MIRT: Multi-Dimensional IRT for SLO-Adaptive Multi-Agent Routing

Anonymous Authors[1]

Abstract

Multi-agent LLM routing policies are typically learned offline from cached query–action outcomes, yet must adapt online to shifting cost and latency constraints without retraining. We formalize this offline-to-online adaptation problem and approach it through Item Response Theory (IRT), decomposing each query–action outcome into latent ability and difficulty factors that are learned entirely offline and independently of constraint thresholds. We identify two structural pitfalls: one-dimensional IRT provably reduces to static action selection regardless of query difficulty, and end-to-end training collapses routing diversity. Our method, Multi-dimensional IRT (**MIRT**), resolves both via D-dimensional latent factors and two-stage decomposition. Because the learned factors are constraint-independent, online adaptation reduces to recalibrating only two Lagrangian dual variables, enabling a single offline-trained model to serve shifting SLO regimes. MIRT outperforms the best parametric baseline by +3.9 pp F1 (0.797 vs. 0.758), maintains stable performance across regimes (F1 0.796–0.800), and is the only method keeping both cost and latency violations below the 5% target across all three regimes.

1. Introduction

Multi-agent LLM systems route each query to one of K configurations (topologies $\times$ model tiers) (Wu et al., 2023; Hong et al., 2023). Because exhaustive online evaluation is impractical, the routing policy is learned *offline* from cached query–action outcomes. At deployment, however, Service Level Objectives (SLOs) on cost and latency shift frequently—latency may tighten from 90 s to 60 s during peak hours or relax during off-peak (Chen et al., 2023)—and the policy must adapt *online* to these shifting constraints without retraining. This *offline-to-online adaptation* challenge is the central problem we address: how to learn query–action structure once from offline data and then adapt to new constraint regimes with minimal online recalibration.

We approach this through Item Response Theory (IRT) (Hambleton et al., 1991), a latent-factor framework from psychometrics that decomposes test-taker $\times$ test-item interactions into ability and difficulty parameters. LLM routing has an analogous structure: each query–action pair produces an outcome (quality, cost, latency) that depends jointly on the query's characteristics and the action's capabilities. IRT's bilinear decomposition captures this interaction in a low-dimensional latent space, and crucially, the learned factors are independent of external constraint thresholds—separating *what is learned offline* (query–action structure) from *what is adapted online* (SLO targets), which directly enables offline-to-online transfer.

Applying IRT to routing, however, encounters two structural pitfalls that go beyond implementation choices: each is a provable or empirical limitation that cannot be fixed by tuning hyperparameters.

Pitfall 1: 1D ability models collapse to static routing. In its standard one-dimensional form (Birnbaum, 1968), IRT models the expected outcome for query i and action j as $\hat{Y}_{ij} = \sigma(a_i\theta_j - b_i)$, where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function, θ_j is action j's ability, and a_i, b_i are query-level parameters. However, this one-dimensional formulation *always selects the same action*: because σ is monotone, the ranking $\theta_1 > \theta_2 > \cdots$ is preserved for every query (Reckase, 2009; Hambleton et al., 1991), reducing the router to static routing regardless of query difficulty.

Pitfall 2: end-to-end training causes representation collapse. One might instead train the query encoder and IRT parameters jointly in an end-to-end (E2E) fashion. Our ablation shows this causes dimensional collapse (Jing et al., 2022; Guo et al., 2024): routing diversity drops from 11 unique actions to 3, near-static behavior worse than the two-stage decomposition (Table 6). Pairwise cosine similarity of

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the learned $\mathbf{a}_i$ vectors confirms the mechanism: E2E drives mean cosine to 1.00 (100% of pairs $>$0.95), effectively degenerating MIRT back to 1D-IRT.

MIRT addresses both pitfalls: (1) multi-dimensional IRT (Reckase, 2009) replaces scalar ability with D-dimensional vectors, enabling per-query routing; and (2) a two-stage architecture (MIRT fitting followed by MLP mapping) prevents representation collapse by keeping the latent structure fixed during mapper training. Cost and latency SLOs are enforced via online Lagrangian dual-variable adaptation (Eq. 5–6), and because MIRT parameters are SLO-independent, only λ requires recalibration when the SLO changes. The result is *offline-to-online adaptive* routing: the latent query–action structure is learned once from offline data, and online adaptation to new constraint regimes requires only recalibrating two scalar dual variables—no retraining, no access to the original training set.

Our contributions:

1. **Structural analysis of IRT-based routing.** We prove that 1D IRT reduces to a static policy (Proposition 1) and show empirically that end-to-end training collapses policy diversity.
2. **MIRT: offline-learned representations for online adaptation.** MIRT decomposes query–action interactions via multi-dimensional latent factors learned entirely from offline data. Because its parameters are constraint-independent, online adaptation to new SLO regimes requires only recalibrating two Lagrangian dual variables.
3. **Empirical validation of offline-to-online transfer**: MIRT outperforms all parametric baselines (AdaptDecLag +3.9 pp, BaRP-GBT +0.8 pp) and is the only method that keeps both CV and LV below ε=5% across all three SLO regimes, thanks to violation-rate-aware feasibility filtering and aggressive dual-variable calibration.

