tells the model which values are observed ($m = 0$) and which column should be reconstructed ($m = 1$).

Column-specific heads

$$\hat{x}_i = a_i^\top h + b_i, \qquad i \in \mathcal{N}$$

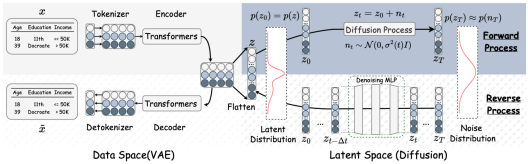
$$\hat{p}_i = \text{softmax}(W_i h + b_i), \qquad i \in \mathcal{C}$$

- $i \in \mathcal{N}$: numerical regression.
- $i \in \mathcal{C}$: categorical classification.

Diffusion Models

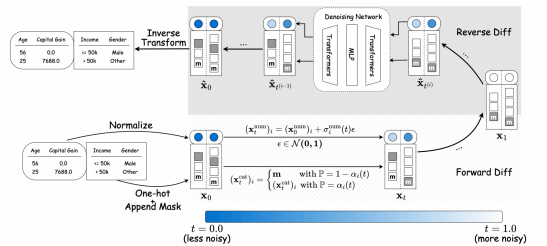
TabSyn

Zhang et al., ICLR 2024.



TabDiff

Shi et al., ICLR 2025.



TabCSDI: Zheng and Charoenphakdee, 2022.

Experimental Protocol

Datasets

- Adult
- Beijing
- Default
- Magic
- Shoppers

LOCO protocol

For every target column j, hide it and predict it from all remaining columns:

$$x_j \leftarrow \text{missing}, \qquad \hat{x}_j \sim p_\theta(x_j \mid x_{\setminus j})$$

Metrics

RMSE and MAE for numerical columns; Accuracy, F1 and Recall for categorical columns.

LOCO Results

Model	RMSE ↓	MAE ↓	Acc ↑	F1 ↑	Recall ↑
Mean/Mode	0.9774	0.6246	0.4483	0.1399	0.2294
Linear/Logistic	0.6880	0.4078	0.6133	0.3972	0.3995
CatBoost	**0.5578**	0.3020	0.7068	0.5261	0.5234
TabImputer	0.5541	**0.2974**	**0.7907**	**0.6158**	**0.5769**
TabCSDI	2.1298	1.5686	0.5364	0.3598	0.3590
TabSyn	0.6962	0.3418	0.5621	0.4128	0.4066
TabDiff	0.8708	0.4322	0.5806	0.4448	0.4452

Takeaway: one shared TabImputer is competitive with CatBoost on numerical targets and stronger on categorical targets.

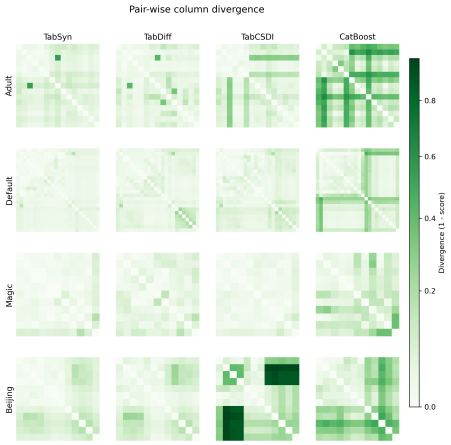
Prediction vs Distribution Quality

Model	α-Prec.	β-Recall	C2ST
TabCSDI	0.768	0.316	0.697
TabSyn	**0.945**	**0.383**	0.906
TabDiff	0.931	0.312	**0.925**
CatBoost	0.692	0.223	0.557

Takeaway:

Diffusion models preserve distributions.
CatBoost predicts averages/classes.

Metrics follow TabSyn (Zhang et al., ICLR 2024).



Practical Conclusions

- Modeling a table as a joint system enables arbitrary-column prediction.
- Target-specific baselines remain strong, but they require a separate model for each target.
- A shared discriminative imputer can be a practical alternative when many target columns are needed.
- Diffusion models better preserve distributional structure, but this does not automatically give the best point predictions.
- The useful design principle is explicit treatment of mixed feature types and missingness masks.

