Permutation-Invariant Row Abduction for Per-Dataset Table Learning

Heyang Gong

Model-seed working manuscript v0.2 — July 30, 2026

MODEL-SEED MANUSCRIPT. This draft fixes the network, symmetry, and train–inference contracts only. It contains no implementation, benchmark result, predictive-superiority claim, novelty claim, foundation-model claim, or causal claim.

Abstract

We specify a table learner fitted separately to each fixed-schema dataset. A train-fitted adapter first turns every visible raw cell into a scalar. One cell encoder, shared across all rows and features, maps that scalar to a cell token; a learned dataset-specific feature token is then added directly to identify what was measured. A position-free Transformer and a symmetric mean-and-count readout infer one dynamic representation from the resulting unordered evidence set. The same representation can answer several target-feature queries in parallel through a recommender-like row–target pair function and type-routed proper distribution heads. Training removes observed query values and uses them only as likelihood labels. Inference freezes all parameters and encodes an unseen row without a row-ID lookup. Because self-attention is permutation equivariant without position or slot encodings and the readout is symmetric, jointly reordering evidence values, semantic feature keys, and masks leaves the inferred row representation and prediction unchanged. The construction supports same-schema new-row masked prediction; it does not by itself support unseen schemas, cross-dataset transfer, recovery of a persistent Individual, or causal queries.

1 Architecture

1.1 The fixed-schema conditional-set object

Let a dataset $\mathcal{D}$ have d named features. Let $M_{it} = 1$ mean that cell (i, t) is actually recorded and $M_{it} = 0$ mean natural missingness, and define $J_i^O = \{t : M_{it} = 1\}$. A complete table is the special case $M_{it} = 1$ for every row–feature pair. During training, an episode chooses a query set $J_i^Q \subseteq J_i^O$, then defines the disjoint evidence set

$$J_i^E = J_i^O \setminus J_i^Q, \qquad J_i^E \cap J_i^Q = \varnothing. \tag{1}$$

The raw evidence and its tokenized counterpart are different objects:

$$\mathcal{X}_i^E = \{(j, x_{ij}^{\text{raw}}) : j \in J_i^E\}, \tag{2}$$

$$\mathcal{E}_i = \{e_{ij} : j \in J_i^E\}. \tag{3}$$

The model learns a feature-indexed family of conditional set predictors,

$$F_{\Theta_\mathcal{D}} : (\mathcal{X}_i^E, t) \longmapsto q_{\Theta_\mathcal{D}, t}(\cdot \mid \mathcal{X}_i^E), \qquad t \in J_i^Q. \tag{4}$$

It is fitted separately for this dataset. It is not a row-embedding table and is not, without further assumptions, a joint, causal, or Individual-level model.

The trainable parameters are

$$\Theta_{\mathcal{D}} = \{V_{\mathcal{D}}, g_\phi, T_\theta, \rho, h_\varnothing, \phi_{\text{pair}}, g_{\text{num}}, g_{\text{bin}}, s_{\text{cat}}\}. \tag{5}$$

All are shared across rows; all are learned only from the training split of $\mathcal{D}$. Feature-specific preprocessing adapters are fitted training state, not separate neural cell encoders.

1.2 From a raw cell to one evidence token

Each feature has a learned item-like token

$$v_j \in \mathbb{R}^p, \qquad V_{\mathcal{D}} = [v_1; \ldots; v_d] \in \mathbb{R}^{d \times p}. \tag{6}$$

The lookup key is semantic feature identity, not the feature's current display position. Consequently, the architecture uses no column-position embedding.

A train-fitted adapter first maps the raw dtype into one scalar,

$$\xi_{ij} = a_j(x_{ij}^{\text{raw}}) \in \mathbb{R}. \tag{7}$$

Continuous values can be standardized, Boolean values receive fixed codes, and scalar-compatible ordinal values receive a stable train-local code. The same neural encoder is then applied to every adapted cell in every feature:

$$c_{ij} = g_\phi(\xi_{ij}) \in \mathbb{R}^p. \tag{8}$$

A minimal realization is a two-layer scalar-to-vector MLP with LayerNorm and GELU after each affine layer. Crucially, g_ϕ does not read j or v_j: by itself, c_{ij} says what scalar was observed, not which feature produced it.

Feature identity is attached by direct addition,

$$\boxed{e_{ij} = c_{ij} + v_j} \in \mathbb{R}^p. \tag{9}$$

There is no feature-specific value MLP and no extra interaction MLP before the set encoder. A value such as 42 can therefore share a value geometry across cells while age = 42 and temperature = 42 remain distinct through their different feature tokens.

How to read Fig. 1. Band A separates value content from feature identity and then adds them. Band B turns an unordered, variable-size evidence set into a target-independent dynamic row token. Band C performs the recommender-like operation: the dynamic row token is paired with each requested target-feature token, without rerunning the row encoder for targets that share the same evidence set.

1.3 Permutation-invariant row abduction

For row i, arrange the $k_i = |J_i^E|$ evidence tokens in an arbitrary matrix $R_i \in \mathbb{R}^{k_i \times p}$. A position-free Transformer produces

$$H_i = T_\theta(R_i) \in \mathbb{R}^{k_i \times p}. \tag{10}$$

It has no absolute position, column-slot encoding, or order-dependent concatenation. Feature identity is already carried by e_{ij}. This is the set-attention portion of the construction; related invariant and equivariant set architectures include Deep Sets and Set Transformer [3, 5].