2. Related Work

Our work sits at the intersection of four research areas: policy learning from offline data, multi-agent routing, LLM routing, and Item Response Theory applied to machine learning.

Policy learning from offline data. Learning policies from previously collected data is studied in offline RL (Levine et al., 2020), off-policy evaluation (Dudík et al., 2014), and contextual bandits from logged data (Swaminathan & Joachims, 2015). Our setting is simpler: the offline data provides full-information feedback (all query–action outcomes observed), so importance weighting and pessimistic estimation are unnecessary. The challenge specific to our setting is cross-regime transfer: a policy learned under one set of cost/latency constraints must operate under different thresholds without retraining.

Multi-agent routing. Fixed multi-agent topologies (Wu et al., 2023; Hong et al., 2023; Du et al., 2023) cannot adapt to per-query requirements. Recent adaptive approaches use VGAEs (Zhang et al., 2024), probabilistic supernets (Zhang et al., 2025), or cascading LLM controllers (Yue et al., 2025; Su et al., 2025). These optimize for quality without enforcing hard cost or latency constraints.

LLM routing. Single-model routers select among LLMs based on quality (Ding et al., 2024; Ong et al., 2024) or Pareto trade-offs (Chen et al., 2023). EvoRoute (Zhang et al., 2026) uses experience-driven self-routing but does not enforce hard SLO constraints. PROTEUS (Bhatti et al., 2026) uses Lagrangian RL for SLA-aware routing across multiple LLMs, but does not model query–action interactions via latent decomposition. BaRP (Wang et al., 2025) applies multi-objective bandits with SLO parameters as model inputs, requiring retraining when SLOs change. None of these combine multi-agent topology selection with cross-SLO transfer.

Item Response Theory in ML. IRT, originally from psychometrics (Rasch, 1960; Birnbaum, 1968), has been applied to evaluate NLP models (Martínez-Plumed et al., 2019; Lalor et al., 2019; Sedoc & Ungar, 2020) and benchmark difficulty calibration (Benedetto et al., 2023). Recent work has extended IRT to routing: IRT-Router (Song et al., 2025) fits both 1D and multi-dimensional IRT models to predict per-LLM accuracy, routing queries between a pair of LLMs by selecting the cheaper model when predicted accuracy exceeds a threshold. RouterDC (Chen et al., 2024) trains per-LLM binary classifiers using dual contrastive learning, treating routing as a matchmaking problem. These approaches use IRT for single-model selection without cost/latency constraint enforcement. MIRT differs in two key ways: we operate over multi-agent *topologies* (not just single models), and enforce cost/latency SLOs via online Lagrangian λ adaptation, enabling cross-SLO transfer without retraining.

3. Method

3.1. Problem Setup

We select among K=33 multi-agent configurations. Five topologies are combined with three GPT-4.1 model tiers (OpenAI, 2025) (T1=nano, T2=mini, T3=large): SAS (single agent with self-refinement), DEC (decentralized multi-agent debate), IND (independent parallel agents), CEN (centralized orchestrator with workers), and HYB

(hybrid orchestration). SAS and DEC use a single tier for all roles (3 each), while IND, CEN, and HYB assign separate tiers to brain and worker roles (3×3=9 each), giving 3+9+9+3+9=33 (Appendix A). Each query i is represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ (d=384), the frozen sentence embedding from MiniLM-L6-v2 (Reimers & Gurevych, 2019; Wang et al., 2020), z-score normalized per dimension. For each of the N training queries and K actions, we observe quality (token-level F1), cost, and latency from an oracle cache. The goal is to select the action maximizing F1 subject to SLO constraints: cost $\leq \gamma$ and latency $\leq \tau$.

3.2. IRT Formulation for Routing

1D IRT (2PL model). In the two-parameter logistic model (Birnbaum, 1968), the expected outcome for query i and action j is:

$$\hat{Y}_{ij} = \sigma(a_i \cdot \theta_j - b_i) \tag{1}$$

where $\theta_j \in \mathbb{R}$ is action j's ability, $b_i \in \mathbb{R}$ is query i's difficulty, $a_i > 0$ is query i's discrimination, and σ is the sigmoid function.

Proposition 1 (1D IRT is static)**.** *For any 1D IRT model (Eq. 1), the action ranking is independent of the query: $\hat{Y}_{ij} > \hat{Y}_{ik} \iff \theta_j > \theta_k$ for all queries i.*

Proof. Since σ is strictly monotone increasing, $\hat{Y}_{ij} > \hat{Y}_{ik}$ iff $a_i\theta_j - b_i > a_i\theta_k - b_i$ iff $a_i(\theta_j - \theta_k) > 0$. Since $a_i > 0$ (discrimination is positive), this reduces to $\theta_j > \theta_k$, which is independent of i. □

This means 1D IRT *always selects the same action* regardless of query difficulty; it is structurally equivalent to static routing (Figure 1, left). Our experiments confirm this: 1D IRT selects a single action (IND-T2-T3) for all 495 test queries.

