Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9035

Retraction: Unlearning in Tabular-to-Hypergraph Learning via Selective Distillation

Rongxing Zhu
College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
Email: 1687462030@qq.com

How to cite this paper: Zhu, R. X. (2025). Unlearning in Tabular-to-Hypergraph Learning via Selective Distillation. *Journal of Computer Science and Frontier Technologies*, 2(1), 100-107. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9035
Published: 2025-12-23

Abstract

Machine unlearning requires efficiently removing the influence of specified information (rows/columns/values) from a deployed model to satisfy privacy and regulatory constraints. Converting tables into hypergraphs can capture high-order relations such as "same column", "same value", and multi-column interactions, but unlearning for hypergraph-based models is often implemented via costly retraining or gradient/Hessian-based approximations, and becomes especially challenging for column- and value-level deletion due to broad structural dependencies and limited verifiability. We propose HERMES, a fundamentally different unlearning paradigm for tabular-to-hypergraph modeling based on selective knowledge transfer rather than parameter rollback or second-order correction. HERMES freezes the original trained model as a teacher and trains a student as the released unlearned model: on the retained set, the student learns from the teacher through structure-augmented distillation and consistency regularization to preserve utility; on the forget set, the student is driven toward maximum-entropy predictions and explicitly pushed away from the teacher via anti-distillation divergence, actively erasing memorized behaviors that can be exploited by membership inference. For column/value deletion, HERMES introduces counterfactual invariance by replacing deleted attributes/values and enforcing prediction consistency, preventing the student from relying on prohibited information even through hypergraph message-passing "detours". The framework requires neither graph partitioning nor second-order information, and can be implemented as a lightweight post-processing stage that completes in a few epochs while offering practical verification signals based on forget-set entropy, teacher–student disagreement, and membership inference risk.

Keywords

Tabular Learning; Hypergraph Modeling; Machine Unlearning; Knowledge Distillation; Counterfactual Invariance; Membership Inference

1. Introduction

Tabular data is prevalent in high-stakes domains such as finance, healthcare, and remote sensing, where privacy regulations increasingly require the ability to "delete

and forget" specific training information. Recent work has shown that converting tabular records into graph/hypergraph structures can improve predictive performance by modeling higher-order relations (e.g., shared values, column-level coupling, and interactions). However, unlearning in hypergraph-based models is more complex than standard IID learning: removing a single row affects its incident hyperedges; removing a column can alter a large portion of the representational pathways; and forgetting specific values (e.g., sensitive categories or buckets) requires the model to stop exploiting them.

Most existing unlearning solutions rely on retraining or on parameter-level approximations (often gradient- or second-order based) that may require careful tuning and are difficult to validate—particularly for column/value deletion where the influence can spread through the hypergraph.

We propose a completely different perspective: treat unlearning as selective knowledge transfer rather than parameter correction. Instead of "editing" the original model, we freeze it as a teacher and train a new student model that (i) inherits knowledge only from the retained data and (ii) explicitly suppresses and diverges from the teacher's behavior on the forget set. This yields an auditable and practical unlearning procedure.

Contributions.

- We introduce HERMES, a selective distillation unlearning framework for tabular-to-hypergraph modeling that does not require graph partitioning, influence functions, or second-order statistics.
- We formulate a unified objective for row/column/value deletion using retained-set distillation, forget-set maximum entropy, anti-distillation divergence, and counterfactual invariance.
- We propose structure-aware augmentations to stabilize distillation and mitigate information leakage through hypergraph message passing.
- We provide practical unlearning verification signals, including forget-set entropy, teacher‑student disagreement, and membership inference risk.

2. Hypergraph Modeling for Tabular Data

2.1. Hypergraph Construction

Given a tabular dataset$D = (x_i, y_i)_{i=1}^{N}$with$x_i \in \mathbb{R}^C$,, we construct a hypergraph $H = (V, E)$ to encode high-order relations. Nodes may represent rows, columns, and/or value/bucket entities, while hyperedges capture relations such as:

- Column hyperedges that link entities associated with the same column,
- Value hyperedges that connect samples sharing the same categorical value or falling into the same numerical bucket,
- Interaction hyperedges (optional) that encode multi-column patterns.

Importantly, HERMES is model-agnostic: it can be applied on top of any tabular-to-hypergraph model (e.g., HGNN-style, AllSet-style, or transformer-like hyper-

graph encoders) because it only modifies the unlearning procedure, not the underlying architecture.