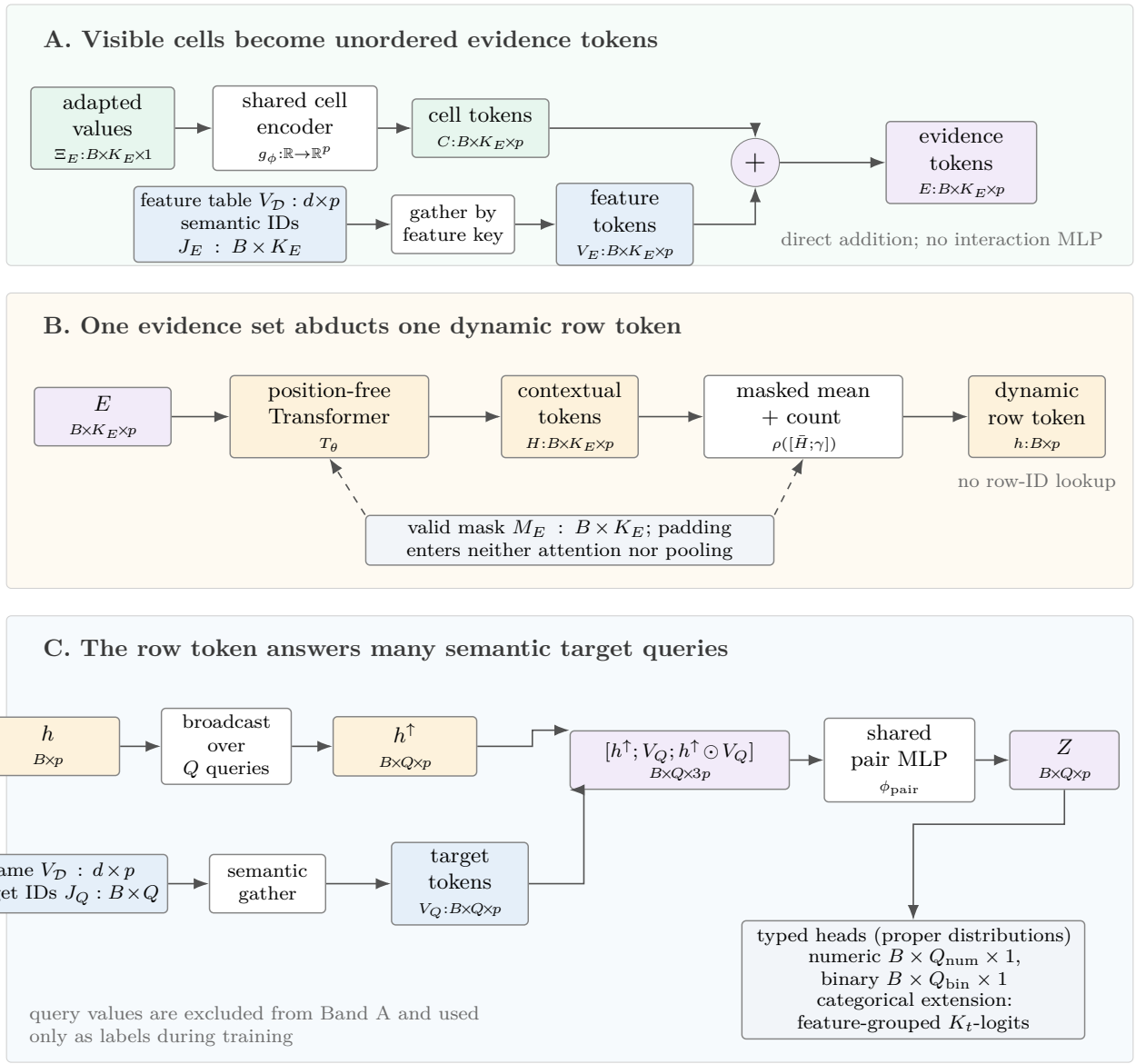

Figure 1: Tensor flow of the per-dataset learner (Bands A→B→C). Symbols: B batch size, $K_E \leq d$ padded evidence width, Q target queries, d schema width, p hidden width. Semantic feature IDs gather feature tokens (not column positions). A single position-free evidence pass yields h, reused for all Q targets. Padding is ignored by attention and pooling; categorical outputs are ragged by target feature.

The mask-aware symmetric readout is

$$\bar{h}_i = \frac{1}{k_i} \sum_{\ell=1}^{k_i} H_i^{(\ell)}, \qquad \gamma_i = \frac{\log(1 + k_i)}{\log(1 + d)}, \tag{11}$$

$$h_i(\mathcal{E}_i) = \rho([\bar{h}_i; \gamma_i]) \in \mathbb{R}^p. \tag{12}$$

When $k_i = 0$, the model uses a learned $h_\varnothing$ instead of an empty mean. We use "Abduct map" as

shorthand for the composition

$$A_\theta := \rho \circ [\mathrm{Mean}, \mathrm{Count}] \circ T_\theta, \qquad h_i(\mathcal{E}_i) = A_\theta(\mathcal{E}_i), \tag{13}$$

not as an additional encoder. The result is an evidence-dependent predictive compression, not a persistent row identity. Mean pooling may lose information; the count channel removes only one obvious ambiguity and does not make the map invertible.