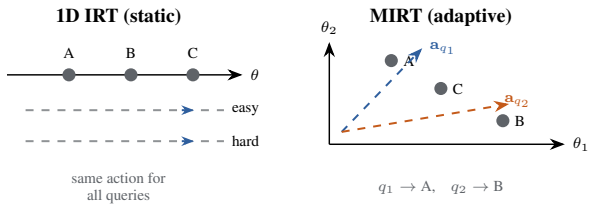

Figure 1. 1D IRT preserves action ranking for all queries (Prop. 1). MIRT enables query-adaptive routing: different emphasis vectors $\mathbf{a}_i$ project actions differently, changing the optimal selection.

Multi-dimensional IRT (MIRT). MIRT (Reckase, 2009) extends ability and discrimination to D-dimensional vectors:

$$\hat{Y}_{ij} = \sigma(\mathbf{a}_i^\top \boldsymbol{\theta}_j - b_i) \tag{2}$$

where $\boldsymbol{\theta}_j \in \mathbb{R}^D$ is action j's ability profile and $\mathbf{a}_i \in \mathbb{R}^D$ is query i's emphasis vector. The key difference: the inner product $\mathbf{a}_i^\top \boldsymbol{\theta}_j$ allows different queries to weight ability dimensions differently. An action may excel in dimension 1 (e.g., reasoning depth) but not dimension 2 (e.g., cost efficiency). A query emphasizing dimension 2 will rank this action lower, breaking the static ranking of 1D IRT.

3.3. MIRT Router

MIRT operates in two stages (Figure 2).

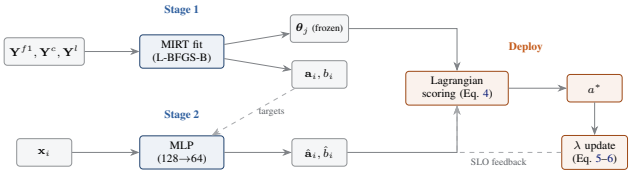

Figure 2. MIRT pipeline. Stage 1 fits MIRT on outcome matrices; Stage 2 trains an MLP to predict per-query IRT parameters. At deploy time, Lagrangian scoring (Eq. 4) selects actions and λ adapts online.

Stage 1: MIRT fitting. We fit three independent MIRT models on the $N \times K$ outcome matrices for F1, cost efficiency, and latency efficiency. The cost and latency targets are normalized continuous scores, $Y^c_{ij} = \text{clip}(1 - \text{cost}_{ij}/c_{\max}, 0, 1)$ and $Y^l_{ij} = \text{clip}(1 - \text{lat}_{ij}/l_{\max}, 0, 1)$, rather than binary labels. Parameters $\{\boldsymbol{\theta}_j, \mathbf{a}_i, b_i\}$ are estimated via L-BFGS-B (Byrd et al., 1995) minimization of:

$$\mathcal{L} = \frac{1}{2NK}\sum_{i,j}(\hat{Y}_{ij} - Y_{ij})^2 + \lambda_{\text{reg}}(\|\boldsymbol{\theta}\|^2 + \|\mathbf{a}\|^2 + \|b\|^2) \tag{3}$$

Since all targets are continuous scores in $[0, 1]$ rather than binary labels, we use MSE with the sigmoid link, a nonlinear least squares formulation consistent with fractional response regression (Papke & Wooldridge, 1996) and continuous IRT models (Samejima, 1973).

Stage 2: Feature-to-IRT mapping. The fitted $\mathbf{a}_i, b_i$ describe training queries but do not generalize to unseen queries. We train an MLP to predict IRT parameters from query features: $(\hat{\mathbf{a}}_i, \hat{b}_i) = \text{MLP}(\mathbf{x}_i)$. At test time, we predict $\hat{\mathbf{a}}, \hat{b}$ from the feature vector and compute expected outcomes via Eq. 2 using the fixed $\boldsymbol{\theta}_j$ from Stage 1.

Deploy: Lagrangian action selection. Given predicted outcomes for F1 ($\hat{Y}^{f1}$), cost efficiency ($\hat{Y}^c$), and latency efficiency ($\hat{Y}^l$), we select actions via Lagrangian scalarization:

$$a^* = \arg\max_j \; \hat{Y}^{f1}_{ij} - \lambda_c \cdot P^c_{ij} - \lambda_l \cdot P^l_{ij} \tag{4}$$

where $P_{ij}^c = \max(0, (c_{\max}/\gamma)(1 - \hat{Y}_{ij}^c) - 1)$, $P_{ij}^l = \max(0, (l_{\max}/\tau)(1 - \hat{Y}_{ij}^l) - 1)$, and $c_{\max}, l_{\max}$ are 95th-percentile normalizers. The Lagrangian dual variables $\lambda_c, \lambda_l \geq 0$ are updated online:

$$\lambda_c \leftarrow \text{clip}[\lambda_c + \eta(v_t^c - \varepsilon),\ 0,\ \lambda_{\max}] \tag{5}$$

$$\lambda_l \leftarrow \text{clip}[\lambda_l + \eta(v_t^l - \varepsilon),\ 0,\ \lambda_{\max}] \tag{6}$$

where $\text{clip}[x, a, b] = \min(\max(x, a), b)$, $v_t^c, v_t^l \in \{0, 1\}$ are observed violations, and ε is the target violation rate. As violations exceed ε, λ increases, shifting score weights toward penalty terms and steering selection toward SLO-compliant actions.

