INDUCTIVE ALIGNMENT FOR TABLE REPRESENTATION WITH FIDELITY AND CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective representation learning for tabular data is critical for downstream tasks such as information retrieval, classification, and missing value imputation. However, existing transformer-based models often fail to generalize across in-domain tables, either preserving schema–value semantics at the cost of robustness or enforcing stability while losing fidelity. We propose NAVI—Entropy-aware Alignment via Header–Value Induction—a framework that unifies both desiderata. NAVI introduces header–value segments as the atomic unit of table representation, serialized in an order-independent manner and anchored by global header embeddings. Structure-aware masked segment modeling enforces schema–value dependencies via balanced masking over headers, values, and tokens, while entropy-driven segment alignment aligns low-entropy (domain-coherent) columns with global headers and high-entropy (entity-discriminative) columns with row-specific values. This joint design yields representations that are both consistent and semantically faithful. Extensive experiments on large-scale benchmarks show that NAVI consistently outperforms baselines in generative and discriminative tasks while mitigating schema-level inconsistencies. The source code of NAVI is available at: `https://anonymous.4open.science/r/navi`.

1 INTRODUCTION

Motivation. Tabular data is pervasive across domains such as e-commerce, healthcare, and finance (Zavitsanos et al., 2024; Batko & Ślęzak, 2022). In contrast to unstructured text, whose semantics naturally emerge from an inherent sequence of tokens, tables are defined by an inherent structure of rows, columns, and headers. This structure not only constrains how information is organized but also facilitates the sharing of domain knowledge across tables within the same application context. Consequently, the domain-aware semantics of tabular data fundamentally arise from structurally aligned header–value relationships, which makes table representation learning distinct from that of unstructured text. To this end, models must capture domain-specific structural semantics in tables while preserving value *distinctiveness* and ensuring schema *robustness* within the same domain.

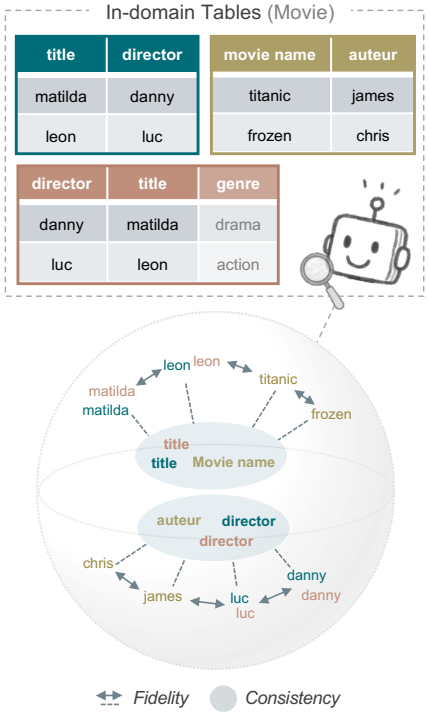

Figure 1: Table embedding desiderata.

As illustrated in Fig. 1, we formalize these requirements into two desiderata, **fidelity** (i.e., distinctiveness) and **consistency** (i.e., robustness), for effective in-domain table representation learning. Fidelity requires that representations remain faithful to table semantics by preserving both structural and domain-specific information. Specifically, (1) *structural fidelity* ensures that cells reflect their functional roles within rows, columns, and headers; and (2) *domain fidelity* ensures that cell

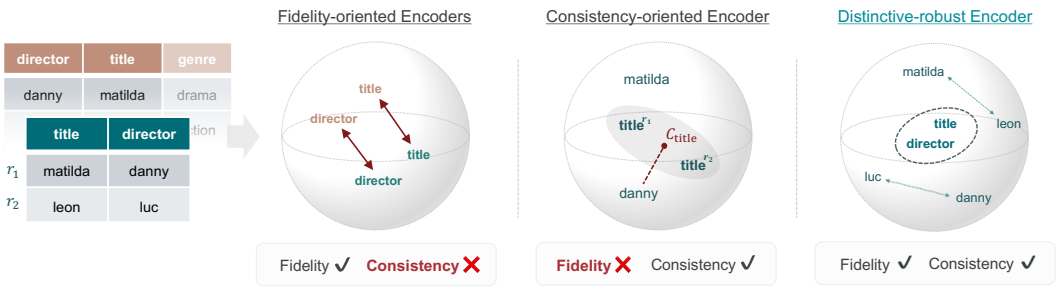

Figure 2: Existing approaches exhibit a trade-off between fidelity and consistency. They either achieve structural fidelity but sacrifice consistency or achieve domain consistency but weaken fidelity. Our framework unifies both desiderata in a balanced manner.

values remain distinguishable within tables. Consistency requires that representations remain stable despite structural or domain-level variations. Specifically, (3) *structural consistency* maintains invariance to reordering or perturbations of rows and columns; and (4) *domain consistency* maintains coherence of equivalent headers within tables.

Existing Works. Most previous efforts represent tabular data using Transformer (Fang et al., 2024; Badaro et al., 2023), which have demonstrated remarkable capability on unstructured data (Vaswani et al., 2017; Devlin et al., 2019). These approaches serialize tabular data into token sequences and inject table-specific inductive biases in the self-attention mechanism to preserve structural properties. However, as in the typical trade-off between distinctiveness and robustness in representation learning, these methods face clear limitations in simultaneously achieving both fidelity and consistency for in-domain tables, as illustrated in Fig. 2.

On the one hand, fidelity-oriented methods (Herzig et al., 2020; Yin et al., 2020; Iida et al., 2021; Deng et al., 2022; Wang et al., 2021) explicitly model rows, columns, or tree structures to capture fine-grained schema information of a table. By contextualizing tokens according to their functional roles in a table with vertical or horizontal attention, these methods achieve strong *fidelity*. However, they compromise the consistency of table representations; vulnerability to schema variations undermines *structural consistency*, while table-specific designs hinder generalization across in-domain tables, weakening *domain consistency*.

On the other hand, consistency-oriented methods (Jung & Yoon, 2025) emphasize schema-level stability. By leveraging universal embeddings and regularization, they preserve stable domain knowledge shared across in-domain tables (e.g., common headers), thereby achieving *domain consistency* against schema variants. Yet this stability comes at the expense of fidelity; overly smoothed header–value semantics undermine *structural fidelity*, while decoupled learning of common header semantics weakens *domain fidelity*.

These limitations of existing methods fundamentally arise from the token-level contextualization of tabular data, similar to unstructured text, where table-specific adaptations function only as a limited workaround. Merely token-level encoding fails to accurately learn header-value relationships, undermining fidelity, and to be fully aware of schema or lexical variation to preserve consistency.

Main idea and Contributions. To bridge this gap, we introduce the concept of a header-value *segment*, a minimal yet semantically meaningful unit of a table that integrates structural roles with domain semantics. By treating the segment as the fundamental building block of representation learning, models can encode the essence of in-domain tables into a unified embedding that simultaneously balances fidelity and consistency. Grounded in this concept, we propose a novel tabular embedding framework **NAVI**; ENtropy-aware Alignment with Header-Value Induction. NAVI aims to capture the structural properties of tables through *schema-aware segment induction and modeling*. It also employs *entropy-driven alignment of segments* to selectively incorporate domain knowledge shared among in-domain tables.

In summary, we make the following contributions: (1) We identify the two key desiderata, fidelity and consistency, as a principled foundation for effective in-domain table representation learning. (2) We introduce the notion of a header-value segment as the fundamental building block of tables, and

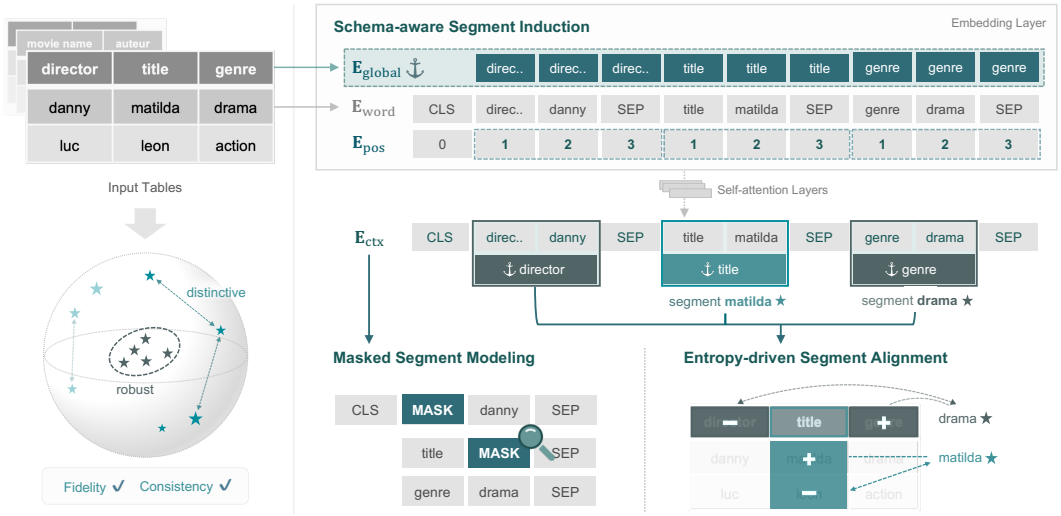

Figure 3: Overall procedure of NAVI. We optimize schema-induced representations for tokens and segments with masked modeling and entropy-driven alignment, preserving intra-table fidelity and inter-table consistency for in-domain tables.

propose NAVI, a novel segment-centric embedding framework, with a theoretical analysis for the two desiderata. (3) We conduct extensive experiments on real-world in-domain tables, showing that NAVI outperforms existing baselines in both discriminative and generative downstream tasks. In addition, qualitative analyses further demonstrate the effectiveness of NAVI.

