Multidimensional DKVMN
as a Dynamic MIRT Model with a GPCM Head

January 23, 2026

1 Motivation and Setting

Let i index learners, $t = 1, \dots, T_i$ time steps, and $j = 1, \dots, Q$ items. At time t, learner i is administered item $q_{i,t} \in \{1, \dots, Q\}$ and produces a response $r_{i,t}$ belonging to a discrete set $\mathcal{Y}_{q_{i,t}}$ (binary, ordered categories, nominal options, or multi-response codes).

KT view. In deep knowledge tracing, DKVMN models maintain a key memory M^k and a value memory $M^v_{i,t}$, and learn a mapping

$$(q_{i,1:t}, r_{i,1:t-1}) \mapsto p(r_{i,t} \mid q_{i,t}, \text{history}),$$

typically with a scalar logit $z_{i,t}$ and a logistic output. The hidden state is a high-dimensional representation with no explicit psychometric semantics.

MIRT view. In MIRT, a learner is represented by a latent trait vector $\boldsymbol{\theta}_i \in \mathbb{R}^D$ and each item j by item parameters ψ_j. The response distribution is given by an item-specific model

$$p(r_{i,j} \mid \boldsymbol{\theta}_i, \psi_j),$$

often in log-linear form. This yields an interpretable and rotation-invariant latent space, but does not prescribe dynamics.

Objective. We want a model that is simultaneously:

- *dynamic*: $\boldsymbol{\theta}_{i,t}$ evolves with the response sequence;
- *memory-based*: dynamics are implemented by a DKVMN-type architecture with attention over learned memory slots;
- *MIRT-consistent*: at each t, responses are generated by a polytomous IRT head (GPCM) acting on $\boldsymbol{\theta}_{i,t}$ with item parameters;
- *extensible*: the structure supports future ordinal heads (e.g., GRM) but the current implementation is GPCM-only.

The core design choice is to treat the DKVMN value memory as a *slot-level MIRT state*, and attention as the mechanism that couples items, slots, and time.

2 Static MIRT in Log-Linear Form

We summarize only the structure required for the later construction. Let $\boldsymbol{\theta} \in \mathbb{R}^D$ denote a latent trait vector. For item j with response set $\mathcal{Y}_j$, we model

$$p_j(y \mid \boldsymbol{\theta}) = \mathbb{P}(X_j = y \mid \boldsymbol{\theta}), \quad y \in \mathcal{Y}_j. \tag{1}$$

A wide family of MIRT models fits

$$\log p_j(y \mid \boldsymbol{\theta}) = \mu_j(y) + \boldsymbol{a}_j(y)^\top \boldsymbol{\theta} - A_j(\boldsymbol{\theta}), \tag{2}$$

with category intercepts $\mu_j(y)$, loadings $\boldsymbol{a}_j(y) \in \mathbb{R}^D$, and log-partition A_j ensuring normalization. Many familiar models are special cases:

- **2PL / 3PL**: $\mathcal{Y}_j = \{0, 1\}$, $\boldsymbol{a}_j(1) = \boldsymbol{a}_j$, $\boldsymbol{a}_j(0) = \boldsymbol{0}$, $\mu_j(1) = -b_j$, $\mu_j(0) = 0$, optionally with a lower asymptote.
- **GPCM / GRM**: step or cumulative thresholds; still log-linear after reparameterization.
- **Nominal**: distinct $\boldsymbol{a}_{jc}$ and μ_{jc} per category c.

Invariances. If $\boldsymbol{\theta}' = B\boldsymbol{\theta} + \boldsymbol{c}$ with invertible B and vector $\boldsymbol{c}$, then

$$\boldsymbol{a}'_j(y) = B^{-\top}\boldsymbol{a}_j(y), \qquad \mu'_j(y) = \mu_j(y) + \boldsymbol{a}_j(y)^\top B^{-1}\boldsymbol{c} \tag{3}$$

yields exactly the same probabilities. Static models fix this indeterminacy by constraints such as

$$\mathbb{E}[\boldsymbol{\theta}] = \boldsymbol{0}, \qquad \mathrm{Cov}(\boldsymbol{\theta}) = I_D, \tag{4}$$

and, if desired, structural restrictions on $\boldsymbol{a}_j(y)$.