1.4 Target-feature queries and proper distributions

For target t, the same row token is combined with target token v_t:

$$z_{it} = \phi_{\mathrm{pair}}([h_i(\mathcal{E}_i); v_t; h_i(\mathcal{E}_i) \odot v_t]) \in \mathbb{R}^p. \tag{14}$$

The Hadamard term exposes a direct recommender-style interaction while the MLP can learn non-bilinear corrections. The target token is absent from the row encoder: one $h_i(\mathcal{E}_i)$ can therefore serve all queries sharing $\mathcal{E}_i$. Unlike ordinary ID-based matrix factorization [2], no persistent row lookup is available.

For a standardized numeric target, the reference head predicts one rating-like score and assigns a fixed-variance Gaussian observation model:

$$\mu_{it} = g_{\mathrm{num}}(z_{it}), \tag{15}$$

$$q_{\Theta_\mathcal{D},t}(\widetilde{X}_{it} \mid \mathcal{X}_i^E) = \mathcal{N}(\mu_{it}, 1). \tag{16}$$

Its negative log-likelihood is squared error up to an additive constant, as in explicit-rating factor models [2]. Learning a scale or a heteroscedastic variance is a richer extension, not part of the reference head. A binary target uses a Bernoulli head,

$$q_{\Theta_\mathcal{D},t}(X_{it} = 1 \mid \mathcal{X}_i^E) = \mathrm{sigmoid}(g_{\mathrm{bin}}(z_{it})). \tag{17}$$

The declared target type routes each cell to one normalized likelihood rather than mixing unrelated classification and regression losses for that cell.

Nominal categorical encoding is intentionally not frozen in the formal core. One scalar-coded baseline uses train-local category code $a_t(k)$, candidate cell token $c_{tk} = g_\phi(a_t(k))$, and

$$S_{itk} = s_{\mathrm{cat}}(z_{it}, c_{tk}), \tag{18}$$

$$q(X_{it} = k \mid \mathcal{X}_i^E, t) = \frac{\exp S_{itk}}{\sum_{\ell=1}^{K_t} \exp S_{it\ell}}. \tag{19}$$

This exact, feature-local softmax is the reference categorical output. It preserves the shared cell encoder but imposes an artificial geometry on nominal codes. A feature-local candidate-token alternative, category-relabeling tests, and computational approximations for unusually high-cardinality targets remain open design branches. Such approximations would sample candidate values within feature t, not unobserved cells or other features as negative interactions. Pairwise ranking objectives are not part of the reference model. Since K_t varies by target, categorical logits are feature-grouped ragged outputs rather than a fictitious common $B \times Q \times K$ tensor.

1.5 Exact symmetry and semantic boundary

Proposition 1 (Evidence-order invariance)**.** *Let P jointly permute an evidence row's token matrix and valid-token mask. If the Transformer has no positional or fixed-slot input, then $T_\theta(PR) = PT_\theta(R)$, and*

$$h_i(P\mathcal{E}_i) = h_i(\mathcal{E}_i). \tag{20}$$

Consequently, the predictive distribution for the same semantic target t is unchanged. A joint relabeling of target features permutes the corresponding prediction list equivariantly.

Proof. Self-attention without positional or slot encodings commutes with a permutation of token rows. The output rows are therefore permuted by the same P, while their masked mean and valid count are unchanged. The deterministic readout receives the same input; so does every target head for fixed v_t. □

The symmetry removes presentation order, not feature identity. A legitimate permutation moves the adapted value, semantic feature key, and padding mask together. Pairing a value with another feature's key changes the evidence and is not a symmetry. Exact mathematical invariance can still appear as tiny roundoff differences in floating-point kernels and should be tested under a declared numerical tolerance.

2 Training and Inference

2.1 Dense-table cell risk and basic typed losses

The core regime is a complete or nearly complete table, not a positive-only user–item interaction log. Every actually recorded cell supplies an explicit response value. Training creates a query by artificially hiding such a value; natural missingness supplies no value label. Let

$$\begin{aligned} O_t &= \{i : x_{it} \text{ is observed in the training split}\}, \\ \mathcal{T}_{\text{train}} &= \{t : O_t \neq \varnothing \text{ and a proper head for } t \text{ is defined}\}. \end{aligned} \tag{21}$$

The unambiguous v0.2 reference sampler first draws a feature uniformly from $\mathcal{T}_{\text{train}}$, then draws a row uniformly from O_t, sets $J_i^Q = \{t\}$, and removes x_{it} before any cell token is computed. Optional evidence dropout can remove additional non-target cells to vary the amount of evidence; it does not manufacture negative responses. For target type $\tau(t) \in \{\text{num}, \text{bin}, \text{cat}\}$, the reference model uses the basic explicit-response losses

$$\ell_{it} = \begin{cases} \frac{1}{2}(\widetilde{x}_{it} - \mu_{it})^2, & \tau(t) = \text{num}, \\ -x_{it} \log p_{it} - (1 - x_{it}) \log(1 - p_{it}), & \tau(t) = \text{bin}, \\ -S_{it,x_{it}} + \log \sum_{k=1}^{K_t} \exp S_{itk}, & \tau(t) = \text{cat}, \end{cases} \tag{22}$$

where $p_{it} = \text{sigmoid}(g_{\text{bin}}(z_{it}))$. These are standardized mean-squared error, binary cross-entropy, and full-softmax cross-entropy. Equivalently, they are the negative log-likelihoods of the fixed-variance Gaussian, Bernoulli, and categorical heads defined in Section 1. They instantiate basic explicit numeric-rating, pointwise binary-response, and one-of-K_t choice models familiar from recommendation [1, 2]; they are not copied from an undisclosed LimiX multi-head objective.