2 METHODOLOGY

We present a three-stage framework for segment-grounded representation learning from tabular data. Our methodology consists of: (1) Schema-aware Segment Induction, which defines the header–value segment as a structural unit and incorporates context-free header embeddings to ensure structural consistency; (2) Masked Segment Modeling, which extends the masked language modeling (MLM) objective with balanced masking of headers and values to enforce fine-grained schema–value dependencies, thereby achieving structural fidelity; and (3) Entropy-driven Segment Alignment, which leverages column entropy to distinguish domain-defining from entity-defining attributes, applying cross-column and cross-row alignment to ensure domain consistency and domain fidelity.

2.1 SCHEMA-AWARE SEGMENT INDUCTION

Header-Value Segment. Unlike natural language, tables are inherently organized into rows, where each row represents a distinct entity. To preserve this entity-level semantics, table representation should be managed at the row level, avoiding unnecessary entanglement across rows. Within each row, however, a sequential input format is needed to explicitly capture its features without introducing spurious dependencies on schema order. This calls for a more explicit structural unit—one that grounds schema semantics without imposing order bias. We introduce the *header-value segment*, where each segment is defined as a header–value pair (i.e., `header : value[SEP]`). A row is then serialized into a set of segments, prefixed by a special token. For example:

director	title	genre	...
danny	matilda	drama	...
...	...	...	...

→ segment

`[CLS] director : danny [SEP]`
`title : matilda [SEP] ... [SEP]`

Tables are serialized row-wise, treating each row as an independent sequence of segments. This guarantees row permutation invariance, i.e., the representation of the table remains unchanged regardless of row order. Within each row, we apply segment-wise positional encoding. Specifically, for segment k of length m_k:

$$E_{\text{pos}}(x_j^{(k)}) = P_j, \quad j = 0, \ldots, m_k - 1,$$

where P_j is a sinusoidal positional embedding. This local reinitialization ensures that column order does not bias the encoding, achieving column permutation invariance. Together, these invariances constitute a form of *structural consistency*, a key desideratum for tabular representation learning.

Global Header Representation. To anchor header semantics consistently across contexts (e.g., rows, tables), we introduce a lightweight encoder dedicated for encoding header strings. Given a header tokenized as $h = [t_1, \ldots, t_n]$, the encoder produces self-attended embeddings $\{e_{t_1}, \ldots, e_{t_n}\}$, $e_{t_k} \in \mathbb{R}^d$. A single universal embedding for header h, $E_{\text{global}}(h) \in \mathbb{R}^d$ is then obtained by pooling, independent of any specific table context.

Unlike prior approaches that construct column embeddings by coupling headers with local values (Yin et al., 2020; Iida et al., 2021), our header representations remain context-free. This provides a consistent semantic anchor across diverse tables and serves as a supportive bias for domain consistency. It is further complemented by stronger distribution-level regularization through entropy-driven segment alignment.

Header-conditioned Token Representation. To provide a supportive bias toward domain-level consistency, we condition token embeddings on their corresponding global header representations. Specifically, $E_{\text{global}}(h)$ is added as a bias to each token $x_j^{(k)}$ within its segment:

$$z_j^{(k)} = E_{\text{word}}(x_j^{(k)}) + E_{\text{pos}}(x_j^{(k)}) + E_{\text{global}}(h_k).$$

These conditioned embeddings are contextualized by a transformer encoder, yielding token representations $\mathbf{e}_t \in \mathbb{R}^d$. This mechanism enforces a stable schema bias, promoting *structural fidelity* by maintaining schema-value dependencies under noisy or mutated tables.

Header-conditioned Segment Representation. At the row level, we construct segment embeddings that integrate both schema anchors and contextualized representations. For header h in row r, we first obtain contextualized token embeddings from the transformer encoder output. The contextualized header and value embeddings, respectively denoted as $H_{\text{ctx}}(r, h)$ and $V_{\text{ctx}}(r, h)$, are extracted by pooling the contextualized token embeddings at the positions of h and its corresponding value.

Finally, the segment embedding concatenates the global header with row-aware components:

$$E_{\text{seg}}(r, h) = g\big(E_{\text{global}}(h) \parallel H_{\text{ctx}}(r, h) \parallel V_{\text{ctx}}(r, h)\big),$$

where $\parallel$ denotes concatenation, and $g(\cdot)$ is a projection network. Such integrated embeddings capture both schema–value dependencies and global semantics, thereby providing a relationally expressive foundation that promotes *domain fidelity* across heterogeneous tables.

2.2 Masked Segment Modeling

Structure-based Masking. Standard masked language modeling (MLM) has proven effective for natural language (Devlin et al., 2019), but its direct application to tables is suboptimal. Headers and values are semantically different, and their dependencies are crucial for relational reasoning. By treating all tokens uniformly, Vanilla MLM risks overlooking the structural distinction between schema and content, undermining its ability to maintain *structural fidelity*. To address this, we introduce a structure-aware masked segment modeling (MSM) that explicitly models schema–value dependencies by partitioning each row of segments into three masking regimes:

- **Header-masked segments:** header tokens in selected segments are masked, forming the set $\mathcal{M}_h$. The model must recover header names from associated values.
- **Value-masked segments:** value tokens in selected segments are masked, forming the set $\mathcal{M}_v$. The model must infer values from headers and row context.
- **Vanilla MLM:** a random subset of remaining tokens is masked, forming $\mathcal{M}_r$. This acts as a regularization term that prevents overfitting to header–value co-occurrence patterns.

Objective. For masked tokens $t \in \mathcal{M}$ with token embeddings $\mathbf{e}_t$, the MSM loss is:

$$\mathcal{L}_{\text{msm}} = -\frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \log \frac{\exp(W\mathbf{e}_t + b)}{\sum_{v \in \mathcal{V}} \exp(W\mathbf{e}_v + b)}, \quad \mathcal{M} = \mathcal{M}_h \cup \mathcal{M}_v \cup \mathcal{M}_r,$$

where $\mathcal{V}$ is the vocabulary, and (W, b) are classifier parameters. This MSM objective with structured masking compels the encoder to learn functional roles of tokens and schema-value dependencies, thereby realizing the desideratum of *structural fidelity*.

2.3 Entropy-driven Segment Alignment

Entropy-based Column Categorization. While the preceding methods ensure structural consistency and structural fidelity, they do not by themselves guarantee domain consistency or domain fidelity. To achieve these desiderata, we require an additional mechanism that explicitly aligns representations. Contrastive learning (Oord et al., 2018; Chen et al., 2020; Lee et al., 2022) has been widely used to arrange embeddings according to a target semantic objective, but the straightforward adoption—applying contrastive loss directly at the row (i.e., instance, entity) level—fails to distinguish between schema-level semantics and instance-specific attributes. This results in entangled representations that blur domain boundaries or collapse row-level distinctions.

To overcome this limitation, we propose an entropy-based column categorization. Instead of aligning rows indiscriminately, we categorize columns by the entropy of their empirical value distributions and use this categorization as the foundation for aligning segments and headers-values:

- **Domain-coherent columns $\mathcal{H}_{\text{dom}}$:** low-entropy columns (e.g., below the first quartile, Q1) with stable domain-level concepts (e.g., genre and director in movie tables). Aligning their segments and headers enforces consistent semantics across tables, promoting *domain consistency*.
- **Entity-discriminative columns $\mathcal{H}_{\text{ent}}$:** high-entropy columns (e.g., above the third quartile, Q3) with instance-specific attributes (e.g., title and url in movie tables). Aligning their segments and values enhances row separability within a domain, enhancing *domain fidelity*.

Objective. Given a query q, a positive sample x^+, a set of negative samples $\mathcal{X}^-$, and a temperature τ, the InfoNCE objective (Oord et al., 2018) is set as:

$$\mathcal{L}_{\text{InfoNCE}}(q, x^+, \mathcal{X}^-, \tau) = -\log \frac{\exp(q \cdot x^+/\tau)}{\exp(q \cdot x^+/\tau) + \sum_{x^- \in \mathcal{X}^-} \exp(q \cdot x^-/\tau)}.$$

For headers in domain-coherent columns $h_{\text{dom}} \in \mathcal{H}_{\text{dom}}$, cross-header alignment matches segments with their global header embeddings. This ensures that headers representing similar domain concepts are consistently aligned across rows and tables. We optimize the domain-coherent loss:

$$\mathcal{L}^t_{\text{dom}} = \mathbb{E}_{r\sim\mathcal{R},\ h\sim\mathcal{H}_{\text{dom}}}\big[\mathcal{L}_{\text{InfoNCE}}(q_{\text{dom}}(r,h), x^+_{\text{dom}}(h), \mathcal{X}^-_{\text{dom}}(h), \tau_{\text{dom}})\big], \text{where}$$
$$q_{\text{dom}}(r,h) = E_{\text{seg}}(r,h),\ x^+_{\text{dom}}(h) = E_{\text{global}}(h),\ \mathcal{X}^-_{\text{dom}}(h') = \{\, E_{\text{global}}(h') \mid h' \in \mathcal{H}_{\text{dom}},\ h' \neq h \,\}.$$