In the dynamic setting, (3) persists at each t, and we must keep it in mind when designing and interpreting the architecture.

3 Dynamic MIRT Viewpoint

A dynamic MIRT model assumes a latent trait trajectory

$$\boldsymbol{\theta}_{i,1:T_i} = (\boldsymbol{\theta}_{i,1}, \ldots, \boldsymbol{\theta}_{i,T_i}),$$

with some prior $p(\boldsymbol{\theta}_{i,1:T_i})$ and observation model

$$\mathbb{P}(r_{i,t} = y \mid \boldsymbol{\theta}_{i,t}, q_{i,t} = j) = p_j(y \mid \boldsymbol{\theta}_{i,t}), \quad y \in \mathcal{Y}_j. \tag{5}$$

Suppressing i for clarity, the trajectory likelihood is

$$p(\{(q_t, r_t)\}_{t=1}^T \mid \boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p_{q_t}(r_t \mid \boldsymbol{\theta}_t). \tag{6}$$

Rather than specifying $p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1})$ parametrically, we use a memory network to learn a deterministic map

$$\boldsymbol{\theta}_t = \Phi_\eta\big(\{(q_s, r_s)\}_{s \le t}\big), \tag{7}$$

where η collects the memory parameters. The current implementation uses a DKVMN with single-head slot attention as a realization of Φ_η.

4 Multidimensional DKVMN with Slot Attention

We now describe a multidimensional DKVMN whose value memory has D-dimensional rows and whose attention is implemented with a single-head item-to-slot read.

Throughout this section we suppress the learner index i.

4.1 Slot-level trait memory

We assume N learned memory slots (no explicit KC labels). Their static embedding matrix is

$$M^k \in \mathbb{R}^{N \times d_k},$$

which plays the role of a key memory and is shared across learners and time. The dynamic value memory at time t is

$$M_t^v = \begin{bmatrix} \boldsymbol{s}_{1,t}^\top \\ \vdots \\ \boldsymbol{s}_{N,t}^\top \end{bmatrix} \in \mathbb{R}^{N \times D}, \tag{8}$$

where $\boldsymbol{s}_{n,t} \in \mathbb{R}^D$ is the D-dimensional trait profile for slot n at time t.

Relationship to classical DKVMN and to MIRT dimensionality. When $D = 1$ and attention is single-head, (8) reduces to the standard DKVMN value memory with scalar slot mastery, and the update rule (13) becomes the familiar erase/add mechanism. Here D is the MIRT trait dimension, while N indexes a set of *memory slots* used for content addressing. Two interpretations are useful for psychometric work:

- **Slots aligned with dimensions.** If we enforce $D = N$ (or tie D to N) and set the value dimension to D, the readout can be interpreted as a direct mapping from slot activations to latent traits.
- **Slots as a content-addressable basis.** For general D and N, the rows $\boldsymbol{s}_{n,t} \in \mathbb{R}^D$ form a learned basis of slot-conditioned trait vectors, and w_t selects a convex combination relevant for the current item.

We initialize M_1^v either at zero or from a small Gaussian, and treat its evolution as deterministic given the interaction sequence.

4.2 Item embeddings

Each item j has a one-hot code that is mapped to an embedding

$$\boldsymbol{e}_j = E_q \boldsymbol{e}_j^{\text{onehot}} \in \mathbb{R}^{d_q}, \tag{9}$$

where $E_q \in \mathbb{R}^{d_q \times Q}$. At time t, we write $\boldsymbol{e}_{q_t}$ simply as $\boldsymbol{e}_t$.