Feasibility pre-filtering. Before Lagrangian scoring, we apply a violation-rate-aware feasibility mask: for each action j, we compute its empirical constraint violation rate on the training set, $\hat{v}_j^c = \frac{1}{N}\sum_i \mathbb{1}[c_{ij} > \gamma]$ (and analogously for latency). Actions whose violation rate exceeds 2ε on either constraint are excluded from the candidate set (their scores are set to $-\infty$). This is stricter than a mean-cost filter, which can admit actions that appear cheap on average but frequently violate the constraint on individual queries. Under tight, moderate, and relaxed SLOs, this reduces the candidate set from 33 to 8, 8, and 12 actions, respectively.

Cross-SLO transfer. MIRT parameters $(\boldsymbol{\theta}, \mathbf{a}, b)$ are SLO-independent: they describe the inherent query–action interaction structure. When the SLO changes from $(\gamma, \tau, \varepsilon)$ to $(\gamma', \tau', \varepsilon')$, MIRT updates only the penalty thresholds and recalibrates λ on a small calibration set; the MIRT models and feature mappings remain unchanged. A single trained model therefore serves all SLO regimes. Algorithm 1 summarizes the full procedure.

Algorithm 1 MIRT Router

Train: Outcome matrices $\mathbf{Y}^{f1}, \mathbf{Y}^c, \mathbf{Y}^l \in \mathbb{R}^{N \times K}$; features $\mathbf{X} \in \mathbb{R}^{N \times d}$

Fit MIRT (Eq. 2) on each outcome matrix $\rightarrow \{\boldsymbol{\theta}_j^{(\cdot)}, \mathbf{a}_i^{(\cdot)}, b_i^{(\cdot)}\}$

Train MLP: $\mathbf{x}_i \rightarrow (\hat{\mathbf{a}}_i, \hat{b}_i)$ for F1, cost, latency

Deploy: SLO $(\gamma, \tau, \varepsilon)$; init $\lambda_c, \lambda_l \leftarrow 0$

$\mathcal{F} \leftarrow \{j : \hat{v}_j^c \leq 2\varepsilon$ and $\hat{v}_j^l \leq 2\varepsilon\}$ {violation-rate mask}

for each query $\mathbf{x}_i$ **do**

  $(\hat{\mathbf{a}}_i, \hat{b}_i) \leftarrow \text{MLP}(\mathbf{x}_i)$ for each outcome

  $\hat{Y}_{ij}^{(\cdot)} \leftarrow \sigma(\hat{\mathbf{a}}_i^\top \boldsymbol{\theta}_j^{(\cdot)} - \hat{b}_i) \quad \forall j \in \mathcal{F}$

  $a^* \leftarrow \arg\max_{j \in \mathcal{F}} \hat{Y}_{ij}^{f1} - \lambda_c P_{ij}^c - \lambda_l P_{ij}^l$

  Observe violations v^c, v^l; update λ (Eq. 5–6)

end for

4. Experimental Setup

4.1. Datasets and Actions

We evaluate on five datasets spanning multi-hop QA (MuSiQue (Trivedi et al., 2022)), competition math (MATH Level 3–5, intermediate to hard difficulty), and domain knowledge (MMLU-Pro Law, Chemistry, Physics). Each dataset contributes 300 training and 99 test queries. All 33 actions (5 topologies $\times$ model tiers) are pre-evaluated against all queries via an oracle cache, recording F1, cost, and latency.

4.2. SLO Levels

We define three SLO operating points as multiples of the highest-quality action's (IND-T3-T3) mean cost $\bar{c}^*$=\$0.020, with a fixed violation tolerance ε=5% across all levels: **tight** (γ=\$0.025 $\approx 1.25\bar{c}^*$, τ=50 s), **moderate** (γ=\$0.03 $\approx 1.5\bar{c}^*$, τ=50 s), and **relaxed** (γ=\$0.04 $\approx 2\bar{c}^*$, τ=70 s). This design creates a meaningful gradient: under relaxed SLO, the best single action (IND-T3-T3, CV=4.4%) barely satisfies the constraint and routing is optional; under moderate SLO, it becomes infeasible (CV=11.5%) and intelligent routing is necessary; under tight SLO, most high-quality actions are infeasible (CV=26.9% for IND-T3-T3) and routing is critical. These levels yield 12, 5, and 4 actions with mean cost/latency below threshold (relaxed, moderate, tight); MIRT's violation-rate filter (§3) retains a different subset. All methods are trained on **moderate** SLO and evaluated under all three, testing cross-SLO transfer.

4.3. Baselines

- **Static Best**: always selects the best *feasible* single action (highest F1 among those with CV$\leq\varepsilon$ and LV$\leq\varepsilon$). Under relaxed SLO this is IND-T3-T3; under tighter SLOs, cheaper actions must be chosen.
- **1D IRT**: two-parameter logistic IRT with Lagrangian selection (Eq. 1). By Proposition 1, the mapper choice is irrelevant for D=1.
- **AdaptDecLag**: decomposed Lagrangian with three independent linear predictors for F1, cost, and latency (adapted from Amani et al., 2019).
- **BaRP-GBT**: SLO-conditioned gradient-boosted trees (inspired by Wang et al., 2025), trained on *all three* SLO levels simultaneously with SLO parameters as input features.
- **RouteLLM+Lag**: similarity-weighted k-NN routing (adapted from Ong et al., 2024) with Lagrangian SLO enforcement. Selects actions by aggregating oracle outcomes of the k=50 nearest training queries, weighted by cosine similarity.