For headers in entity-discriminative columns $h_{\text{ent}} \in \mathcal{H}_{\text{ent}}$, cross-row alignment matches segments with row-aware values. This encourages row-level separability by ensuring distinct rows in the same table remain distinguishable. We optimize the entity-discriminative loss:

$$\mathcal{L}^t_{\text{ent}} = \mathbb{E}_{r\sim\mathcal{R},\ h\sim\mathcal{H}_{\text{ent}}}\big[\mathcal{L}_{\text{InfoNCE}}(q_{\text{ent}}(r,h), x^+_{\text{ent}}(h), \mathcal{X}^-_{\text{ent}}(h), \tau_{\text{ent}})\big], \text{where}$$
$$q_{\text{ent}}(r,h) = E_{\text{seg}}(r,h),\ x^+_{\text{ent}}(r,h) = V_{\text{ctx}}(r,h),\ \mathcal{X}^-_{\text{ent}}(r,h) = \{\, V_{\text{ctx}}(r',h) \mid r' \in \mathcal{R},\ r' \neq r \,\}.$$

Finally, given a batch $\mathcal{B}$ in input tables and a balancing parameter λ_{align} for generative and discriminative supervision, the overall training objective is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{msm}} + \lambda_{\text{align}} \cdot \mathcal{L}_{\text{align}}, \text{ where } \mathcal{L}_{\text{align}} = 1/|\mathcal{B}| \cdot \sum_{t\in\mathcal{B}} (\mathcal{L}^t_{\text{dom}} + \mathcal{L}^t_{\text{ent}}).$$

Appendix A discusses the theoretical analysis of the schema induction and contrastive alignment.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets. We evaluate on four datasets from two domains, Movie and Product. For pretraining, we use subsets of WDC WebTables (Peeters et al., 2024), selecting the 100 largest tables per domain—WDC Movie (480,817 rows) and WDC Product (3,930,877 rows)—and subsample 480,817 rows from each for balance. To ensure compatibility with BERT-style models, all tables are processed through a standardized pipeline (see Appendix C.2). For downstream evaluation, we construct held-out subsets of 45,000 rows per domain (≈10% of pretraining). We uniformly subsample 1,000 rows per each evaluation run for consistency and efficiency.

Baseline Methods. We evaluate our approach against representative table embedding models spanning major paradigms. BERT serves as the generic transformer backbone underlying most language model–based table encoders; its performance highlights the limitations of applying vanilla language models to tabular data. TAPAS, the most widely adopted table encoder, exemplifies fidelity-oriented approaches, while HAETAE represents a consistency-oriented encoder. This selection enables a systematic comparison of their strengths and limitations with respect to fidelity and consistency.

Implementaion. We configure NAVI to balance domain and structural objectives. For domain objectives, we set contrastive temperature τ to 0.07 for entity-discriminative columns and 0.13 for domain-coherent columns, and vary the alignment weight λ_{align} across tasks. For structural objectives, we adjust the header–value–baseline (H:V:B) masking ratio. We use $\lambda_{\text{align}} = 0.5$ and H:V:B = 4:4:2 as the default. Further details and sensitivity analysis appear in Appendices C and D. All models are trained on the same datasets for 2 epochs with a batch size of 32, AdamW (Loshchilov & Hutter) with a learning rate of 3×10^{-5}, and a weight decay of 0.01.

3.2 FIDELITY ANALYSIS

We evaluate *fidelity*, the faithfulness of representations to table semantics. Fidelity spans two dimensions: *domain fidelity*, which preserves entity-level discriminability, and *structural fidelity*, which models schema–value dependencies within rows. We probe domain fidelity through **discriminative tasks** (Row Classification and Row Clustering) and structural fidelity through **generative tasks** (Value Imputation and Header Prediction).

Table 1: Performance on discriminative tasks. The table shows results from [CLS] token embeddings. Macro-F1 scores for classification (using XGBoost and Logistic Regression) and Silhouette and B³-F1 scores for clustering (using KMeans and Agglomerative).

	Product				**Movie**			
	R-Cls (F1)		**R-Clt** (Sil. / B³-F1)		**R-Cls** (F1)		**R-Clt** (Sil. / B³-F1)	
Model	**XGB**	**LR**	**KMeans**	**Agglo.**	**XGB**	**LR**	**KMeans**	**Agglo.**
BERT	0.280	0.360	0.053 / 0.210	0.060 / 0.222	0.251	0.297	0.101 / 0.214	0.132 / 0.215
TAPAS	0.239	0.356	**0.085** / 0.234	**0.084** / 0.234	**0.289**	**0.335**	0.137 / 0.194	0.141 / 0.200
HAETAE	0.250	0.343	0.055 / 0.202	0.061 / 0.225	0.256	0.295	0.115 / 0.209	0.132 / 0.210
NAVI	**0.355**	**0.417**	0.062 / **0.248**	0.069 / **0.273**	0.275	0.313	**0.225 / 0.236**	**0.300 / 0.233**

Discriminative Tasks. To assess *domain fidelity*—whether embeddings preserve entity-level separability—we evaluate *Row Classification* and *Row Clustering*. Classification uses 20 balanced classes per domain (top product categories, top movie genres; ∼1,000 samples), with Macro-F1 from Logistic Regression and XGBoost. Clustering probes the same label space via KMeans and Agglomerative, scored by Silhouette and B³-F1. Results are averaged over 8 subsampled runs. As shown in Table 1, BERT achieves only modest F1 from shallow cues, while HAETAE underperforms as rigid anchoring limits row discriminability. TAPAS improves fidelity but remains schema-sensitive. By contrast, NAVI consistently leads across classifiers and clustering, yielding compact, coherent manifolds. These results show that entropy-driven alignment mitigates row collapse and enhances entity-level fidelity under schema diversity.

Generative Tasks. We examine *structural fidelity*, i.e., whether embeddings capture schema–value dependencies, through two tasks: *Header Prediction* and *Value Imputation*, respectively recovering masked headers and values from contextualized row tokens. We compare NAVI against BERT and HAETAE, which are naturally suited for generative tasks, but exclude TAPAS as its QA-oriented pretraining objective makes it infeasible for this setting. As shown in Table 2, NAVI achieves near-perfect header prediction, validating its global header encoder as a stable semantic anchor, and also outperforms in value imputation, where header-conditioned representations and structure-aware masking reinforce schema–value dependencies.

Table 2: Generative tasks.

Model	Product	Movie
Header		
BERT	0.8788	0.8758
HAETAE	0.8496	0.8439
NAVI	**0.9958**	**0.9985**
Value		
BERT	0.7230	0.6235
HAETAE	0.7298	0.6225
NAVI	**0.7406**	**0.6414**

3.3 Consistency Analysis

We next evaluate *consistency*, the stability of representations under schema diversity. Consistency has two dimensions: *domain consistency* and *structural consistency*, which together denote invariance to lexical and structural diversity. Domain consistency is measured by clustering semantically equivalent headers (e.g., `director` vs. `auteur`) using agglomerative clustering, with quality assessed by B³-F1 and NMI. Structural consistency is measured by permuting rows and computing the permutation sensitivity index (PSI $= \mathbb{E}_k[1 - \cos(z, \tilde{z}^{(k)})]$), where z is the original row embedding and $\tilde{z}^{(k)}$ its k-th permutation, using both CLS and mean pooling. Consistent models should form compact header clusters and yield low PSI.

Table 3: Domain consistency of header clustering (H-Clt) is evaluated by B³-F1 and NMI (higher is better), and structural consistency of row permutation is evaluated with PSI (lower is better).

	Product		Movie	
Model	**H-Clt** (B³-F1 / NMI)	**PSI** (cls / mean)	**H-Clt** (B³-F1 / NMI)	**PSI** (cls / mean)
BERT	0.6951 / 0.8642	6.60 e-2 / 6.09 e-3	0.6727 / 0.8717	6.21 e-2 / 5.49 e-3
TAPAS	0.7335 / 0.8814	1.29 e-2 / 5.92 e-3	0.6617 / 0.8641	9.75 e-3 / 5.05 e-3
HAETAE	0.7552 / 0.8855	6.81 e-2 / 5.73 e-3	0.7056 / 0.8806	6.34 e-2 / 4.91 e-3
NAVI	**0.7948 / 0.8978**	**8.60 e-8 / 7.62 e-9**	**0.7969 / 0.9071**	**2.82 e-7 / 5.96 e-10**

Lexical Diversity. On header clustering, NAVI yields the most coherent groups. HAETAE is competitive but still weaker than NAVI's alignment. Figure 4 illustrates this for the `actor` set: under NAVI (top), lexical variants converge into one cluster, while under BERT (bottom) they remain split. This contrast shows BERT encodes surface forms, whereas NAVI collapses aliases into canonical representations. Thus, the induction with contrastive alignment enforces domain consistency.

Structural Diversity. As shown in Table 3, NAVI achieves near-zero PSI across both domains, far outperforming BERT, TAPAS, and HAETAE. This indicates strong invariance to column reordering, unlike baselines that drift. HAETAE reduces PSI relative to BERT and TAPAS (e.g., 5.73×10^{-3} on Product, 4.91×10^{-3} on Movie), yet NAVI improves by orders of magnitude, confirming that its schema induction stabilizes embeddings more effectively under structural perturbations.

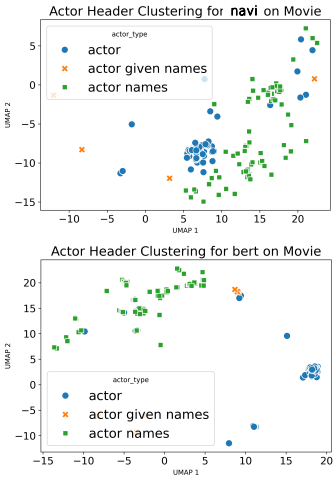

Figure 4: Header embeddings.