2.2. Deletion Requests

We consider three unlearning request types:

- Row unlearning: forget a set of records $F \subset D$,
- Column unlearning: delete one or more columns $J \subset \{1, \dots, C\}$,
- Value unlearning: delete sensitive values/buckets (e.g., a pair (j, k) denoting a value/bucket k in column j).

The goal is to produce an updated model f_{unl} whose behavior approximates retraining on the retained data, while reducing leakage about the forgotten information.

3. Proposed Method: HERMES

HERMES maintains two models:

- A teacher model $T(\cdot)$ trained on the full dataset and then frozen;
- A student model $S(\cdot)$ trained after receiving a deletion request and released as the unlearned model.

The key idea is to transfer knowledge selectively: preserve performance by distilling from the teacher on the retained set, while actively erasing knowledge on the forget set by enforcing maximum-entropy predictions and explicit divergence from the teacher.

3.1. Selective Retained-Set Distillation

Let the retained set be $R = D \backslash F$. We train the student to match the teacher on R.

(a).Supervised loss (optional):

$$L_{\text{sup}}(S; R)$$

using true labels on retained data.

(b).Distillation loss:

Let $z_T(x)$ and $\text{z}_\text{S}(\text{x})$ be teacher/student logits, and τ be the temperature:

$$L_{\text{KD}} = \mathbb{E}_{(x,y)\in R}\, \text{KL}(\sigma(z_T(x)/\tau) \| \sigma(z_S(x)/\tau))$$

(c) Structure-augmented consistency:

We apply hypergraph augmentations (e.g., hyperedge dropout, node masking, neighbor subsampling) to obtain $\text{aug}_1(x)$and $\text{aug}_2(\text{x})$, and enforce consistent predictions:

$$L_{cons} = \mathbb{E}_{x\in R} \| p_S(\text{aug}_1(x)) - p_S(\text{aug}_2(x)) \|_2^2$$

Retain objective:

$$L_{retain} = L_{sup} + \alpha L_{\text{KD}} + \beta L_{cons}$$

3.2. Forgetting via Maximum Entropy and Anti-Distillation

On the forget set F, we explicitly prevent the student from inheriting the teacher's

behavior.

(a) Maximum-entropy forgetting (uninformative outputs):

Let $p_s(x)$ be the student predictive distribution. Maximizing entropy is equivalent to minimizing negative entropy:

$$L_{\mathrm{ME}} = \mathbb{E}_{x \in F} \sum_{k=1}^{K} p_S^k(x) \log p_S^k(x)$$

Minimizing L_{ME}drives $p_s(x)$toward a uniform distribution, discouraging memorization.

(b) Anti-distillation divergence (explicitly moving away from the teacher):

We enforce a minimum divergence between teacher and student on the forget set using a margin form:

$$L_{\mathrm{AD}} = \mathbb{E}_{x \in F} \max(0, m - \mathrm{KL}(p_T(x) \| p_S(x)))$$

where $m > 0$ is a divergence margin. This ensures the student does not replicate the teacher's responses on forgotten samples.

Forget objective:

$$L_{forget} = \gamma L_{\mathrm{ME}} + \delta L_{\mathrm{AD}}$$

3.3. Column/Value Deletion via Counterfactual Invariance

Row-level constraints alone may not eliminate reliance on deleted attributes. For column/value deletion, we impose counterfactual invariance: predictions should remain unchanged when deleted attributes/values are replaced.

Column deletion

For deleted columns J, we create counterfactual inputs x^{cf} by randomly replacing the values in columns J (using a prior or the empirical distribution). We enforce:

$$L_{\mathrm{CF}-col} = \mathbb{E}_{x \in R} \| ps(x) - ps(x^{cf}) \|_2^2$$

This reduces dependence on the deleted columns.

Value deletion

For deleting a sensitive value/bucket (j, k), we construct counterfactuals for records containing that value by replacing it with alternative values/buckets:

$$L_{\mathrm{CF}-val} = \mathbb{E}_{x \in R: x_j = k} \| p_S(x) - p_S(x^{cf}) \|^2$$

Counterfactual objective:

$$L_{\mathrm{CF}} = \eta L_{\mathrm{CF}-col} + \mu L_{\mathrm{CF}-val}$$

3.4. Final Training Objective

The student is trained with the unified objective:

$$L_{\mathrm{total}} = L_{retain} + L_{forget} + L_{\mathrm{CF}}$$

Practical training choices.