For some constructions below, we also use an item embedding $\boldsymbol{k}_j$ in the same space as M^k,

$$\boldsymbol{k}_j = A^\top \boldsymbol{e}_j^{\text{onehot}} \in \mathbb{R}^{d_k}, \tag{10}$$

consistent with DKVMN.

4.3 Item-to-slot read

The current implementation uses a single-head dot-product attention over slots. Let $\boldsymbol{q}_t$ be the item embedding and M^k the key memory. We compute

$$w_t = \operatorname{softmax}\Big(\boldsymbol{q}_t^\top M^k\Big), \qquad r_t = \sum_{n=1}^{N} w_t(n)\, \boldsymbol{s}_{n,t}. \tag{11}$$

We then form a summary representation by concatenating the read vector with the item embedding,

$$\boldsymbol{h}_t = \tanh(W_h[r_t; \boldsymbol{q}_t] + \boldsymbol{b}_h)\,, \tag{12}$$

and extract $(\boldsymbol{\theta}_t, \boldsymbol{\alpha}_{q_t}, \boldsymbol{\beta}_{q_t})$ from $\boldsymbol{h}_t$ with a learned parameter head. Optionally, $\boldsymbol{\theta}_t$ can be projected directly from the read vector when D and the value dimension are aligned.

4.4 Write and memory evolution

The write step uses the same attention weights w_t and a DKVMN-style erase/add update. Let $\boldsymbol{v}_t$ be the embedded interaction (item, response), projected to the value dimension. Then

$$M^v_{t+1}(n) = M^v_t(n) \odot \big(1 - w_t(n)\, \boldsymbol{e}_t\big) + w_t(n)\, \boldsymbol{a}_t, \tag{13}$$

where $\boldsymbol{e}_t = \sigma(W_e \boldsymbol{v}_t)$ and $\boldsymbol{a}_t = \tanh(W_a \boldsymbol{v}_t)$ are the erase/add signals. This preserves the classic DKVMN dynamics while keeping a multidimensional trait representation.

4.5 Intuition

The mapping Φ_η in (7) is implemented by the slot attention, summary readout, and DKVMN write update. The slots store trait-relevant state, the attention weights select item-relevant slots, and the GPCM head maps $\boldsymbol{\theta}_t$ to polytomous response probabilities.

5 IRT Heads and Item Parameter Output

Implementation status. The current codebase implements the GPCM head only; other heads below are retained as future extensions.

Given $\boldsymbol{\theta}_t$ and $q_t = j$, responses are generated via an item-specific head

$$p(r_t \mid \boldsymbol{\theta}_t, q_t = j) = p_{\tau_j}(r_t \mid \boldsymbol{\theta}_t; \psi_j),$$

where τ_j encodes the item type and ψ_j the item parameters. All heads share the latent space $\mathbb{R}^D$.

5.1 General log-linear head

We first define a unified log-linear head for item j:

$$\eta_{j,y}(\boldsymbol{\theta}_t) = \mu_j(y) + \boldsymbol{a}_j(y)^\top \boldsymbol{\theta}_t, \quad y \in \mathcal{Y}_j, \tag{14}$$

and

$$p_j(y \mid \boldsymbol{\theta}_t) = \frac{\exp\big(\eta_{j,y}(\boldsymbol{\theta}_t)\big)}{\sum_{c \in \mathcal{Y}_j} \exp\big(\eta_{j,c}(\boldsymbol{\theta}_t)\big)}. \tag{15}$$

Different IRT families are recovered by structural constraints on $\mu_j(\cdot)$ and $\boldsymbol{a}_j(\cdot)$.

5.2 Parameter output in the current code

The current implementation conditions item parameters on the time-step summary and the item embedding. Concretely:

- $\boldsymbol{\theta}_t$ is produced from the summary state $\boldsymbol{h}_t$ (or optionally from the read vector when value and trait dimensions are aligned).
- $\boldsymbol{\alpha}_{q_t}$ is produced from $[\boldsymbol{h}_t; \boldsymbol{q}_t]$ via a shallow network with a positive transform.
- $\boldsymbol{\beta}_{q_t}$ is produced from the item embedding $\boldsymbol{q}_t$ and constructed as an ordered threshold vector.