3.4 Ablation Study

To disentangle the contribution of each component in NAVI, we organize our analysis along the four desiderata of our evaluation framework. On the fidelity side, we assess *domain fidelity* with

Row Classification (R-Cls) and *structural fidelity* with Value Imputation (Val). On the consistency side, we measure *domain consistency* with Header Clustering (H-Clt) and *structural consistency* with the Permutation Sensitivity Index (PSI). This one-to-one mapping provides a clear lens into how Schema-aware Segment Induction (SSI), Structure-aware MSM (SMSM), and Entropy-driven Segment Alignment (ESA) each contribute to representations that are both faithful and consistent.

Table 4: Classification (Logistic Regression - F1), Accuracy for Value Imputation, Header Clustering (Agglomerative - B3-F1), Permutation Sensitivity Index (computed from mean pooled row embeddings) across Product and Movie domains.

	Product				Movie			
Variant	**R-Cls**	**Val**	**H-Clt**	**PSI**	**R-Cls**	**Val**	**H-Clt**	**PSI**
NAVI	**0.4166**	**0.7365**	0.7948	**7.6e-9**	**0.3125**	**0.6414**	**0.7969**	**6.0e-10**
w/o SSI	0.3532	0.2462	0.1297	1.8e-8	0.2710	0.2261	0.1456	1.8e-8
w/o SMSM	0.2671	0.6926	**0.8030**	1.0e-8	0.2667	0.5771	0.7666	1.1e-8
w/o ESA	0.3805	0.7354	0.7811	1.1e-8	0.3039	0.6062	0.7915	2.2e-8

Results in Table 4 clarify how each module of NAVI sustains our four desiderata. Removing SSI yields the most severe degradation, collapsing Value Imputation and Header Clustering, which confirms schema anchoring as the linchpin of both fidelity and consistency. Excluding SMSM sharply reduces Row Classification and weakens Value Imputation, showing that balanced masking is essential for schema–value fidelity; its slight gain in Header Clustering further reveals a trade-off between fine-grained dependencies and global alignment. Dropping ESA leads to moderate losses in classification and imputation and noticeably higher PSI, highlighting its role in safeguarding entity-level discriminability and robustness to permutation. Taken together, these results show that SSI, SMSM, and ESA are not interchangeable but complementary: only their integration produces embeddings that are simultaneously faithful, consistent, and robust across domains.

3.5 Qualitative Analysis

Core–Periphery Structure in Segment Embeddings. The scatter plot in Figure 5 provides qualitative evidence that our segment-centric framework captures the geometry of in-domain table representations. We plot embeddings from five movie tables using t-SNE, with convex hulls marking table boundaries. A clear core–periphery structure emerges: low-entropy segments (blue), corresponding to domain-coherent attributes (e.g., genre), cluster in the central region shared across tables. This overlap is a manifestation of domain consistency—the model collapses schema semantics into a global domain center. In contrast, high-entropy segments (red), representing entity- or table-specific attributes (e.g., title), disperse outward with little overlap, reflecting domain fidelity by preserving distinctiveness. The distribution plot, based on radial distances from the low-entropy centroid in the original embedding space, provides complementary evidence for this geometry. Low-entropy segments form a sharp density peak near the centroid, indicating alignment across tables. High-entropy segments spread over broader radii, confirming they remain distinctive rather than collapsing. Together, these results illustrate that entropy-driven alignment achieves the desired balance: a shared domain semantic core with a flexible entity-specific periphery. This evidence complements quantitative gains and offers an intuitive geometric account of how segment embeddings realize both fidelity and consistency.

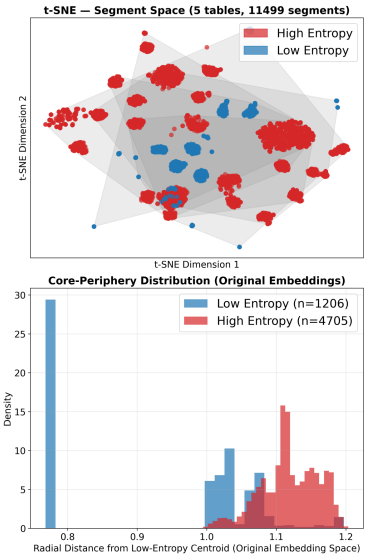

Figure 5: visualization of segment embedding space and the radial distance based distributions.

4 CONCLUSION

In this paper, we revisit table representation through the two principled desiderata, fidelity and consistency, and exploit the header–value segment as the atomic unit to balance them. NAVI implements this idea with (i) Schema-aware Segment Induction (SSI) that builds segment embeddings anchored by a global, context-free header encoder, (ii) Masked Segment Modeling (MSM) that enforces fine-grained schema–value dependencies, and (iii) Entropy-driven Segment Alignment (ESA) that aligns domain-coherent columns while preserving separation for entity-discriminative ones. Empirical studies demonstrated that NAVI achieves higher performances on both header prediction and value imputation, in addition to consistent gains on classification and clustering tasks. Qualitatively, the resulting embedding space exhibits a core–periphery geometry (i.e., a shared semantic core for stable headers and a flexible periphery for instance-specific attributes) in accordance with our learning objectives. Ablation studies also confirm that the efficacy of the three main components: SSI as the building blocks for fidelity and consistency, SMSM for schema–value coupling, and ESA for permutation stability and row discriminability. Together, these results position NAVI as a segment-centric, alignment-guided alternative to existing token-oriented encoders, narrowing the gap between symbolic tabular data and contextualized representations. We believe this work opens up practical opportunities and future work for applications with LLM-table interactions, such as question answering and retrieval-augmented generation on in-domain tables.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 2023.

Kornelia Batko and Andrzej Ślęzak. The use of big data analytics in healthcare. *Journal of big Data*, 9(1):3, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes (eds.), *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models on tabular data: Prediction, generation, and understanding–a survey. *arXiv preprint arXiv:2402.17944*, 2024.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *NeurIPS 2022 First Table Representation Workshop*.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. Tabbie: Pretrained representations of tabular data. *arXiv preprint arXiv:2105.02584*, 2021.

Woojun Jung and Susik Yoon. HAETAE: In-domain table pretraining with header anchoring. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3065–3069, 2025.

Janghyeon Lee, Jongsuk Kim, Hyounguk Shon, Bumsoo Kim, Seung Hwan Kim, Honglak Lee, and Junmo Kim. Uniclip: Unified framework for contrastive language-image pre-training. *Advances in Neural Information Processing Systems*, 35:1008–1019, 2022.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. Table-GPT: Table fine-tuned gpt for diverse table tasks. 2024.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. Tapex: Table pre-training via learning a neural sql executor. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Markus Mueller, Kathrin Gruber, and Dennis Fok. Continuous diffusion for mixed-type tabular data. *arXiv preprint arXiv:2312.10431*, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ralph Peeters, Alexander Brinkmann, and Christian Bizer. The web data commons schema.org table corpora. In *Companion Proceedings of the ACM Web Conference 2024*, 2024.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International conference on machine learning*, pp. 5628–5637. PMLR, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pp. 9929–9939. PMLR, 2020.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1780–1790, 2021.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Elias Zavitsanos, Dimitris Mavroeidis, Eirini Spyropoulou, Manos Fergadiotis, and Georgios Paliouras. Entrant: A large financial dataset for table understanding. *Scientific Data*, 11(1): 876, 2024.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. Tapera: enhancing faithfulness and interpretability in long-form table qa by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12824–12840, 2024.

A THEORETICAL ANALYSIS

A.1 SCHEMA INDUCTION: A MECHANISTIC ANALYSIS OF STRUCTURAL PROPERTIES

We analyze our **Schema-aware Segment Induction**, a theoretically grounded mechanism that introduces two inductive biases essential for table representation learning: (1) *Header–Value Coupling*, which enforces schema–value dependencies and preserves token roles, thereby realizing **Structural Fidelity**; and (2) *Segment-Order Equivariance*, which treats rows as sets of header–value segments and removes spurious order dependence, thereby realizing **Structural Consistency**.

STRUCTURAL FIDELITY VIA SCHEMA-CONSISTENT ATTENTION ROUTING

Setup. For segment k with header $h^{(k)}$, let the token representation be

$$z_p = z_{\text{base}} + E_{\text{global}}(h^{(k)}),$$

where z_{base} contains word and positional embeddings and $E_{\text{global}}(h^{(k)})$ is the universal header embedding. Queries and keys are linear maps $Q_p = W_Q z_p$, $K_q = W_K z_q$.

Analysis. Let the token representation for p in segment k be

$$z_p = b_p + E, \qquad b_p := z_{\text{base},p}, \quad E := E_{\text{global}}(h^{(k)}).$$

With $Q_p = W_Q z_p$, $K_q = W_K z_q$ and $M := W_Q^\top W_K$, the attention logit expands to

$$\begin{aligned} \ell_{pq} = Q_p^\top K_q &= (b_p + E)^\top M (b_q + E) \\ &= b_p^\top M b_q + b_p^\top M E + E^\top M b_q + E^\top M E. \end{aligned} \tag{1}$$

The quadratic term $E^\top M E$ is independent of (p, q) and thus acts as a *shared, header-dependent bias* within the entire segment k. The two cross-terms vary with p, q, but under LayerNorm (Ba et al., 2016) ($\mathbb{E}[b_p] = \mathbb{E}[b_q] = 0$) their expectations vanish. Hence the expected logit decomposes as

$$\mathbb{E}_{p,q}\, \ell_{pq} = \mathbb{E}_{p,q}[b_p^\top M b_q] + E^\top M E,$$

where the second term is the segment-wide bias.