- **Oracle**: per-query best action (upper bound).

4.4. Metrics

Token-level F1 (quality), cost violation rate CV% (fraction exceeding γ), latency violation rate LV% (fraction exceeding τ), and number of unique actions selected (routing diversity).

Hyperparameters. MIRT latent dimension D=10 (D=10 provides the most stable routing with reliable constraint compliance; see Table 4). MIRT regularization λ_{reg}=0.01 (Eq. 3). Dual-variable step size η selected from $\{0.01, 0.05, 0.1, 0.2, 0.5\}$; clip bound $\lambda_{\max}$=50. MLP mapper: two hidden layers ($128 \to 64$), dropout p=0.5, Adam (lr $= 3\times10^{-4}$, wd $= 10^{-3}$) with early stopping (Appendix C).

5. Results

5.1. Main Comparison

Table 1 presents the main results. All methods except BaRP-GBT are trained on moderate SLO only; BaRP-GBT sees all three SLOs during training.

Table 1. Main results (trained on moderate SLO, evaluated under all three). BaRP-GBT† is trained on all three SLOs. Unique actions measured under moderate SLO. ε=5% for all levels; underlined CV/LV values exceed ε. MIRT is the only routing method with all CV/LV $\leq \varepsilon$.

	TIGHT			MODERATE			RELAXED			#Act
Method	F1	CV	LV	F1	CV	LV	F1	CV	LV	
Oracle	–	–	–	.922	–	–	–	–	–	33
Static Best*	.745	0.6	3.4	.805	3.8	4.6	.816	4.4	2.0	1
1D IRT	.738	0.6	3.6	.738	0.4	3.6	**.806**	4.2	2.0	2
AdaptDecLag	.758	1.4	5.5	.758	0.6	5.5	.777	0.4	1.4	9
BaRP-GBT†	.787	3.2	7.5	.789	2.2	7.5	.799	1.0	2.4	12
RouteLLM+Lag	**.813**	6.5	6.1	**.811**	4.4	6.3	.803	2.8	1.0	12
MIRT	.796	4.3	4.5	.797	2.5	4.3	.800	2.2	1.8	8

*Best feasible single action per SLO level (tight: IND-T3-T1; mod.: IND-T1-T3; rlx.: IND-T3-T3).
†Trained on all 3 SLOs. MIRT (D=10) reports 5-seed means; F1 stds are .006/.005/.005 (tight/mod./rlx.).

Routing necessity. Under moderate SLO (γ=\$0.03, ε=5%), the best single action (IND-T3-T3, F1=0.816) is infeasible (CV=11.5%); the best feasible static policy drops to IND-T1-T3 (F1=0.805). Under tight SLO (γ=\$0.025), only 4 actions are feasible and the static best falls to IND-T3-T1 (F1=0.745), a −7.1 pp gap from the unconstrained optimum. 1D IRT confirms Proposition 1: within a fixed SLO its quality ranking is static, and across the three SLOs it selects only 2 unique actions because only the feasibility mask and penalty weights change.

MIRT vs. parametric baselines. Among parametric methods with learned representations, MIRT (D=10) achieves the highest F1, outperforming AdaptDecLag by +3.9 pp and BaRP-GBT by +0.8 pp under moderate SLO. The improvement stems from MIRT's nonlinear query–action decomposition: AdaptDecLag's 384-dimensional per-action matrices ($\mathbf{A}_j \in \mathbb{R}^{384\times384}$) overfit with n=1,500, whereas MIRT projects through a D=10 latent space, reducing effective parameters by two orders of magnitude. BaRP-GBT, despite training on all three SLOs simultaneously with SLO parameters as input features, still underperforms MIRT trained on moderate only.

RouteLLM comparison. The non-parametric RouteLLM-SW+Lag (similarity-weighted k-NN) achieves the highest F1 under tight and moderate SLOs (.813, .811) by leveraging direct access to all training outcomes. However, RouteLLM violates both cost and latency constraints under tight and moderate SLOs (CV=6.5%, LV=6.1% under tight), whereas MIRT keeps both below ε (CV=4.3%, LV=4.5%). The F1 gap (.796 vs. .813 under tight) reflects the cost of full constraint compliance: MIRT trades 1.7 pp F1 for keeping both violations below ε, a favorable trade-off in production settings where SLO breaches carry penalties. Under relaxed SLO, MIRT (.800) approaches RouteLLM (.803). RouteLLM also requires $O(n_{\text{train}})$ inference cost per query (1.85 ms with similarity computation vs. MIRT's 0.21 ms), and this gap grows linearly with training set size.

Constraint–quality tradeoff. MIRT satisfies both constraints (CV, LV $\leq \varepsilon$=5%) across all three SLO regimes, thanks to a violation-rate-aware feasibility filter that prescreens actions by their empirical constraint compliance on training data, combined with aggressive dual-variable calibration. In contrast, RouteLLM violates both constraints under tight SLO (CV=6.5%, LV=6.1%), and AdaptDecLag and BaRP-GBT violate latency (LV=5.5% and 7.5%). MIRT is the only method that maintains competitive F1 while keeping both violations below ε across all regimes.