Suppressing the optional evidence-dropout expectation, the observed-cell risk is

$$\mathcal{L}_{\mathcal{D}}(\Theta_{\mathcal{D}}) = \frac{1}{|\mathcal{T}_{\text{train}}|} \sum_{t \in \mathcal{T}_{\text{train}}} \frac{1}{|O_t|} \sum_{i \in O_t} \ell_{it}. \tag{23}$$

Feature-first, row-second sampling is an unbiased minibatch estimator of this risk. When the table is complete, O_t contains all training rows for every feature, so Eq. (23) is simply the uniform average over all recorded cells. No confidence weighting, unobserved-item negative sampling, BPR, or catalog-ranking term is needed; those techniques address sparse implicit feedback rather than dense typed cell prediction [4].

An observed binary zero, $M_{it} = 1, x_{it} = 0$, is a valid negative response. It is categorically different from $M_{it} = 0$, which contributes no value loss. Artificial query masks are never encoded as natural-missing tokens; natural missing cells and batch padding are excluded by default. Modelling M_{it} itself would require a separate missingness head and measurement contract. Loss magnitudes can still differ by response type, so evaluation must retain per-feature and per-type metrics.

The model also permits a nonempty multi-target set J_i^Q. All its targets share one evidence set and one row token:

$$\mathcal{L}_i = \frac{1}{|J_i^Q|} \sum_{t \in J_i^Q} \ell_{it}. \tag{24}$$

The query sampler is part of feature weighting and must be reported explicitly. The multi-target average consists of conditional marginal losses; it does not create a normalized joint likelihood over the whole masked block.

2.2 Per-dataset parameter fitting

All components in Eq. (5) are optimized jointly on $\mathcal{D}$: the feature-token table, one shared cell encoder, the position-free Transformer, the symmetric readout, the row–target pair MLP, and type-routed score heads. There is no free row-ID embedding. Training rows influence the model through shared parameters and masked cell losses, not through a table of persistent row vectors. A held-out row therefore resembles a cold-start user only in the narrow sense that no row lookup is available. Unlike an implicit-feedback recommender, the features do not compete as a retrieval catalog and recorded cells provide explicit typed responses. Scalar adapters, including centers, scales, and any train-local codes, are also fitted on the training split only and then frozen.

The current proposal does not prescribe an optimizer, masking curriculum, batch size, or regularizer. Those choices remain experiment-level seeds rather than parts of the mathematical network contract.

2.3 Frozen-parameter new-row inference

For a new row under the same schema, inference freezes $\Theta_{\mathcal{D}}$ and all train-fitted adapter state. It then performs one evidence pass,

$$\mathcal{X}_i^E \xrightarrow{a_j, g_\phi, +v_j} \mathcal{E}_i \xrightarrow{T_\theta, \text{Mean+Count}} h_i(\mathcal{E}_i), \tag{25}$$

followed by parallel semantic target queries,

$$\{v_t : t \in J_i^Q\},\ h_i(\mathcal{E}_i) \longrightarrow \{q_{\Theta_{\mathcal{D}},t}(X_{it} \mid \mathcal{X}_i^E) : t \in J_i^Q\}. \tag{26}$$

Table 1: Training and inference differ in what is optimized, while sharing the same masked-query network.

	Training	New-row inference
Parameters	Optimize $\Theta_{\mathcal{D}}$ on training rows	Freeze $\Theta_{\mathcal{D}}$
Row representation	Computed after sampled queries are removed	Computed once from the current shared evidence set
Row ID	Never used as a parameter lookup	Not required
Target value	Supplies the likelihood term only	Unknown and excluded from abduction
Target queries	Singleton by default; multi-target is explicit	Parallel when targets share one evidence set
Schema	Fixed for $\mathcal{D}$	Must match the fitted schema

The new row requires neither a stored ID, a new embedding-table entry, gradient updates, nor a fixed ordering or count of visible cells. As more cells become visible, its dynamic row token is recomputed from the enlarged evidence set.

One pass can serve several targets only when they share the same $\mathcal{X}_i^E$. Computing every leave-one-out conditional $q_t(X_{it} \mid X_{i,-t})$ changes the evidence set for each t, so it requires distinct set-encoder passes (or their expansion along the batch dimension). Simultaneous multi-target outputs are conditional marginals; v0.2 does not claim their product is a learned joint distribution.

2.4 Supported and unsupported generalization

The architectural contract supports same-schema generalization from training rows to unseen rows, variable-size evidence sets, and arbitrary presentation orders of correctly paired values, semantic feature keys, and masks. It also supports querying different target features through their target tokens and likelihood heads.

It does not automatically support a new dataset, an unseen feature, a changed measurement contract, or zero-shot transfer across schemas. Dataset-specific feature tokens and train-fitted adapters must exist before a feature can be used. Nor does masked likelihood identify a natural missingness mechanism, recover a real Individual, or license intervention, counterfactual, and causal-mechanism claims. These are separate scientific objects, not consequences of reconstruction accuracy.

LimiX motivates the construction of context-conditional masked queries, not the choice of the three basic response losses in Eq. (22), and it trains one cross-dataset model with a different adaptation contract [6]. The present architecture instead makes the per-dataset feature tokens and inferred row representation explicit and requires single-forward evidence-order invariance by construction. Whether that combination is scientifically distinct or empirically useful remains an open evidence gate.

A Three Reconstruction Objectives

The main text fixes the simplest objective for the proposed per-dataset learner: observed cells are temporarily hidden and scored by mean-squared error, binary cross-entropy, or full-softmax cross-entropy according to their declared response type. This appendix separates that reference choice from two different modeling objects that can also be called reconstruction. The three objectives

share a masked-query interface, but they do not learn the same conditional object and are not interchangeable implementations of one loss.