Initialization: student can be initialized from teacher (faster) or randomly (potentially stronger forgetting).

Batching: each iteration samples from both R and F; counterfactuals are generated on the fly for column/value deletion.

Stopping: early stopping can be based on retained-set validation (utility) jointly with

forget-set indicators (entropy/disagreement).

3.5. Unlearning Verification

HERMES provides practical, model-agnostic verification signals:

Forget-set predictive entropy: entropy on F should increase toward uniform, indicating reduced confidence/memorization.

Teacher–student disagreement: $\mathrm{KL}(p_T \| p_S)$ should be significantly larger on F than on R.

Membership inference (optional): using F as members and held-out data as non-members, MIA AUC close to 0.5 indicates reduced membership signal.

4. Experiments

4.1. Datasets

We conduct experiments on three tabular benchmarks: Adult (binary classification), Bank Marketing (binary classification), and Covertype (multi-class classification). Each dataset is split into train/validation/test sets using a standard protocol. Continuous features are normalized, and categorical features are represented by embeddings. For constructing value-level relations in the hypergraph, continuous columns are discretized into quantile-based buckets.

4.2. Experimental Setup

We first train a teacher tabular-to-hypergraph model on the full training data and keep the best checkpoint according to validation performance. After receiving an unlearning request, we train a student model and release it as the unlearned model.

The student is optimized with three goals:

- Retain utility: match the teacher on the retained data via distillation (and optionally supervised learning).
- Forget targets: on the forget set, encourage uninformative outputs (high-entropy predictions) and explicitly avoid reproducing the teacher's behavior.
- Column/value deletion (if applicable): enforce prediction invariance under counterfactual replacement of deleted columns/values.

In practice, unlearning is run for only a few epochs, which is much faster than retraining from scratch.

4.3. Baselines and Metrics

We compare Retrain-from-scratch (retraining the same model on the retained data as a reference), Retain-only fine-tuning (fine-tuning the teacher using only the retained data), and HERMES (ours) (training a student model with the HERMES objective). We report utility in terms of test accuracy (and macro-F1 for Covertype when

needed), a forgetting indicator measured by the predictive entropy on the forget set (higher entropy indicates the model is less confident on forgotten samples), and efficiency measured by the unlearning time relative to retraining.

4.4. Results and Discussion

Tables 1–3 summarize the main results of our experiments. Overall, HERMES achieves a favorable balance between predictive utility and unlearning effectiveness, while requiring substantially less time than full retraining. We observe consistent trends across the three datasets and the compared baselines.

Utility on the test set.

As shown in Table 1, the retrain-from-scratch baseline provides the reference performance after deletion. Compared with retraining, retain-only fine-tuning and distillation-only generally preserve test accuracy (and F1 when reported) but do not explicitly enforce forgetting. In contrast, forget-only tends to reduce test utility because it focuses on disrupting behavior on the forget set without sufficient constraints to maintain performance on retained data. HERMES maintains test performance close to the retrain baseline, and in several cases matches or slightly improves over other lightweight baselines (retain-only fine-tuning / distillation-only), indicating that combining retained-set learning with explicit forgetting constraints can avoid the degradation observed in forget-only training.

Forgetting behavior and privacy signal.

Beyond utility, the key goal of unlearning is to reduce the model's dependence on deleted data. Table 1 reports forgetting-related indicators, where higher predictive uncertainty on the forget set and lower membership inference signal are desirable. Retain-only fine-tuning and distillation-only typically show weaker forgetting behavior, while forget-only increases forget-set uncertainty but at the cost of utility. HERMES produces stronger forgetting signals than retain-only fine-tuning and distillation-only, while preserving utility better than forget-only. This suggests that the proposed combination—retained-set knowledge transfer plus explicit forget-set constraints—can suppress memorized behavior without substantially damaging generalization.

Efficiency compared with retraining.

The time columns in Table 1 indicate that all lightweight post-processing baselines are significantly faster than retraining, and HERMES achieves the smallest time ratio among the compared methods. This is expected because HERMES performs unlearning through a small number of additional student-training epochs rather than full retraining, providing a practical speed advantage especially when deletion requests are frequent.

Column/value deletion behavior.

For column/value deletion settings, Table 2 shows that HERMES remains stable in utility while better aligning with the intended deletion behavior compared with

simple fine-tuning. In particular, when deleted columns/values are involved, models that only fine-tune on retained data may still implicitly depend on removed attributes through representation shortcuts. By contrast, HERMES explicitly encourages invariance under counterfactual replacements of deleted attributes/values, which helps reduce such dependency while maintaining overall performance.