Gradient w.r.t. E, Differentiating equation 1 gives

$$\nabla_E \ell_{pq} = M b_q + M^\top b_p + (M + M^\top) E.$$

Averaging over all (p, q) within the segment yields

$$\mathbb{E}_{p,q}\, \nabla_E \ell_{pq} = (M + M^\top) E,$$

since $\mathbb{E}[b_p] = \mathbb{E}[b_q] = 0$. Thus, the expected update direction is the same for all tokens in the segment, depending only on E and the projection matrices.

Since the MSM loss is token-level cross-entropy and analytically unwieldy, we study a surrogate quadratic objective $\mathcal{J}(E)$ that isolates the effect of header embeddings on attention logits.

$$\mathcal{J}(E) := \sum_{p,q} \ell_{pq} = \sum_{p,q} b_p^\top M b_q + 2E^\top M \sum_q b_q + |S_k|^2 E^\top \mathrm{Sym}(M) E,$$

where $|S_k|$ is the number of tokens in the segment. The stationary point satisfies

$$\nabla_E \mathcal{J}(E) = 2M \sum_q b_q + 2|S_k|^2 \mathrm{Sym}(M) E = 0,$$

so that

$$E^\star = -\big(|S_k|^2\, \mathrm{Sym}(M)\big)^{-1} M \sum_q b_q.$$

With LayerNorm, $\sum_q b_q \approx 0$, making the optimizer align with the quadratic term $E^\top \mathrm{Sym}(M) E$.

Conclusion. Adding $E_{\text{global}}(h^{(k)})$ to all tokens yields a shared quadratic bias $E^\top \mathrm{Sym}(M) E$ independent of values, and a uniform update direction $(M + M^\top) E$. Together, these reinforce schema–value coupling consistently across tokens in a segment, ensuring **Structural Fidelity**.

STRUCTURAL CONSISTENCY VIA EQUIVARIANCE

Setup. Each row is serialized as a set of header–value segments $\{s(r, h_k)\}$, with segment-wise positional encodings but no global positions. Thus the encoder $g(\cdot)$ processes each segment independently, without reference to their global order.

Analysis. Since each segment is processed locally, the encoder g is *permutation-equivariant*:

$$g(\pi \cdot \{s(r, h_k)\}) = \pi \cdot g(\{s(r, h_k)\}).$$

For any permutation π, the encoder output followed by a permutation-invariant readout ρ, specifically mean pooling over segment embeddings, satisfies

$$f_{\mathrm{mean-pool}}(r) = \rho\left(\sum_k \phi(s(r, h_k))\right),$$

which matches the functional form of Deep Sets (Zaheer et al., 2017), with $\phi = g$ and ρ the pooling. By the universal approximation theorem for Deep Sets, f_{mean} can approximate any continuous permutation-invariant function over sets of segments.

Let z_{cls} be the row token. One self–attention update is

$$z'_{\mathrm{cls}} = \sum_q \alpha_{\mathrm{cls}\to q}\, V_q, \qquad \alpha_{\mathrm{cls}\to q} = \frac{\exp(\ell_{\mathrm{cls},q}/\tau)}{\sum_{q'} \exp(\ell_{\mathrm{cls},q'}/\tau)}, \qquad \ell_{\mathrm{cls},q} = (W_Q z_{\mathrm{cls}})^\top (W_K z_q). \quad (2)$$

For any permutation π of segments in the row, the value/key sequences are merely reindexed:

$$\{(z_q, V_q)\}_q \mapsto \{(z_{\pi(q)}, V_{\pi(q)})\}_q \quad \Rightarrow \quad \{\ell_{\mathrm{cls},q}\}_q \mapsto \{\ell_{\mathrm{cls},\pi(q)}\}_q \quad \Rightarrow \quad \{\alpha_{\mathrm{cls}\to q}\}_q \mapsto \{\alpha_{\mathrm{cls}\to\pi(q)}\}_q.$$

Plugging the reindexed weights/values into equation 2 gives

$$z'_{\mathrm{cls}}(\pi\cdot\{s(r, h_k)\}) = \sum_q \alpha_{\mathrm{cls}\to\pi(q)}\, V_{\pi(q)} = \sum_q \alpha_{\mathrm{cls}\to q}\, V_q = z'_{\mathrm{cls}}(\{s(r, h_k)\}),$$

so the CLS update is permutation *invariant* when the operation is a pure reindexing (no extra biases, identical residual paths, exact arithmetic).

Relaxation to ε–stability. In practice, residual connections, layernorm/biases and finite precision introduce small deviations. We measure these by the permutation sensitivity index (PSI):

$$\mathrm{PSI} \;=\; \mathbb{E}_\pi\big[\,1 - \cos\big(f(r), f_\pi(r)\big)\big]\,,$$

with $f(r)$ the row embedding (CLS or mean-pooled) and $f_\pi(r)$ after permuting segments by π. We say the encoder is *ε–permutation-stable* if $\mathrm{PSI} \le \varepsilon$.

Conclusion. Mean pooling yields $f(r) = \rho(\sum_k \phi(s(r, h_k)))$, targeting invariance. For CLS, the derivation above shows invariance in the ideal limit and *strong approximate* invariance in practice, with ε empirically small (Table 3). Hence both readouts achieve **Structural Consistency**.

A.2 CONTRASTIVE ALIGNMENT: GEOMETRIC FOUNDATIONS OF DOMAIN PROPERTIES

We analyze Entropy-driven Segment Alignment, an InfoNCE-based objective that provably induces a *core–periphery geometry* in the segment embedding space. Building on the alignment–uniformity framework of Wang & Isola (2020), we show that optimizing $\mathcal{L}_{\text{align}}$ contracts low-entropy (domain-coherent) columns toward a shared centroid, realizing entropy-aware alignment and thereby **Domain Consistency**, while simultaneously repelling high-entropy (entity-specific) columns toward the periphery, realizing entropy-aware uniformity and thereby **Domain Fidelity**. These guarantees provide the theoretical foundation for the empirical patterns in Figure 5, where schema-stable attributes collapse into a central core while entity-specific attributes disperse outward.

PRELIMINARIES

Let $(\mathcal{T}, \mathcal{F}, P)$ be a probability space over tables, where $\mathcal{T}$ denotes the set of admissible tables, $\mathcal{F}$ is a σ-algebra, and P is a probability measure capturing the empirical distribution of tables. An encoder $f_\theta : \mathcal{T} \to \mathcal{V}$ maps each table $T \in \mathcal{T}$ into a metric space $(\mathbb{R}^d, D)$, where $\mathcal{V}$ denotes the representation space endowed with distance D. We adopt the following assumptions:

(A1) (**Normalization**) All embeddings $E_{\text{seg}}(\cdot)$, $V_{\text{ctx}}(\cdot)$, and $E_{\text{global}}(\cdot)$ are ℓ_2-normalized, i.e., lie on the unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$.

(A2) (**Geometry**) The distance metric is the cosine distance $D(u, v) = 1 - u^\top v$, inducing a geodesic structure consistent with the sphere.

(A3) (**Information-Theoretic Objective**) The InfoNCE loss uses in-batch negative sampling sufficiently dense over rows, approximating a variational lower bound on mutual information.

(A4) (**Optimization**) The temperature $\tau > 0$ is fixed, scaling contrastive forces smoothly.

(A5) (**Entropy Estimation**) Column entropy is estimated from the empirical distribution. Misclassification probability decays exponentially in the number of rows (via large deviation bounds).

ENTROPY-AWARE ALIGNMENT AND UNIFORMITY

Following Wang & Isola (2020), contrastive learning can be understood via two functionals: *alignment*, the expected closeness of positive pairs, and *uniformity*, the spreading of representations across the unit sphere. We adapt these notions by conditioning on column entropy.

Notation. For a column c, let $\mu_c := E_{\text{global}}(h_c)/\|E_{\text{global}}(h_c)\|$ be its normalized global header (centroid). Let $\mathcal{N}_{\text{cent}}(c) = \{\mu_{c'} : c' \neq c\}$ denote centroids of other headers. For high-entropy columns, let $v(r, h_c) := V_{\text{ctx}}(r, h_c)/\|V_{\text{ctx}}(r, h_c)\|$ be the normalized contextual value.

Definition 1 (Domain Consistency (entropy-aware alignment))**.** *For $c \in \mathcal{C}_{\text{low}}$, positives are centroid pairs $(s(r, h_c), \mu_c)$. Define*

$$\mathcal{L}^{\text{low}}_{\text{align}}(f; \alpha) := \mathbb{E}_r \, \|s(r, h_c) - \mu_c\|_2^\alpha.$$

The model is ϵ_{con}-consistent if $\mathcal{L}^{\text{low}}_{\text{align}} \leq \epsilon_{\text{con}}$.

Definition 2 (Domain Fidelity (entropy-aware uniformity))**.** *For $c \in \mathcal{C}_{\text{high}}$, the positive is $(s(r, h_c), v(r, h_c))$ and negatives are $(s(r, h_c), v(r', h_c))$ with $r' \neq r$. Dispersion is measured by*

$$\mathcal{L}^{\text{high}}_{\text{unif}}(f; t) := \log \, \mathbb{E}_{r \neq r'} \exp\big(-t\|s(r, h_c) - s(r', h_c)\|_2^2\big).$$

The model is ϵ_{dom}-faithful if $\mathcal{L}^{\text{high}}_{\text{unif}} \geq -\epsilon_{\text{dom}}$.