Table 2: The three reconstruction objectives differ before a scalar loss is evaluated: they condition on different data, introduce different latent objects, and support different claims.

	LimiX-style CCMM	Dense typed row–feature loss	Statistically pooled Unit belief
Data object	Context rows plus masked query rows from one task	One complete or nearly complete fixed-schema table	Evidence/query split for each row, optionally with repeated observations
Learned object	A family of context-conditioned masked conditionals	Type-routed cell-response predictors	Contextual messages, a parameter-free statistical readout, and its induced belief
Explicit latent	None in the optimized network objective	None; the row token is deterministic	Continuous candidate Unit variable $U \in \mathbb{R}^p$
Reference loss	Abstract conditional NLL	MSE, BCE, or full-softmax CE	Gaussian or Cauchy predictive NLL
Main boundary	Per-head criterion and block factorization are not public	Missing cells are not negative interactions	Message dispersion is not automatically a calibrated posterior

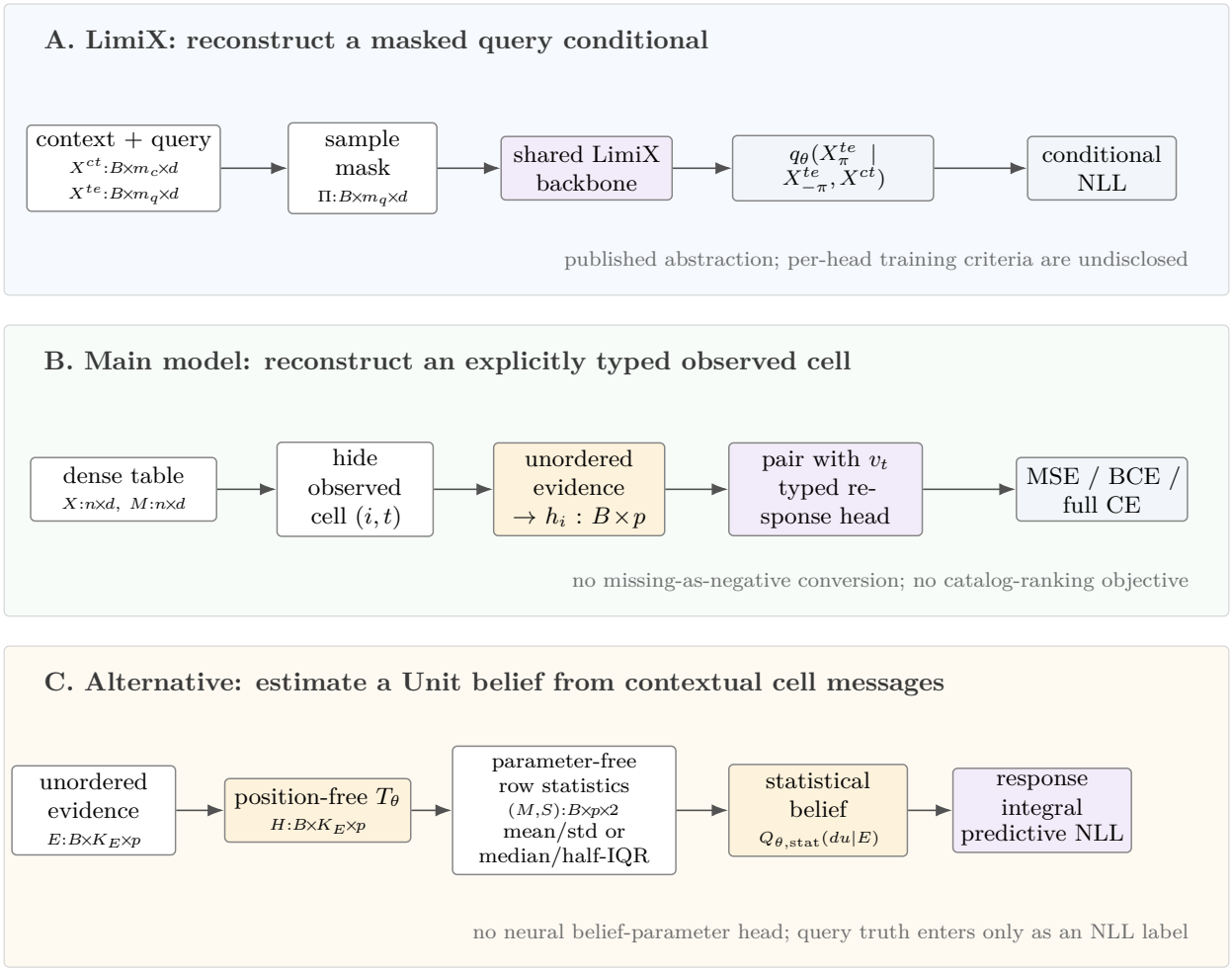

Figure 2: Three superficially similar reconstruction pipelines. B is an episode batch size, m_c, m_q are context and query row counts, d is schema width, $K_E \le d$ is padded evidence width, and p is hidden width. The top objective predicts a masked block directly, the middle objective scores an explicit typed cell, and the bottom objective turns a contextual cell table into a random variable by parameter-free, symmetric row statistics before decoding.