This matches the implemented extractor in the current codebase and keeps item parameters conditioned on the active item and the time-step state.

Binary 2PL / 3PL. For $\mathcal{Y}_j = \{0, 1\}$ we set

$$\begin{aligned} \boldsymbol{a}_j &= W_a^{\text{irt}} \boldsymbol{h}_j, && (16)\\ b_j &= \boldsymbol{w}_b^\top \boldsymbol{h}_j + b_b, && (17)\\ \eta_{j,1}(\boldsymbol{\theta}_t) &= \boldsymbol{a}_j^\top \boldsymbol{\theta}_t - b_j, && (18)\\ \eta_{j,0}(\boldsymbol{\theta}_t) &= 0, && (19) \end{aligned}$$

so that

$$p_j(1 \mid \boldsymbol{\theta}_t) = \sigma(\boldsymbol{a}_j^\top \boldsymbol{\theta}_t - b_j). \tag{20}$$

A 3PL variant is obtained by adding a lower asymptote $c_j = \sigma(\boldsymbol{w}_c^\top \boldsymbol{h}_j)$ and defining

$$p_j(1 \mid \boldsymbol{\theta}_t) = c_j + (1 - c_j)\, \sigma(\boldsymbol{a}_j^\top \boldsymbol{\theta}_t - b_j).$$

Multidimensional GPCM. For ordered categories $k = 0, \dots, K_j - 1$, we use a common loading $\boldsymbol{a}_j$ and $K_j - 1$ cumulative step difficulties. Define

$$\begin{aligned} \boldsymbol{a}_j &= W_a^{\text{irt}} \boldsymbol{h}_j, && (21)\\ s_j &= \frac{\|\boldsymbol{a}_j\|_2}{\sqrt{D}}, && (22)\\ \delta_{j0} &= \boldsymbol{w}_0^\top \boldsymbol{h}_j + b_0, && (23)\\ \Delta_{jh} &= \operatorname{softplus}(\boldsymbol{w}_h^\top \boldsymbol{h}_j + b_h), \quad h = 1, \dots, K_j - 2, && (24)\\ \delta_{jh} &= \delta_{j0} + \sum_{\ell=1}^{h} \Delta_{j\ell}, && (25) \end{aligned}$$

and set the baseline category $k = 0$ to have zero logit. Then

$$p_j(r_t = k \mid \boldsymbol{\theta}_t) = \frac{\exp\Big(\sum_{h=0}^{k-1} \big(\boldsymbol{a}_j^\top \boldsymbol{\theta}_t - s_j\, \delta_{jh}\big)\Big)}{\sum_{c=0}^{K_j-1} \exp\Big(\sum_{h=0}^{c-1} \big(\boldsymbol{a}_j^\top \boldsymbol{\theta}_t - s_j\, \delta_{jh}\big)\Big)}. \tag{26}$$

Equation (26) is the vector extension of the standard K-1 step GPCM head: in the special case $D = 1$ and scalar $\boldsymbol{a}_j$, it collapses to the usual 1D formulation.

Graded Response Model (GRM). For GRM, we parameterize ordered thresholds γ_{jk} via

$$\gamma_{j0} = \boldsymbol{u}_0^\top \boldsymbol{h}_j + c_0, \tag{27}$$

$$\Delta_{jk}^{(\gamma)} = \mathrm{softplus}(\boldsymbol{u}_k^\top \boldsymbol{h}_j + c_k), \quad k = 1, \ldots, K_j - 1, \tag{28}$$

$$\gamma_{jk} = \gamma_{j0} + \sum_{\ell=1}^{k} \Delta_{j\ell}^{(\gamma)}. \tag{29}$$