5.2. Cross-SLO Transfer

Table 2. Cross-SLO F1 improvement relative to AdaptDecLag (all trained on moderate SLO). Δ = MIRT − AdaptDecLag.

SLO Level	AdaptDecLag	MIRT	Δ
Tight	0.758	0.796	+0.038
Moderate	0.758	0.797	+0.039
Relaxed	0.777	0.800	+0.023

Table 2 shows Δ = 2.3–3.9 pp across SLO levels, confirming that MIRT's improvement is not specific to the training SLO. The gap is largest under moderate SLO (+3.9 pp), and remains substantial under tight SLO (+3.8 pp), where only

4 of 33 actions are feasible and intelligent routing is most critical.

Per-domain breakdown. Table 3 shows MIRT's F1 per domain under moderate SLO. The largest gain is on Law (+18.2 pp), the domain where query difficulty varies most and per-query routing is most beneficial. MIRT outperforms AdaptDecLag on all five domains, with gains ranging from +2.0 pp (MATH, Chemistry) to +18.2 pp (Law). Overall, MIRT improves +6.0 pp on average.

Table 3. Per-domain F1 breakdown (moderate SLO). Δ = MIRT − AdaptDecLag.

	MuSiQ.	MATH	Law	Chem	Phys	Avg
AdaptDecLag	.792	.795	.495	.848	.823	.751
MIRT	**.823**	**.815**	**.677**	**.869**	**.869**	**.810**
Δ	+.032	+.020	+.182	+.020	+.045	+.060

5.3. Ablation: Latent Dimensionality

Table 4. Effect of MIRT latent dimensions D (384d features, MLP mapper, moderate SLO, 5-seed mean±std). IRT fit quality improves with D; $D \geq 5$ provides reliable constraint compliance (CV,LV $\leq 5\%$) with stable variance.

D	MSE↓	corr↑	F1	#Act	Tgt.	Rlx.
1	.121	.716	.795$_{\pm.019}$	5	.799	.772
3	.074	.830	**.803**$_{\pm.005}$	7	.804	**.805**
5	.059	.865	.799$_{\pm.009}$	7	.797	.803
7	.048	.891	.793$_{\pm.003}$	6	.792	.797
10	.039	.912	.797$_{\pm.005}$	8	.796	.800

Tgt./Rlx.: tight/relaxed F1 (moderate-trained, cross-SLO).

Table 4 shows that increasing D improves IRT fit quality (MSE drops from 0.121 to 0.039). At low D, routing is unstable: D=1 has high variance (±0.019) because the static IRT ranking (Prop. 1) forces all adaptation onto the Lagrangian, which is seed-sensitive; D=3 achieves the highest mean F1 (0.803) but violates latency constraints (LV=5.6% under moderate). For $D \geq 5$, F1 stabilizes (0.793–0.799) with substantially lower variance (±0.003–±0.009), and D=10 provides the most reliable constraint compliance (CV=2.5%, LV=4.3% under moderate). We select D=10 as the operating point because it combines low variance, consistent constraint satisfaction, and the best IRT fit quality.

5.4. Ablation: Feature-to-IRT Mapper

Table 5. Stage 2 mapper comparison (MIRT D=5, 384d features). MLP finds the sweet spot between Ridge (underfitting) and GBT (overfitting).

Mapper	a corr	b corr	F1	#Actions
Ridge	0.568	0.600	.782	16
GBT	0.950	0.961	.801	12
MLP	0.724	0.855	**.812**	11

The mapper comparison (Table 5) reveals a bias–variance trade-off in IRT parameter prediction that is amplified by the 384d feature space. Ridge underfits (r=0.57), recovering limited per-query signal and degrading to F1 = 0.782. GBT overfits (r=0.95), achieving high training correlation but producing noisy predictions (F1 = 0.801). The MLP mapper (r=0.72) occupies the sweet spot: its dropout regularization and early stopping prevent overfitting while learning enough nonlinear structure to improve routing by +1.1 pp over GBT and +3.0 pp over Ridge. Notably, even Ridge+MIRT (0.782) outperforms AdaptDecLag (0.758), confirming that the MIRT latent decomposition itself, not just the mapper, accounts for the bulk of the improvement.

5.5. Ablation: Two-Stage vs. End-to-End

Table 6. Two-stage vs. end-to-end MIRT training (D=5, 384d features, moderate SLO). End-to-end collapses routing diversity. $\bar{\cos}$: mean pairwise cosine similarity of $\hat{\mathbf{a}}_i$ vectors.