A.1 Objective I: LimiX context-conditional masked modeling

LimiX is trained across synthetic tasks. At the level of its theoretical analysis, an episode contains context rows

$$X^{ct} \in \Omega^{m_c \times d} \tag{27}$$

and a query row $X^{te} \in \Omega^d$. A feature mask $\pi \subseteq [d]$ separates hidden values X^{te}_{π} from visible query values $X^{te}_{-\pi}$. For fixed cardinality k, the paper defines

$$\Pi_k = \{\pi \subseteq [d] : |\pi| = k\}, \qquad \pi \sim \mathrm{Unif}(\Pi_k), \tag{28}$$

and optimizes

$$\widehat{L}_k(\theta) = \frac{1}{N} \sum_{a=1}^{N} - \log q_\theta\Big(x^{te,(a)}_{\pi_a} \mid x^{te,(a)}_{-\pi_a}, x^{ct,(a)}\Big). \tag{29}$$

The practical masking schedule described by the paper mixes cell-wise, column-wise, and block masks and uses mask ratios between 0.1 and 0.4 [6]. Those masks determine which conditional questions the shared model sees; they are not response-type likelihood definitions.

The optimized object in Eq. (29) is a conditional. LimiX relates the complete family of random-mask conditionals to $p(X^{te} \mid X^{ct})$ through an identifiability argument, but the network does not directly expose one tractable joint-density object. For $|\pi| > 1$, the notation treats X^{te}_{π} as a multivariate target. Neither the paper nor the released inference repository establishes whether the implemented training criterion is a normalized joint block likelihood or a sum of per-cell losses. The public heads establish output shapes—classification logits, one regression scalar, and grouped feature predictions—but do not establish CE, MSE, reconstruction weights, or a three-loss weighted sum.

During pretraining, the hidden truth is known to the trainer and updates the shared parameters. During downstream inference, the shared parameters are frozen; observed rows form context, the requested test value is hidden, and the same masked-query interface returns a prediction. Table-local preprocessing and ensembling are not gradient adaptation of the backbone. Thus the transferable lesson for our model is the construction of diverse conditional questions, not an undocumented per-type loss implementation.

A.2 Objective II: dense row–feature explicit responses

Let X^{raw} be an $n \times d$ heterogeneous table and $M \in \{0,1\}^{n \times d}$ its observation mask. The complete-table case is $M = \mathbf{1}_{n \times d}$. For a recorded target cell (i,t), define

$$\mathcal{X}^E_{i,t} = \{(j, x^{\mathrm{raw}}_{ij}) : M_{ij} = 1,\ j \neq t\}. \tag{30}$$

The target is removed before tokenization. The set encoder produces $h_i(\mathcal{X}^E_{i,t})$, which is paired with v_t and routed by target type. The three basic losses are exactly those in Eq. (22): standardized MSE for numeric responses, BCE for binary responses, and exact feature-local softmax CE for nominal categorical responses.

For a complete table, the empirical risk reduces to

$$\mathcal{L}_{\mathrm{dense}} = \frac{1}{nd} \sum_{t=1}^{d} \sum_{i=1}^{n} \ell_{it}. \tag{31}$$

Sampling a feature and then a row is merely a stochastic estimator of Eq. (31); it is not negative sampling. A recorded binary zero is a valid negative response, whereas $M_{it} = 0$ has no response label.

Features do not compete as a catalog, and cross-feature score ordering has no declared meaning. Consequently, implicit-feedback confidence weights and BPR are excluded from the basic objective [4].

Training hides recorded cells, updates all per-dataset parameters, and uses the hidden values only in the terminal losses. Inference freezes the parameters and train-fitted adapters, computes one dynamic row token from the currently visible cells, and answers all requested features that share that evidence set. There is no row-ID lookup or test-time gradient update.

A.3 Objective III: a Unit belief estimated by row statistics

This alternative does *not* attach a neural head that directly regresses the parameters of a Unit belief. The learnable network first transforms every visible cell into a contextual message; a fixed statistical operator then turns the messages in one row into a location–scale distribution. Thus the two stages are

$$R_i \in \mathbb{R}^{k_i \times p} \xrightarrow{T_\theta} H_i \in \mathbb{R}^{k_i \times p} \xrightarrow{\mathsf{RowStat}} Q_{\theta,\mathrm{stat}}(du \mid \mathcal{E}_i), \tag{32}$$

where R_i contains the evidence tokens in an arbitrary order and $H_i^{(\ell,a)}$ is coordinate a of contextual cell message ℓ. All statistics below use only the k_i valid messages; padding is ignored. There is no separate ψ or MLP after a pooled summary: the statistics themselves are the distribution parameters. The Transformer is trained to express heterogeneous feature-conditioned evidence as commensurate messages; only after that transformation are coordinatewise row statistics taken.

Gaussian row-statistic belief. For $k_i \geq 1$, define the coordinatewise sample location and dispersion

$$\widehat{m}_{ia} = \frac{1}{k_i} \sum_{\ell=1}^{k_i} H_i^{(\ell,a)}, \tag{33}$$

$$\widehat{s}_{ia}^{\,2} = \begin{cases} \dfrac{1}{k_i - 1} \displaystyle\sum_{\ell=1}^{k_i} \big(H_i^{(\ell,a)} - \widehat{m}_{ia}\big)^2, & k_i \geq 2, \\ 0, & k_i = 1, \end{cases} \qquad \widetilde{s}_{ia} = \sqrt{\widehat{s}_{ia}^{\,2} + \varepsilon^2}, \tag{34}$$

with one fixed numerical floor $\varepsilon > 0$. The resulting belief is

$$Q^{\mathrm{G}}_{\theta,\mathrm{stat}}(du \mid \mathcal{E}_i) = \bigotimes_{a=1}^{p} \mathcal{N}(du_a; \widehat{m}_{ia}, \widetilde{s}_{ia}^{\,2}). \tag{35}$$

The standard deviation here is the empirical dispersion of contextual cell messages. The current construction does not divide it by $\sqrt{k_i}$ or include an explicit count-calibration head.