We keep a common loading $\boldsymbol{a}_j = W_a^{\mathrm{irt}} \boldsymbol{h}_j$ as above. Define cumulative probabilities

$$P(r_t \geq k \mid \boldsymbol{\theta}_t) = \sigma(\boldsymbol{a}_j^\top \boldsymbol{\theta}_t - \gamma_{jk}), \tag{30}$$

and category probabilities

$$p_j(r_t = k \mid \boldsymbol{\theta}_t) = P(r_t \geq k \mid \boldsymbol{\theta}_t) - P(r_t \geq k+1 \mid \boldsymbol{\theta}_t), \tag{31}$$

with $P(r_t \geq 0) = 1$ and $P(r_t \geq K_j) = 0$.

Nominal Response Model (NRM). For unordered categories $c = 1, \ldots, C_j$ we assign category-specific loadings and intercepts:

$$\boldsymbol{\alpha}_{jc} = W_{\alpha,c} \boldsymbol{h}_j, \tag{32}$$

$$\beta_{jc} = \boldsymbol{w}_{\beta,c}^\top \boldsymbol{h}_j + b_{\beta,c}, \tag{33}$$

and define

$$p_j(r_t = c \mid \boldsymbol{\theta}_t) = \frac{\exp(\boldsymbol{\alpha}_{jc}^\top \boldsymbol{\theta}_t + \beta_{jc})}{\sum_{d=1}^{C_j} \exp(\boldsymbol{\alpha}_{jd}^\top \boldsymbol{\theta}_t + \beta_{jd})}. \tag{34}$$

The NRM is thus obtained as a particular instantiation of (15).

Multi-response items. For items allowing multiple selections or sub-scores, two strategies are possible:

- treat each sub-part as a separate item sharing $\boldsymbol{\theta}_t$ and its own $(\boldsymbol{a}_{j'}, \mu_{j'})$;
- model the joint outcome via a higher-dimensional categorical with its own $(\mu_j(y), \boldsymbol{a}_j(y))$.

In both cases the parameter networks are shared, and the latent space is unchanged.

6 Training Objective and Constraints

6.1 Sequence likelihood

Given a sequence $\{(q_t, r_t)\}_{t=1}^{T}$, the model computes

$$M_1^v \xrightarrow{\text{updates with } (q_1,r_1),\ldots,(q_{t-1},r_{t-1})} M_t^v \xrightarrow{\text{slot attention + summary read}} \boldsymbol{\theta}_t \xrightarrow{\text{IRT head for } q_t} p(r_t \mid \boldsymbol{\theta}_t, q_t). \tag{35}$$

The approximate log-likelihood for one learner is

$$\mathcal{L} = \sum_{t=1}^{T} \log p_{q_t}(r_t \mid \boldsymbol{\theta}_t), \tag{36}$$

and for a dataset we sum over learners and sequences. The model is trained by maximising $\mathcal{L}$ (or minimizing the negative log-likelihood) via stochastic gradient descent.

6.2 Optimization and engineering details

In practice, KT data exhibit long-tailed item frequencies, missingness, and long sequences. The current implementation uses straightforward sequence-model training choices:

- **Causal computation.** $\boldsymbol{\theta}_t$ is computed from $(q_{1:t}, r_{1:t-1})$ only; the likelihood term at time t uses the pre-update state to avoid label leakage.
- **Optimizer.** AdamW with fixed learning rate and weight decay; no warmup schedule is currently applied.
- **Regularization.** Optional penalties include attention entropy, θ norm, and priors on α and β (configurable weights).
- **Batching.** Variable-length sequences are padded with masks. Sequences are processed in full length with masking.

These choices do not change the statistical model but materially improve stability and reproducibility.

6.3 Regularization and approximate identifiability

The static MIRT invariances in (3) extend to the dynamic case: for any invertible B and $\boldsymbol{c}$, the transformed traits $\boldsymbol{\theta}'_t = B\boldsymbol{\theta}_t + \boldsymbol{c}$ and transformed item parameters produce identical likelihood. With a flexible DKVMN mapping, Φ_η and the item networks can jointly rotate without changing predictions.