Thank you

References

- S. Zheng and N. Charoenphakdee. Diffusion Models for Missing Value Imputation in Tabular Data. arXiv, 2022.
- H. Zhang et al. Mixed-Type Tabular Data Synthesis with Score-Based Diffusion in Latent Space. ICLR, 2024.
- J. Shi et al. TabDiff: A Mixed-Type Diffusion Model for Tabular Data Generation. ICLR, 2025.
- L. Prokhorenkova et al. CatBoost: Unbiased Boosting with Categorical Features. NeurIPS, 2018.
- Y. Gorishniy et al. On Embeddings for Numerical Features in Tabular Deep Learning. NeurIPS, 2022.
- E. Telesheva and M. Hushchyn. Diffusion Models for Synthetic Tabular Data Generation. Dokl. Math. 112, 581–591, 2025. https://doi.org/10.1134/S1064562425700553.

Acknowledgement

The talk was prepared within the framework of the project “Mirror Laboratories” HSE University.

Appendix: Unified Objective

- Let $x = (x_1, \ldots, x_N)$, where columns are numerical or categorical.
- A binary mask $m \in \{0,1\}^N$ defines observed and hidden features.
- Training samples random masks and reconstructs hidden cells.

$x_o = \{x_i : m_i = 1\}, \qquad x_u = \{x_i : m_i = 0\}$

Training criterion

$$\max_\theta \mathbb{E}_{x,m} \log p_\theta(x_u \mid x_o, m)$$

$$\mathcal{L}(\theta) = \mathbb{E}_{x,m} \sum_{i:m_i=0} \ell_i(x_i, \hat{p}_{\theta,i})$$

LOCO

$$m_j = 0, \quad m_i = 1 \ (i \neq j) \Rightarrow p_\theta(x_j \mid x_{\setminus j})$$

Appendix: Mixed-Type Likelihood

Numerical columns

$$p_\theta(x_i \mid c) = \mathcal{N}\big(x_i; \mu_i(c), \sigma_i^2\big)$$

$$\ell_i^{\text{num}} = -\log p_\theta(x_i \mid c) \propto \frac{(x_i - \mu_i(c))^2}{2\sigma_i^2}$$

Categorical columns

$$p_\theta(x_i = k \mid c) = \text{softmax}(g_i(c))_k$$

$$\ell_i^{\text{cat}} = -\log p_\theta(x_i \mid c)$$

$$\mathcal{L} = \lambda_{\text{num}} \sum_{i \in \mathcal{N}} \ell_i^{\text{num}} + \lambda_{\text{cat}} \sum_{i \in \mathcal{C}} \ell_i^{\text{cat}}$$

Appendix: Diffusion Conditional Prediction

Forward noising

$$z_0 = E_\varphi(x), \qquad q(z_t \mid z_0) = \mathcal{N}\left(\alpha_t z_0, \sigma_t^2 I\right)$$

$$\epsilon_\theta(z_t, t, c) \approx \epsilon, \qquad c = (z_{\text{obs}}, m)$$

Conditional reverse process

$$p_\theta(z_{t-1}^u \mid z_t^u, z_o, m)$$

$$z_0^u \sim p_\theta(z_u \mid z_o, m), \qquad \hat{x}_u = D_\varphi(z_0^u)$$

$$z_t^{(i)} = \alpha_t z_0^{(i)} + a_i \sigma_t \epsilon_i$$

Appendix: MCAR Results

Model	ΔRMSE	ΔMAE	ΔAcc	ΔF1	ΔPrec	ΔRecall
Linear	14.93	15.44	17.22	129.46	230.66	41.16
CatBoost	-1.49	17.94	33.38	194.95	336.46	88.15
TabImputer	**38.36**	**48.56**	**67.36**	**300.42**	**495.37**	**128.38**

Values are average relative improvements over Mean/Mode across five datasets.

Appendix: Modified TabSyn

Model	RMSE ↓	MAE ↓	Acc ↑	F1 ↑	Recall ↑
CatBoost	0.5578	0.3020	0.7068	0.5261	0.5234
TabSyn	**0.6962**	0.3418	0.5621	0.4128	0.4066
TabSyn-MLP	0.7678	0.3624	0.5942	0.4485	0.4454
TabSyn-DBNS	0.7486	0.3447	**0.6093**	**0.4555**	**0.4535**
TabSyn-LS	0.7215	**0.3296**	0.5868	0.4356	0.4329

Appendix: Distribution Metrics

- α-Precision estimates how much generated data lies in high-density regions of real data.
- β-Recall estimates how much of the real data support is covered by generated samples.
- C2ST trains a classifier to distinguish real and generated rows.
- C2ST near chance indicates close distributions; very high values indicate easy separation.