Cauchy row-statistic belief. Let $\widehat{Q}_{ia}(q)$ denote the empirical q-quantile of $\{H_i^{(\ell,a)}\}_{\ell=1}^{k_i}$. Quantile matching gives

$$\widehat{m}^{\mathrm{C}}_{ia} = \widehat{Q}_{ia}(0.5), \tag{36}$$

$$\widehat{\gamma}_{ia} = \frac{\widehat{Q}_{ia}(0.75) - \widehat{Q}_{ia}(0.25)}{2} + \varepsilon, \tag{37}$$

and hence

$$Q^{\mathrm{C}}_{\theta,\mathrm{stat}}(du \mid \mathcal{E}_i) = \bigotimes_{a=1}^{p} \mathrm{Cauchy}(du_a; \widehat{m}^{\mathrm{C}}_{ia}, \widehat{\gamma}_{ia}). \tag{38}$$

The empirical-quantile interpolation rule must be fixed once and used for every row. This is a floor-regularized, coordinatewise quantile-matching estimator, not a claim that the messages are i.i.d. Cauchy samples or that median–half-IQR is the Cauchy maximum-likelihood estimator. Its robustness is a property of the terminal row statistic, not a claim of end-to-end robustness because one input cell can affect every contextual message. Both branches are exactly invariant to a permutation of evidence cells: the position-free Transformer is permutation equivariant and every statistic in Eqs. (33), (34), (36) and (37) is symmetric.

Feature-conditioned response and direct predictive loss. The same statistically constructed belief answers each target-feature query. For a candidate Unit value $u \in \mathbb{R}^p$, let

$$\mu_t(u) = a_\theta(v_t) + \langle g_\theta(v_t), u \rangle, \qquad g_\theta(v_t) \in \mathbb{R}^p, \tag{39}$$

where the shared functions a_θ and g_θ read the learned feature token. The generic predictive distribution is

$$q_{\theta,t}(x \mid \mathcal{E}_i) = \int p_{\theta,t}(x \mid u, v_t) Q_{\theta,\mathrm{stat}}(du \mid \mathcal{E}_i). \tag{40}$$

Writing the response kernel without $\mathcal{E}_i$ is a modeling assumption that the candidate Unit and target feature are response-sufficient for the evidence.

For the Gaussian branch, take $X_{it} = \mu_t(U) + \epsilon_{it}$ with $\epsilon_{it} \sim \mathcal{N}(0, \sigma_t^2)$, independent of U. Marginalization is analytic:

$$\mu_{it}^{\mathrm{G}} = a_\theta(v_t) + \sum_{a=1}^{p} g_{\theta,a}(v_t) \widehat{m}_{ia}, \tag{41}$$

$$(\tau_{it}^{\mathrm{G}})^2 = \sigma_t^2 + \sum_{a=1}^{p} g_{\theta,a}(v_t)^2 \widetilde{s}_{ia}^2, \tag{42}$$

$$\ell_{it}^{\mathrm{G}} = \frac{1}{2} \log\!\big(2\pi (\tau_{it}^{\mathrm{G}})^2\big) + \frac{(x_{it} - \mu_{it}^{\mathrm{G}})^2}{2(\tau_{it}^{\mathrm{G}})^2}. \tag{43}$$

For the Cauchy branch, take $\epsilon_{it} \sim \mathrm{Cauchy}(0, \lambda_t)$, independent of U. Cauchy stability gives

$$m_{it}^{\mathrm{C}} = a_\theta(v_t) + \sum_{a=1}^{p} g_{\theta,a}(v_t) \widehat{m}_{ia}^{\mathrm{C}}, \tag{44}$$

$$\omega_{it}^{\mathrm{C}} = \lambda_t + \sum_{a=1}^{p} |g_{\theta,a}(v_t)| \widehat{\gamma}_{ia}, \tag{45}$$

$$\ell_{it}^{\mathrm{C}} = \log(\pi \omega_{it}^{\mathrm{C}}) + \log\left[1 + \frac{(x_{it} - m_{it}^{\mathrm{C}})^2}{(\omega_{it}^{\mathrm{C}})^2}\right]. \tag{46}$$

The positive event-noise scales σ_t and λ_t remain response parameters; they are distinct from the row-wise message dispersion used to construct the Unit belief. They are structurally distinct, but a marginal NLL that observes only the composed scale does not by itself separately identify event noise and belief dispersion. The Gaussian and Cauchy options also change the row statistic, belief family, scale composition, and likelihood together; they are complete alternative branches, not a likelihood-only ablation.

Training minimizes a marginal predictive NLL over masked observed cells,

$$\mathcal{L}_{\mathrm{stat}} = \frac{1}{\sum_i |J_i^Q|} \sum_i \sum_{t \in J_i^Q} [-\log q_{\theta,t}(x_{it} \mid \mathcal{E}_i)], \tag{47}$$

using Eq. (43) or Eq. (46) for numeric targets. There is no teacher posterior, variational KL, learned mixing-weight head, or test-time optimization. During training, gradients pass through the fixed statistics into T_θ, the shared cell encoder, and the feature tokens; the statistics themselves have no trainable parameters. Empirical quantiles are piecewise differentiable and may be implemented with their standard order-statistic subgradients. For binary or categorical targets, $p_{\theta,t}$ may instead be a normalized Bernoulli or feature-local categorical response; its integral in Eq. (40) generally requires quadrature or Monte Carlo and is not part of the analytic endpoint above. Summing marginal target NLLs as in Eq. (47) is a composite objective. A joint multi-target likelihood would additionally require a declared conditional response factorization or an explicit joint response kernel.