Assumptions for Entropy Partition. $\mathcal{C}_{\text{low}} = \{c : H(c) \leq H_0\}$ and $\mathcal{C}_{\text{high}} = \{c : H(c) \geq H_1\}$ with $H_0 < H_1$. For $c \in \mathcal{C}_{\text{low}}$, positives are $(s(r, h_c), \mu_c)$ and negatives are $(s(r, h_c), \mu_c^-)$ with $\mu_c^- \in \mathcal{N}_{\text{cent}}(c)$. For $c \in \mathcal{C}_{\text{high}}$, positives are $(s(r, h_c), v(r, h_c))$ and negatives are $(s(r, h_c), v(r', h_c))$.

Assumption 1 (MI gap – centroid/value forms). There exist $\Delta_{\text{pos}}, \Delta_{\text{neg}} > 0$ s.t.

$$\mathbb{E}\langle s, \mu_c\rangle - \mathbb{E}\langle s, \mu_c^-\rangle \geq \Delta_{\text{pos}} \quad (c \in \mathcal{C}_{\text{low}})$$
$$\mathbb{E}\langle s,\, v(r', h_c)\rangle - \mathbb{E}\langle s,\, v(r, h_c)\rangle \geq \Delta_{\text{neg}} \quad (c \in \mathcal{C}_{\text{high}}).$$

Assumption 2 (Entropy estimation). $\Pr\big(\sup_c |\, \widehat{H}(c) - H(c)\,| \leq C\sqrt{\frac{\log(1/\delta)}{m_c}}\big) \geq 1 - \delta.$

Theorem 1 (Domain Consistency–Fidelity Guarantee)**.** *Suppose Assumptions 1–2 hold and let $\theta^\star$ satisfy $\mathcal{L}_{\text{align}}(\theta^\star) \leq \eta$. Then there exist functions ϕ_1, ϕ_2 with ϕ_i* nondecreasing *in η, τ and* nonincreasing *in B such that*

$$\sup_{c \in \mathcal{C}_{\text{low}}} \mathcal{L}_{\text{align}}^{\text{low}}(f; \alpha) \leq \phi_1(\eta, \tau, B) = \tfrac{1}{\Delta_{\text{pos}}}\, \psi_1(\eta, \tau, B), \tag{3}$$
$$\inf_{c \in \mathcal{C}_{\text{high}}} \mathcal{L}_{\text{unif}}^{\text{high}}(f; t) \geq \phi_2(\eta, \tau, B) = \tfrac{1}{\Delta_{\text{neg}}}\, \psi_2(\eta, \tau, B), \tag{4}$$

where $\psi_i(\eta, \tau, B) = (\tau(\eta - \log B))_+$. Moreover, with prob. $\geq 1 - \delta$, any $\widehat{\theta}$ with $\widehat{\mathcal{L}}_{\text{align}}(\widehat{\theta}) \leq \widehat{\eta}$ and $\|\nabla \widehat{\mathcal{L}}_{\text{align}}(\widehat{\theta})\| \leq \varepsilon$ satisfies the same bounds with $\eta = \widehat{\eta} + \mathfrak{R}_n + O(\varepsilon)$, $\mathfrak{R}_n = O\big(\sqrt{\log(1/\delta)/n}\big)$.

Proof. On $\mathbb{S}^{d-1}$, $D(u, v) = 1 - \langle u, v\rangle$. The population InfoNCE risk for batch B, temperature τ is

$$\mathcal{L}_{\text{align}}(\theta) = \mathbb{E}\left[-\log \frac{e^{\langle s, s^+\rangle/\tau}}{e^{\langle s, s^+\rangle/\tau} + \sum_{j=1}^{B-1} e^{\langle s, s_j^-\rangle/\tau}}\right]. \tag{5}$$

For any $a, b_1, \ldots, b_m \in \mathbb{R}$ and $\tau > 0$, the Softmax–margin inequality (Saunshi et al., 2019) is

$$-\log \frac{e^{a/\tau}}{e^{a/\tau} + \sum_{j=1}^{m} e^{b_j/\tau}} \;\leq\; \frac{1}{\tau} \max_j (b_j - a) + \log(1 + m). \tag{6}$$

(Alignment, contraction of low entropy segments)**.** Apply equation 6 to equation 5 with $a = \langle s, \mu_c\rangle$, $b_j = \langle s, \mu_c^{-(j)}\rangle$ to obtain $\mathbb{E}\langle s, \mu_c\rangle - \mathbb{E}\langle s, \mu_c^-\rangle \geq \tau(\log B - \eta)$. By Assumption 1 and $\|u - \mu\|_2^2 = 2(1 - \langle u, \mu\rangle)$,

$$\mathbb{E}_r \|s(r, h_c) - \mu_c\|_2^2 \;\leq\; \frac{2}{\Delta_{\text{pos}}}\, \psi_1(\eta, \tau, B). \tag{7}$$

(Uniformity, repulsion of high entropy segments)**.** Set $a = \langle s,\, v(r, h_c)\rangle$, $b_j = \langle s,\, v(r_j, h_c)\rangle$; then $\mathbb{E}\langle s,\, v(r', h_c)\rangle - \mathbb{E}\langle s,\, v(r, h_c)\rangle \geq \tau(\log B - \eta)$. Assumption 1 yields

$$\log\, \mathbb{E}_{r \neq r'} \exp\big(-t\|s(r, h_c) - s(r', h_c)\|_2^2\big) \;\geq\; \frac{1}{\Delta_{\text{neg}}}\, \psi_2(\eta, \tau, B). \tag{8}$$

For finite-sample, uniform convergence (Saunshi et al., 2019) gives

$$\sup_\theta \left|\widehat{\mathcal{L}}_{\text{align}}(\theta) - \mathcal{L}_{\text{align}}(\theta)\right| \;\leq\; \mathfrak{R}_n = O\Big(\sqrt{\tfrac{\log(1/\delta)}{n}}\Big) \quad \text{w.p. } \geq 1 - \delta. \tag{9}$$

If $\widehat{\mathcal{L}}_{\text{align}}(\widehat{\theta}) \leq \widehat{\eta}$ and $\|\nabla \widehat{\mathcal{L}}_{\text{align}}(\widehat{\theta})\| \leq \varepsilon$, smoothness implies

$$\mathcal{L}_{\text{align}}(\widehat{\theta}) \;\leq\; \widehat{\eta} + \mathfrak{R}_n + O(\varepsilon). \tag{10}$$

Set $\tilde{\eta} := \widehat{\eta} + \mathfrak{R}_n + O(\varepsilon)$ and substitute $\eta = \tilde{\eta}$ into equation 7 and equation 8. Routing by $\widehat{H}(c)$ and Assumption 2 give a misrouting probability δ_{ent} that vanishes with m_c, so the bounds hold with prob. $\geq 1 - \delta - \delta_{\text{ent}}$. □

Corollary 1 (Entropy-aware Alignment ⇒ Domain Consistency)**.** *With probability at least* $1 - \delta - \delta_{\mathrm{ent}}$*, if* η *is small and* B *is large, then*

$$\mathcal{L}_{\mathrm{align}}^{\mathrm{low}}(f;\alpha) \;\le\; \widetilde{O}\Big(\tfrac{\tau}{\Delta_{\mathrm{pos}} B}\Big),$$

so low-entropy columns contract toward their centroids, ensuring ***Domain Consistency****.*

Corollary 2 (Entropy-aware Uniformity ⇒ Domain Fidelity)**.** *Under the same conditions,*

$$\mathcal{L}_{\mathrm{unif}}^{\mathrm{high}}(f;t) \;\ge\; \widetilde{\Omega}\Big(\tfrac{1}{\tau}\Delta_{\mathrm{neg}}\Big),$$

so high-entropy columns preserve row-level separation, ensuring ***Domain Fidelity****.*

Remark. Taken together, these corollaries formalize the *core–periphery geometry* induced by entropy-driven segment alignment: low-entropy (schema-stable) attributes collapse into a compact core, while high-entropy (entity-specific) attributes are repelled to the periphery. Here $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ suppress polylogarithmic factors in n.

B Related Works

B.1 Structure-aware Encoders

A significant body of research has focused on developing structure-aware encoders, which attempt to explicitly model the 2D layout and relational structure of tables. While foundational, these approaches commonly suffer from two major drawbacks; Inefficiency and Inconsistency.

Inefficiency arises from the architectural complexity required to capture structural cues. These models often introduce significant computational and training overhead. TAPAS Herzig et al. (2020), for example, employs a multitude of embedding layers to encode token roles (e.g., `row_id`, `column_id`, `rank_id`), which is expensive to train. TaBERT Yin et al. (2020) linearizes table content, but its representation is suboptimal; embedding a single cell $\langle i, j \rangle$ requires a minimum of three tokens. For real-world tables with hundreds of columns, this approach quickly becomes infeasible within standard token limits. Other models introduce complexity through architectural choices, such as Tabbie Iida et al. (2021) utilizing two separate transformers, or through intricate encoding schemes. Turl Deng et al. (2022) uses a complex entity representation process involving two role embeddings (type and mention) and a projection layer. Tuta Wang et al. (2021) implements a highly complex positional encoding system with multiple levels of independently learned, tree-based positional encodings, in addition to in-cell positional encodings.

Despite this added complexity, these models fail to achieve robust semantic consistency. Their representations remain vulnerable to simple schema variations, such as column reordering. Furthermore, the embeddings for a given concept can drift semantically depending on the specific query or the context of neighboring entities, indicating a lack of true semantic grounding.

B.2 Domain-aware Encoder

More recently, research has shifted toward domain-aware encoders, which prioritize semantic consistency across different table structures, aiming to capture the "domain" of a column. A notable example is HAETAE Jung & Yoon (2025), which contrasts with structure-aware models by using a simpler, lightweight approach. It uses a standard BERT backbone but integrates an additional embedding layer for row context-free header tokens. Haetae trains this universal header embedding using a distance-based objective, which explicitly forces headers with the same semantic meaning (e.g., "First Name" and "f_name") to have similar representations.