Ablation study.

Finally, Table 3 supports the role of each component. Removing the forgetting-related terms weakens unlearning behavior, while removing the counterfactual component degrades deletion compliance for column/value settings. Meanwhile, removing the consistency regularization can slightly affect stability and utility. These results indicate that each term in the objective contributes to the final balance between utility preservation and forgetting strength.

In summary, across the reported results in Tables 1–3, HERMES provides (i) test performance close to retraining, (ii) stronger forgetting signals than retain-only fine-tuning or distillation-only, and (iii) clear efficiency benefits over full retraining, making it a practical approach for unlearning in tabular-to-hypergraph models.

Table 1. Row unlearning results Adults.

Method	Table Column Head		
	Test Acc	MIA AUC	Time (vs Retrain)
Retrain-from-scratch	0.858 ± 0.002	0.52	1.00×
Retain-only FT	0.856 ± 0.003	0.62	0.50×
Distill-only	0.857 ± 0.002	0.56	0.58x
Forget-only	0.842 ± 0.004	0.56	0.55x
HERMES	0.857 ± 0.002	0.53	0.42x

Table 2. Row unlearning results Bank.

Method	Table Column Head		
	Test Acc	MIA AUC	Time (vs Retrain)
Retrain-from-scratch	0.908 ± 0.001	0.59	1.00×
Retain-only FT	0.907 ± 0.002	0.62	0.50×
Distill-only	0.907 ± 0.002	0.60	0.58x
Forget-only	0.907 ± 0.002	0.67	0.50x
HERMES	0.907 ± 0.001	0.61	0.23x

Table 3. Row unlearning results Covertype.

Method	Table Column Head		
	Test Acc	MIA AUC	Time (vs Retrain)
Retrain-from-scratch	0.965 ± 0.001	0.52	1.00×
Retain-only FT	0.926 ± 0.113	0.62	0.50×
Distill-only	0.922 ± 0.220	0.60	0.51x
Forget-only	0.943 ± 0.145	0.60	0.45x

Method	Table Column Head		
	Test Acc	MIA AUC	Time (vs Retrain)
HERMES	0.946 ± 0.307	0.59	0.45x

5. Conclusion

We introduced HERMES, a new unlearning paradigm for tabular-to-hypergraph models that reframes unlearning as selective knowledge transfer. By distilling on retained data with structure-aware consistency and enforcing maximum-entropy plus anti-distillation divergence on forgotten data, HERMES actively erases memorized behavior without relying on graph partitioning or second-order statistics. Counterfactual invariance further prevents the model from depending on deleted columns/values, even through indirect hypergraph propagation paths. This makes HERMES a practical and auditable post-processing approach for compliant unlearning in hypergraph-based tabular modeling.

References

[1] Chen P, Sarkar S, Lausen L, et al. Hytrel: Hypergraph-enhanced tabular data representation learning[J]. Advances in Neural Information Processing Systems, 2023, 36: 32173-32193.

[2] Arik S Ö, Pfister T. Tabnet: Attentive interpretable tabular learning[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(8): 6679-6687.

[3] Newatia A, Cooper M, Krishnan R. Unlearning Tabular Data Without a" Forget Set''[C]//NeurIPS 2024 Third Table Representation Learning Workshop.

[4] Xu C, Huang H, Ying X, et al. HGNN: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks[J]. Information Sciences, 2022, 607: 783-798.

[5] Zhu Z, Fan X, Chu X, et al. HGCN: A heterogeneous graph convolutional network-based deep learning model toward collective classification[C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 1161-1171.

[6] Bretto A. Hypergraph theory[J]. An introduction. Mathematical Engineering. Cham: Springer, 2013, 1: 209-216.

[7] Bourtoule L, Chandrasekaran V, Choquette-Choo C A, et al. Machine unlearning[C]//2021 IEEE symposium on security and privacy (SP). IEEE, 2021: 141-159.

[8] Nguyen T T, Huynh T T, Ren Z, et al. A survey of machine unlearning[J]. ACM Transactions on Intelligent Systems and Technology, 2025, 16(5): 1-46.

[9] Zhang H, Nakamura T, Isohara T, et al. A review on machine unlearning[J]. SN Computer Science, 2023, 4(4): 337.