At inference, the same operations are retained but all parameters are frozen:

$$\mathcal{E}_i \to H_i \to (\widehat{m}_i, \widehat{s}_i) \text{ or } (\widehat{m}_i^{\mathrm{C}}, \widehat{\gamma}_i) \to Q_{\theta,\mathrm{stat}}(du \mid \mathcal{E}_i) \to q_{\theta,t}(x \mid \mathcal{E}_i). \tag{48}$$

Query truth and gradient updates are absent. An empty evidence row has no sample statistics and therefore requires a separately declared fixed fallback; that edge case is outside the present alternative.

Semantic and identification boundary. The fixed statistics make the computational construction explicit, but do not by themselves turn it into a calibrated Bayesian posterior. Predictive training can make contextual messages useful while leaving their coordinates non-identifiable. We call the output a belief by model declaration; the row statistics are not a classical posterior derivation from i.i.d. measurements. To use the word Unit literally, the task must declare a Unit space and an attribution protocol under which the row evidence refers to one persistent Individual; otherwise $Q_{\theta,\mathrm{stat}}$ is conservatively a distribution over a latent row state. If repeated records are attributed to the same Individual, they must contribute to one shared belief rather than be assigned unrelated row variables. These are scientific assumptions, not consequences of predictive NLL minimization.

A.4 Selection rule for this manuscript

The dense typed objective is the main-text reference because it is the smallest loss compatible with the declared per-dataset architecture and complete-table regime. The LimiX objective remains an upstream masking comparison. The statistically pooled Unit-belief branch is a research alternative: it replaces the deterministic mean-and-count row token by an explicit distribution, but it deliberately excludes a direct neural belief head, a discrete gating mixture, and explicit learned count calibration. Empirical comparison among the three must report not only reconstruction accuracy but also calibration, sensitivity to evidence amount, permutation checks, and whether the random branch is actually used.

B Brainstorm: Feature Tokens from Language Descriptions

This appendix records one architectural idea that is intentionally outside the v0.2 core contract: instead of learning every feature token only from tabular supervision, initialize or partially generate feature tokens from feature descriptions with a language model.

B.1 Motivation

The current model learns a per-dataset feature table $V_{\mathcal{D}} = [v_1; \ldots; v_d] \in \mathbb{R}^{d \times p}$ from masked-cell prediction alone. This is simple and stable, but it discards metadata that practitioners usually have,

such as feature names, data dictionaries, units, and short business definitions. The brainstorm asks whether this text can provide a useful prior for feature identity before tabular optimization.

B.2 Minimal formulation

For feature j, let s_j denote a textual description (for example: "Age of customer in years" or "Device operating system category"). Let f_{LM} be a frozen text encoder and $P_\omega : \mathbb{R}^r \to \mathbb{R}^p$ be a learned projector to the tabular hidden width. A language-derived feature token is

$$\hat{v}_j = P_\omega(f_{\mathrm{LM}}(s_j)) \in \mathbb{R}^p. \tag{49}$$

The operational token can then be defined in one of three ways:

$$v_j = \hat{v}_j \qquad \text{(fully language-generated)}, \tag{50}$$

$$v_j = \hat{v}_j + \delta_j,\ \delta_j \in \mathbb{R}^p \qquad \text{(language prior + learnable residual)}, \tag{51}$$

$$v_j = \alpha_j \hat{v}_j + (1 - \alpha_j)\tilde{v}_j,\ \alpha_j \in [0, 1] \qquad \text{(gated fusion)}. \tag{52}$$

The rest of the network is unchanged: $e_{ij} = c_{ij} + v_j$, then position-free abduction and target-conditioned scoring.

B.3 What this may help

- **Faster identity bootstrapping.** Semantically related columns may start closer in token space, reducing early optimization burden.
- **Better naming robustness.** Synonymous feature names may map to nearby priors, improving stability when schemas are edited.
- **Human-auditable priors.** Feature-token initialization becomes partly inspectable through source descriptions s_j.

B.4 Failure modes and boundaries

- **Text can be wrong or vague.** A noisy description can inject a bad prior and hurt performance relative to purely learned v_j.
- **Name leakage risk.** If descriptions contain task labels or proxy label phrases, the model may exploit leakage channels.
- **Ontology mismatch.** Similar names do not imply similar response behavior; tabular evidence must still dominate final prediction.
- **Not a foundation-model claim.** Even with language priors, this manuscript remains a per-dataset learner with no unseen-schema zero-shot claim.

B.5 Protocol sketch (future evidence gate)

Any adoption should pass at least four checks:

1. **Ablation:** learned-only v_j vs language-initialized vs language+residual.
2. **Leakage audit:** redact target-correlated words from s_j and measure sensitivity.

3. **Permutation and contract checks:** preserve exact invariance and no row-ID properties from the main text.

4. **Calibration and robustness:** compare uncertainty quality, not only point error.

Therefore this section is a design brainstorm, not an accepted component of the current v0.2 contract.

References

[1] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.

[2] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[3] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753, 2019.

[4] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[5] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[6] Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, et al. LimiX: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025. URL https://arxiv.org/abs/2509.03505.