While this method successfully ensures header consistency, it introduces a critical limitation: Header-value Misalignment. By forcing header representations to be close while neglecting the semantic information contained in the cell values, the model harms the crucial header-value dependencies. This optimization for header-level consistency weakens the model's ability to perform deep table reasoning. The resulting consistency is not truly grounded in the full domain semantics of the table, as it largely ignores the values, which are essential for defining that domain.

B.3 Task-aware Approaches

A line of research focuses on task-specific pretraining, adapting language models to address the heterogeneity of tabular attributes for supervised prediction. TP-BERTa (Yan et al.), for example, is designed explicitly for regression and classification, introducing relative magnitude tokenization and intra-feature attention to reconcile numerical values with feature semantics, thereby competing with strong tree-based and deep tabular baselines. Complementary paradigms expand task awareness in different directions: TAPEX (Liu et al.) pretrains on SQL execution to enhance table QA, while TabPFN (Hollmann et al.) uses synthetic priors for probabilistic classification without finetuning. More recent work pushes toward broader reasoning capabilities, including modular table reasoning with TAPERA (Zhao et al., 2024), instruction-tuned multi-task alignment in Table-GPT (Li et al., 2024), and generative modeling with CDTD (Mueller et al., 2023) for mixed-type imputation. Collectively, these efforts highlight a shift toward tailoring pretraining to specific downstream tasks—whether predictive modeling, QA, or imputation—though such specialization often comes at the cost of limited transferability across domains requiring general-purpose table understanding.

C Implementation Details

C.1 Architectural Details

Global Header Encoder The header encoder is implemented as a lightweight BERT-based module that generates context-independent embeddings for header strings. The encoder utilizes a frozen BERT tokenizer and embedding layer, followed by two transformer layers (layers 8 and 9 from the pretrained BERT model) to capture semantic representations of header names.

The design choice of using two layers strikes a balance between expressivity and efficiency: a shallow encoder reduces computational overhead while still allowing non-trivial contextualization beyond the embedding layer. Using more layers risks overfitting to sentence-level semantics irrelevant for headers, while fewer layers (e.g., only one) limit the ability to model compositional structure.

The selection of layers 8 and 9 is grounded in empirical analysis of BERT (Clark et al., 2019) shows that mid-to-deep layers (approximately layers 7–10) specialize in syntactic dependencies and head–dependent relations, such as determiners linking to nouns and direct objects linking to verbs. By contrast, earlier layers capture mostly local or lexical information, while the final layers (11–12) are biased toward [CLS]-based sentence aggregation and task-specific adaptation. Leveraging layers 8 and 9 thus provides a strong inductive bias for modeling headers, which are typically short noun phrases requiring syntactic but not full discourse-level context.

Given a header string h, the encoder first tokenizes it using the BERT tokenizer, then processes the tokens through the embedding layer to obtain initial representations. These embeddings are passed through two sequential transformer layers with self-attention mechanisms:

$$\mathbf{h}^{(0)} = \text{BertEmbeddings}(\text{tokenize}(h))$$
$$\mathbf{h}^{(1)} = \text{EncoderLayer}_8(\mathbf{h}^{(0)})$$
$$\mathbf{h}^{(2)} = \text{EncoderLayer}_9(\mathbf{h}^{(1)})$$

The final universal header embedding $E_{\text{global}}(h)$ is obtained through mean pooling over the sequence dimension, weighted by the attention mask to exclude padding tokens:

$$E_{\text{global}}(h) = \frac{\sum_{i=1}^{n} \mathbf{h}_i^{(2)} \cdot \text{mask}_i}{\sum_{i=1}^{n} \text{mask}_i}.$$

The encoder supports flexible input formats, handling single header strings, flat lists of headers, or batched lists, automatically adjusting the output dimensionality and providing appropriate masking for batch processing.

Projection Layer for Segments The segment projection network $g(\cdot)$ implements the transformation that combines universal header embeddings, contextualized header representations, and contextualized value representations into unified segment embeddings. Motivated by projection layers in transformer-based language models (Vaswani et al., 2017), the architecture adopts a two-stage feedforward block with residual connections and normalization, enabling non-linear feature mixing while maintaining training stability.

Given the three input components $E_{\text{global}} \in \mathbb{R}^{B\times H\times D}$, $H_{\text{ctx}} \in \mathbb{R}^{B\times H\times D}$, and $V_{\text{ctx}} \in \mathbb{R}^{B\times H\times D}$, the projection first concatenates them along the feature dimension:

$$\mathbf{x}_{\text{concat}} = [E_{\text{global}} \parallel H_{\text{ctx}} \parallel V_{\text{ctx}}] \in \mathbb{R}^{B\times H\times 3D}$$

The concatenated representation is then processed through a two-layer feedforward network with GELU activation and layer normalization:

$$\mathbf{x}_{\text{hidden}} = \text{LayerNorm}(\text{GELU}(\text{Linear}3D \to 2D(\mathbf{x}_{\text{concat}})))$$
$$s(r,h) = \text{LayerNorm}(\text{Linear}2D \to D(\text{Dropout}(\mathbf{x}_{\text{hidden}})))$$

This design mirrors the intermediate expansion–compression scheme used in transformers, where increasing dimensionality allows richer interactions between features before reducing back to the model dimension for compatibility. By concatenating schema-level and row-level signals, the projection network learns to fuse global header semantics with local contextual patterns. The residual normalization ensures stable optimization, while the intermediate $2D$ bottleneck provides sufficient capacity to capture complex header–value dependencies.

C.2 Dataset Preprocessing

Our dataset preprocessing pipeline is designed to optimize the quality and compatibility of tabular data for BERT-based language model training. The preprocessing consists of three main stages: data cleaning, BERT vocabulary validation, and tokenization optimization.

Data Cleaning and Normalization The raw tabular data undergoes several cleaning steps to ensure consistency and quality. First, we flatten nested JSON structures. For example:

```
"actors": [{"name": "allan"}, {"name": "daniel"}] →
"actors.0.name": "allan", "actors.1.name": "daniel"
```

This creates a uniform representation where each row is represented as a flat dictionary of key-value pairs. This flattening process preserves the hierarchical structure through dot-separated keys.

Next, we handle indexed fields that represent repeated attributes. To prevent information overload and maintain computational efficiency, we sample a maximum of 3 indexed fields per field type, prioritizing the first occurrences to maintain data consistency.

BERT Vocabulary Validation A critical challenge in training BERT on multilingual tabular data is the model's limited vocabulary coverage for non-English languages. To address this, we implement a BERT vocabulary validation step that filters out tables containing content that cannot be effectively tokenized by the BERT tokenizer.

For each table, we extract meaningful text fields (excluding URLs, pure numbers, and very short strings) and tokenize them using the BERT tokenizer. We calculate the ratio of unknown tokens ([UNK]) to total tokens for each field. Tables where more than 30% of the text fields contain excessive unknown tokens (threshold: 30% UNK ratio) are excluded from training. This filtering ensures that the model trains on data it can meaningfully process, significantly reducing the proportion of uninformative [UNK] tokens during training.

Tokenization Optimization Finally, to maximize the utility of the remaining data while respecting BERT's token limit constraints, we implement field-level truncation: Individual fields that exceed 20 tokens are truncated to fit within this limit, preserving the most important information while maintaining field names and separators.

Preprocessing Statistics Our preprocessing pipeline processes data from 100 different e-commerce websites across multiple languages and domains. The BERT vocabulary validation step typically filters out 60-70% of rows containing significant non-English content, resulting in a dataset focused on English-language e-commerce data that can be effectively processed by BERT.

The final preprocessed dataset maintains the structural information of the original tables while ensuring compatibility with BERT's tokenization scheme, enabling effective representation learning for tabular data through masked language modeling objectives.

This approach addresses the fundamental challenge of applying English-centric language models to multilingual structured data, ensuring that the training process focuses on content that the model can meaningfully learn from while preserving the rich structural information inherent in tabular data.

C.3 Training Procedure

Batch Construction. For each domain, we organize the 100 tables into stratified batches using a hierarchical grouping strategy. Specifically, tables are grouped into sets of four (25 groups per domain), with all rows in a group merged into a unified dataset. An epoch processes all groups sequentially, with group order shuffled each time while preserving the 4-table grouping for computational efficiency. Within each group, stratified sampling assigns batch slots in proportion to table size: $\text{batch\_count}_{t_i} = \max\left(1, \text{round}\left(n_i/N \times \text{batch\_size}\right)\right)$, for table t_i with n_i rows out of N. This procedure balances representation across tables, prevents larger tables from dominating training, and ensures that even small tables contribute consistently to every batch.