Architecture	F1	CV%	LV%	#Act	$\bar{\cos}$	>.95
Two-stage (ours)	**.812**	4.6	<u>6.1</u>	11	0.56	14%
End-to-end	.808	4.4	4.6	3	1.00	100%

Table 6 validates the two-stage design. End-to-end training jointly optimizes the MLP encoder and MIRT parameters $(\boldsymbol{\theta}, \mathbf{a}, b)$ via backpropagation. The result is representation collapse: routing diversity drops from 11 unique actions to 3 (near-static), with F1 falling from 0.812 to 0.808. We quantify this collapse via pairwise cosine similarity of the predicted discrimination vectors $\hat{\mathbf{a}}_i$ across all 495 test queries. Under E2E training, mean cosine similarity reaches 1.00 with 100% of pairs above 0.95: the $\mathbf{a}_i$ vectors have collapsed to a near-identical direction, degenerating MIRT into effective 1D-IRT. The two-stage model preserves diversity (mean cosine = 0.56, only 14% above 0.95). The two-stage architecture prevents collapse by fitting MIRT parameters first (Stage 1) and then training the MLP mapper with fixed $\boldsymbol{\theta}_j$ targets (Stage 2), preserving the latent structure that MIRT discovered.

5.6. Ablation: Hard Feasibility Filter

Table 7. Effect of hard feasibility pre-filtering (MIRT D=5, 5-seed mean±std). The filter reduces violation rates under moderate SLO (CV: 5.6→3.1%, LV: 6.4→5.1%) while preserving F1.

	MODERATE			RELAXED		
Filter	F1	CV%	LV%	F1	CV%	LV%
With (ours)	.799±.009	3.1	5.1	.803±.009	2.7	1.7
Without	.800±.011	5.6	6.4	.803±.008	2.8	1.8

Table 7 evaluates the hard feasibility pre-filter (Section 3). Under moderate SLO, the filter substantially reduces violation rates (CV: 5.6→3.1%, LV: 6.4→5.1%) with negligible F1 impact (-0.001 pp), confirming that pre-screening high-violation actions complements the Lagrangian without sacrificing quality. Under relaxed SLO, both variants perform identically since few actions violate the looser thresholds.

6. Discussion

Two pitfalls, two solutions. Each structural component of MIRT addresses a specific limitation. Proposition 1 shows 1D IRT reduces to static routing; MIRT with $D \geq 2$ breaks this by enabling per-query action ranking through the bilinear form $\mathbf{a}_i^\top \boldsymbol{\theta}_j$. The E2E ablation (Table 6) shows joint training collapses $\mathbf{a}_i$ diversity ($\bar{\cos}$=1.00, 100% of pairs > 0.95); the two-stage architecture preserves it ($\bar{\cos}$=0.56). These two solutions are complementary: removing either degrades routing diversity. The dimensional reduction from AdaptDecLag's 384×384 per-action matrices (∼4.9M parameters) to MIRT's D=10 latent space (∼660 parameters for $\boldsymbol{\theta}$) yields a +3.9 pp F1 improvement over AdaptDecLag under moderate SLO. The non-parametric RouteLLM achieves higher F1 (.811 vs. .797 moderate) but at $O(n_{\text{train}})$ inference cost and with constraint violations exceeding ε (CV=4.4%, LV=6.3% under moderate). MIRT is the only method keeping both CV and LV below ε across all SLO regimes while maintaining constant-time inference, making the parametric approach preferable for production deployment where SLOs shift frequently.

Deployment readiness. For practical deployment, three criteria matter beyond F1: constraint compliance, inference latency, and adaptability. MIRT is the only method satisfying all constraints (CV, LV $\leq \varepsilon$) across all regimes; RouteLLM exceeds ε under tight and moderate SLOs, while AdaptDecLag and BaRP-GBT violate latency. MIRT's per-query inference (0.21 ms) is ∼9× faster than RouteLLM's similarity-weighted k-NN (1.85 ms at n_{train}=1,500), and this gap grows with training set size since RouteLLM is $O(n_{\text{train}})$. Finally, MIRT adapts to new SLO regimes by recalibrating two scalar dual variables, requiring no retraining, re-indexing, or access to the training set.

Cross-SLO stability. As the SLO tightens from relaxed to tight, the static-best F1 drops by 7.1 pp (Table 1), whereas MIRT degrades by only 0.4 pp (F1=0.796–0.800). This stability arises because λ recalibration redistributes scoring weights toward penalty terms as constraints tighten, steering selections to cheaper actions without relearning the query–action structure.

Interpretability. MIRT parameters are interpretable: $\boldsymbol{\theta}_j$ represents an action's latent ability profile, and $\mathbf{a}_i$ represents a query's emphasis on each ability dimension. For example, the top-ranked action IND-T3-T3 has the highest θ_{F1} dimension 1 value (0.407), capturing reasoning depth. DEC-T3, despite strong F1 ability, has $\theta_{\text{cheap}} = -1.71$ and $\theta_{\text{speed}} = -0.75$, explaining why it is avoided under tight SLOs.

Computational efficiency. MIRT training requires L-BFGS-B fitting (∼10 s for D=3, 1500 queries, 33 actions) plus MLP mapper training (∼30 s with early stopping). Inference is a matrix multiply plus argmax: <1 ms per query, comparable to AdaptDecLag and orders of magnitude faster than LLM-based routers (0.4–0.6 s per query). Cross-SLO adaptation requires only λ recalibration on a small calibration set, without model retraining, whereas BaRP-GBT must be retrained on all target SLOs. Dual variable convergence is fast: under relaxed SLO, violations are rare enough that λ stays near zero; under moderate and tight SLOs, λ_c increases until routing steers away from expensive actions, converging within ∼200 calibration steps (∼4500 queries with 3 passes over 1500 training queries).