To obtain a stable and interpretable latent space we propose:

- **Standardization penalty.** Add penalties encouraging

$$\frac{1}{N_{\text{tot}}} \sum_{i,t} \boldsymbol{\theta}_{i,t} \approx \mathbf{0}, \qquad \frac{1}{N_{\text{tot}}} \sum_{i,t} \boldsymbol{\theta}_{i,t} \boldsymbol{\theta}_{i,t}^\top \approx I_D.$$

- **Post-hoc alignment.** After training, apply a Procrustes-type transformation to align $\boldsymbol{\theta}_{i,t}$ and loadings to a standard MIRT orientation for interpretability.
- **Structural constraints.** Use shared $\boldsymbol{a}_j$ across categories in graded models, ordered thresholds via softplus, and limited depth in item networks to keep parameterizations close to classical forms.

These steps do not alter predictive performance but make the dynamic traits comparable across runs and close to the usual MIRT conventions.

7 Research Questions and Plan

The proposed architecture raises technical questions at the intersection of deep sequence modeling and measurement theory. We propose to study these questions using controlled synthetic data (where ground truth is known) and mixed-format real datasets.

7.1 Theoretical and synthetic analysis

We will simulate from a generative *dynamic* MIRT model with known item parameters and latent evolution, including binary, graded, and nominal outcomes. This enables parameter recovery and identifiability analyses that are not possible with observational KT datasets alone.

- **RQ1: Approximation of dynamic MIRT filtering.** To what extent can the DKVMN-MIRT model approximate the Bayes filter (or MAP filter) of a known dynamic MIRT model? How does recovery depend on (N, D), memory size, and the choice of θ source (summary vs. memory projection)?
- **RQ2: Rotation, scaling, and stability.** How do standardization penalties, discrimination constraints, and anchoring strategies affect latent orientation and training stability? What residual indeterminacy remains after penalties and post-hoc alignment?
- **RQ3: Mixed-format measurement.** When binary, partial credit, and nominal items co-exist, does using a shared latent space with type-specific heads improve calibration and parameter recovery compared with training separate models per format?
- **RQ4: Attention interpretability.** Do attention weights align with slot contributions to the MIRT score (e.g., θ–attention correlation diagnostics), and how stable is this alignment across D?
- **RQ5: Regularization and stability.** How do attention entropy and trait normalization penalties affect convergence, calibration, and parameter recovery?

We will report (i) recovery of item parameters up to allowable affine transformations, (ii) RMSE/correlation of aligned latent trajectories, and (iii) predictive metrics (AUC/accuracy where applicable, log loss, and calibration).

7.2 Real-data experiments

On real KT and psychometric datasets with mixed formats (binary items, partial credit rubrics, Likert-style items, nominal options), we will:

- compare DKVMN-MIRT against representative baselines:
  - DKVMN with binary head and its mixed-format variants,
  - DKT-style RNN baselines,
  - attention-based KT models (e.g. AKT/Transformer-based KT),
  - unidimensional neural IRT / Deep-IRT,
  - static MIRT baselines where applicable;
- evaluate predictive performance, calibration, and robustness to varying sequence lengths and item sparsity;
- analyze learned dimensions and item parameters, including alignment with external scales or known subtests, and stability of attention patterns across item groups.

8 Contributions

This work aims to make deep knowledge tracing architectures usable as *mathematical psychometric models* rather than purely predictive black boxes:

(a) A multidimensional DKVMN that treats the value memory as a slot-indexed trait state and whose read operation yields a dynamic MIRT latent vector.

(b) A polytomous measurement layer with a concrete GPCM parameterization; other heads remain future extensions.

(c) A principled account of invariances and approximate identifiability in amortized dynamic MIRT, including standardization penalties and post-hoc alignment procedures.

(d) Empirical evidence (synthetic recovery and real-data performance) characterizing tradeoffs between predictive accuracy, calibration, and interpretability in a slot-attention MIRT model.