Algorithm 1 NAVI Training Procedure

Require: Domain $\mathcal{D}$ with tables $\{t_i\}_{i=1}^{100}$, model $\mathcal{M}_\theta$, alignment weight λ_{align}, masking configuration MaskCfg
Ensure: Trained parameters θ^*

```
Initialize θ, optimizer, gradient scaler
for epoch t = 1, ..., T do
    Partition 100 tables into groups G = {G_1, ..., G_25}, |G_i| = 4
    Shuffle group order
    for each group G ∈ G do
        for each table t_i ∈ G do
            Compute normalized entropy per field: H_norm(f) = −Σ_{v∈V_f} p(v) log p(v) / log |V_f|
            Categorize fields: H_dom = {f : H_norm(f) ≤ Q_1},   H_ent = {f : H_norm(f) ≥ Q_3}
            Initialize stratified sampler, Sampler(R_G)
        end for
        for each batch B ~ Sampler(R_G) do
            Apply masking Mask(·; MaskCfg): x̃_b, y_b = mask(x_b; MaskCfg)
            Forward (masked): logits, L_b = M_θ(x̃_b)
            Forward (unmasked): embeddings, E_b = M_θ(x_b)
            Extract components: {E_global, H_ctx, V_ctx} = extract(E_b)
            Segment fusion: s(r, h) = g(E_global(h) ‖ H_ctx(r, h) ‖ V_ctx(r, h))
            Compute losses: L^(b)_msm, L^t_dom, L^t_ent
            Total loss: L^(b)_total = L^(b)_msm + λ_align · (1/|B|) Σ_{t∈B} (L^t_dom + L^t_ent)
            Update: θ ← step(θ, L^(b)_total)
        end for
    end for
end for
return M_θ*
```

Entropy-based Column Categorization Following the entropy-based categorization described in Section 2.3, we compute normalized Shannon entropy for each field f in table t:

$$H_{\text{norm}}(f) = \frac{-\sum_{v \in V_f} p(v) \log_2 p(v)}{\log_2 |V_f|}$$

where V_f is the set of unique values for field f, and $p(v)$ is the probability of value v. The categorization uses quartile-based thresholds computed per table:

- Domain-coherent columns $\mathcal{H}_{\text{dom}}$: $H_{\text{norm}}(f) \leq Q_1$,
- Entity-discriminative columns $\mathcal{H}_{\text{ent}}$: $H_{\text{norm}}(f) \geq Q_3$,

where domain-coherent columns represent stable, low-entropy fields capturing global domain semantics (e.g., genre), entity-discriminative columns represent high-entropy fields that vary across rows, capturing instance-specific attributes (e.g., title). This per-table categorization ensures robust field classification regardless of table size or domain characteristics, with minimum guarantees of at least one field per category when possible. The categorization is computed once per combined dataset and used throughout the training of that group.

Masking Configuration. Building on the structure-aware MSM framework in Section 2.2, we define three masking regimes with token budget control: (1) Header–Value (HV) Masking: Selects k header and value segments under a total budget $\frac{\texttt{max\_tokens}}{\texttt{token\_length\_threshold}}$, split by a configurable ratio (default: 50% values, 50% headers), with each segment contributing up to 8 tokens to $\mathcal{M}_h$ or $\mathcal{M}_v$. (2) BERT-style (B) Masking: Standard MLM regime with 15% uniform masking over non-special tokens to form $\mathcal{M}r$. (3) Hybrid (HVB) Masking: Combines the two by allocating $w_{hv} \times \texttt{max\_tokens}$ (default $w_{hv} = 0.5$) to HV masking and the remainder to BERT-style masking. All regimes follow the usual replacement scheme (80% [MASK], 10% random, 10% unchanged).

Forward Pass and Loss Computation. The forward pass follows the semantic-aware schema induction framework (Section 2.1) and enriched by universal header embeddings described in Appendix C.1. For each batch, the model performs two passes: (1) Masked input → MSM logits for structure-aware segment modeling; (2) Unmasked input → contextualized embeddings for entropy-aware contrastive alignment. From the unmasked pass, we extract universal header embeddings $E_{\text{global}}(h)$, contextualized header representations $H_{\text{ctx}}(r, h)$, and value representations $V_{\text{ctx}}(r, h)$, which are fused via the projection network $g(\cdot)$ into segment embeddings $s(r, h)$. The total loss combines structure-aware MSM and entropy-aware contrastive alignment: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{msm}} + \lambda_{\text{align}} \cdot \mathcal{L}_{\text{align}}$, where $\mathcal{L}_{\text{msm}}$ is computed over the masked sets $\mathcal{M}$, and $\mathcal{L}_{\text{align}} = \mathcal{L}^t_{\text{dom}} + \mathcal{L}^t_{\text{ent}}$ jointly enforces domain consistency (cross-header alignment) and domain fidelity (cross-row alignment).

D Hyperparameter Sensitivity Analysis

We analyze the sensitivity of NAVI to key hyperparameters: the alignment weight λ_{align}, the masking ratio in the structure-aware MSM (*h:v:b*), and the alignment temperature τ. As described in Section 3.1, our default configuration is $\lambda_{\text{align}} = 0.5$, *h:v:b*= 4:4:2, and $\tau_{\text{dom}} = 0.13$ (with $\tau_{\text{ent}} = 0.07$). For λ_{align}, we also test values exponentially ($2^{-2}, 2^{-1}, 2^0, 2^1, 2^2$). For *h:v:b*, with 10 segments selected for masking, we test header-biased (3:1), balanced (2:2), and value-biased (1:3) splits, along with BERT-heavy (4:6) and header–value-heavy (8:2) strategies. For τ, we vary $\tau_{\text{dom}} \in \{0.07, 0.1, 0.13\}$ to examine whether higher temperatures yield smoother alignment distributions that better capture schema-level consistency. Since these hyperparameters primarily affect fidelity-oriented objectives, we restrict our analysis to fidelity tasks. Table 5 summarizes the results, which we discuss below with domain-wise breakdowns.

Table 5: Performance of hyperparameter-tuned variants on downstream fidelity tasks. Evaluation includes Value Imputation (Val; accuracy), Row Classification with XGBoost (XGB; F1-Macro) and Logistic Regression (LR; F1-Macro), and Row Clustering using Agglomerative with Silhouette coefficient (Ag-Sil) and B-cubed F1 (Ag-B^3). Best results per task are in bold, and second best results are underlined.

	Product					**Movie**				
	Val	XGB	LR	Ag-Sil	Ag-B^3	Val	XGB	LR	Ag-Sil	Ag-B^3
Default	0.741	**0.355**	**0.417**	<u>0.069</u>	0.273	**0.641**	0.275	0.313	<u>0.300</u>	<u>0.233</u>
λ_{align}										
0.25	**0.752**	0.344	0.335	0.051	0.267	0.613	0.285	0.315	0.157	0.213
1.0	0.729	<u>0.353</u>	0.377	0.074	0.273	0.608	0.267	0.293	0.278	0.225
2.0	0.741	0.282	0.369	0.053	<u>0.278</u>	0.617	**0.303**	0.294	0.142	0.222
4.0	0.731	0.269	0.321	0.111	0.258	0.603	0.287	0.304	**0.346**	0.230
h:v:b										
2:2:6	0.742	0.343	0.403	0.075	0.266	0.611	0.260	0.285	0.133	0.220
3:1:6	0.738	0.273	0.323	0.046	0.244	0.590	0.282	0.306	0.264	**0.236**
1:3:6	0.742	0.309	0.378	0.049	0.222	0.612	0.264	<u>0.314</u>	0.117	0.224
6:2:2	0.740	0.322	0.404	0.111	**0.314**	0.598	0.290	0.312	0.232	0.203
2:6:2	<u>0.743</u>	0.308	0.384	0.053	0.247	<u>0.618</u>	0.281	0.313	0.096	0.205
τ_{dom}										
0.1	0.739	0.271	0.248	**0.096**	0.210	0.611	<u>0.293</u>	**0.330**	0.208	0.213
0.07	0.733	0.326	0.388	0.073	0.264	0.622	0.288	0.293	0.243	0.224

Effect of λ_{align}. Performance remains stable across λ_{align}, but extremes reveal its role in balancing alignment and schema–value grounding. Too low (0.25) weakens entropy-aware alignment, hurting clustering, while too high (4.0) over-regularizes, suppressing value dependencies and degrading classification. Intermediate settings (1.0–2.0) provide the most stable trade-off. On Product, $\lambda_{\text{align}} = 2.0$ yields the best fidelity, indicating stronger alignment regularization helps under larger schema variability. On Movie, $\lambda_{\text{align}} = 1.0$ is optimal, with 2.0 close behind, suggesting lighter alignment suffices when schemas are more homogeneous. This asymmetry shows how schema diversity dictates the balance between alignment and MSM.

Effect of *h:v:b*. Masking allocation has a pronounced impact on schema–value coupling. Allocating more budget to header–value tokens (header–value-heavy) consistently strengthens fidelity, while BERT-heavy settings (high b) provide weaker schema anchoring. On Product, 6:2:2 produces the strongest fidelity overall, with 2:2:6 next best, confirming explicit schema–value masking is critical when tables exhibit wide header variability. On Movie, no single configuration dominates across tasks: 3:1:6 excels in clustering, 1:3:6 in classification, and 2:6:2 in value imputation. Taken together, 3:1:6 is most reliable, with 2:6:2 competitive. These mixed outcomes suggest Movie schemas benefit from balanced or header-biased allocation, while Product requires stronger header–value masking. Importantly, this shows task type (classification vs. clustering vs. imputation) interacts with masking strategy, underscoring the need for adaptive masking.

Effect of τ_{dom}. Varying τ_{dom} shows its influence on the smoothness of alignment distributions. Higher values (0.1) encourage softer alignments that sometimes aid row classification but reduce clustering precision, while lower values (0.07) sharpen schema alignment. On Product, $\tau_{\text{dom}} = 0.07$ strikes the best balance, reflecting the need for sharper separation in heterogeneous schemas. On Movie, $\tau_{\text{dom}} = 0.1$ performs best, with 0.07 close behind, suggesting that smoother alignments better capture consistency across more uniform schemas. This indicates that temperature tuning is domain-dependent: heterogeneous domains demand sharper distinctions, whereas homogeneous domains benefit from softer, domain-coherent clustering.