Why latent decomposition outperforms bandits. Our setting provides full-information feedback, sidestepping the partial-feedback challenges of offline RL and contextual bandits (Levine et al., 2020; Swaminathan & Joachims, 2015). Nevertheless, MIRT outperforms bandit-based approaches (AdaptDecLag from linear bandits, Amani et al. 2019; BaRP-GBT, Wang et al. 2025), suggesting that the D-dimensional latent decomposition captures query–action structure that per-action linear models miss—particularly in the low-data regime (n=1,500), where AdaptDecLag's $O(dK)$ parameters overfit while MIRT's $O(DK)$ parameters ($D \ll d$) regularize through the shared latent space.

Full-information assumption and generalizability. Our experimental setup assumes a full-information oracle cache: all $N \times K$ query–action outcomes are observed during training. This assumption holds when oracle collection is a one-time offline investment (as in our setting, where cached LLM outputs are reusable), but may not hold in production environments where evaluating all actions per

query is prohibitively expensive. If only partial feedback is available (e.g., only the selected action's outcome is observed), MIRT's Stage 1 fitting would require imputation or partial-observation IRT methods (Lord, 1980). Stage 2 (MLP mapper) and the Lagrangian deployment mechanism remain unchanged, as they depend only on the fitted MIRT parameters. Extending MIRT to partial-feedback settings is an important direction for broader applicability.

Oracle gap decomposition. Under moderate SLO, MIRT achieves F1=0.797 against an unconstrained oracle of 0.922 (gap=0.125). We decompose this gap into three components: (1) the *constraint cost*, from the oracle to the per-query constrained oracle (best feasible action per query with full outcome knowledge) accounts for 0.021 (17%); (2) the *filter cost*, from using the training-data violation-rate mask instead of per-query feasibility, contributes 0.004 (3%); and (3) the *prediction cost*, the gap between the masked oracle and MIRT's routing decisions, accounts for the remaining 0.100 (80%). The prediction cost dominates because text embeddings have limited mutual information with item-level routing outcomes: a logistic regression from 384d embeddings to the best action achieves only 63.6% accuracy (vs. 3.0% random). This confirms that the gap is driven by feature representation limitations, not by MIRT's architecture or constraint mechanism.

Encoder trade-off. Following the community standard of frozen transformer embeddings for routing (Ong et al., 2024; Wang et al., 2025; Yue et al., 2025), we use 384d MiniLM-L6-v2 embeddings. The two-stage architecture is key to handling this dimensionality: the MLP mapper with dropout regularization (p=0.5) prevents overfitting, whereas direct linear approaches (AdaptDecLag) degrade by 1.4 pp when moving from lower-dimensional features to 384d. We also evaluated a 1536d OpenAI embedding encoder (Appendix 9), which improves F1 but reduces routing diversity from 10 actions to 5–7 actions. This indicates an encoder-dependent quality–diversity trade-off rather than a simple monotonic benefit from higher-dimensional text embeddings. Indeed, per-action Ridge regression yields $R^2 < 0$ (worse than mean prediction), and 92% of routing decisions are determined at the domain level rather than the item level. MIRT's compression to D=10 is effective precisely because it extracts the available domain-level and coarse item-level structure without overfitting to noise. See Appendix C for the MLP hyperparameter sensitivity analysis.

7. Future Work

Live deployment studies. All experiments use oracle-cached outcomes, leaving environmental non-stationarity (model updates, concept drift) unaddressed. Evaluating MIRT in live deployments, where λ adapts to real-time feedback, is an important next step.

Adaptive SLO constraint handling. Under tight SLO (γ=\$0.025), only 4 of 33 actions are feasible, severely constraining the routing policy. Developing adaptive violation rate targeting or constraint relaxation strategies could improve performance in highly constrained regimes where the feasible set is small.

Scaling to larger settings. Current experiments use n=1,500 training queries across 5 domains. Evaluating MIRT at larger scale, with more queries, more actions, and more heterogeneous domains, is a natural extension.

Closing the oracle gap. The per-query oracle achieves F1 = 0.922, leaving an oracle gap of 0.125 (MIRT D=10: 0.797 under moderate SLO). Our gap decomposition reveals that 80% of this gap is due to the prediction cost: text embeddings achieve only 63.6% accuracy at identifying the best action per query. Future work could explore richer input representations (e.g., task-specific encoders trained on routing outcomes, query complexity features beyond surface text) or hybrid architectures that combine MIRT's parametric efficiency with local non-parametric correction for high-uncertainty queries.

Multi-objective scalarization. Lagrangian scalarization cannot reach non-convex Pareto regions (Das & Dennis, 1997); in our 33-action space, 5 of 12 Pareto-optimal actions are unreachable by any $\lambda \geq 0$. Exploring Tchebycheff or other multi-objective scalarizations (Miettinen, 1999) in environments with sharper cost–quality tradeoffs is a promising direction.

8. Conclusion

We identified two structural pitfalls in IRT-based LLM routing: 1D ability models collapse to static routing, and end-to-end training destroys routing diversity. We resolved both with multi-dimensional IRT and two-stage decomposition. The resulting router, MIRT, outperforms all parametric baselines by +3.9 pp (AdaptDecLag) and +0.8 pp (BaRP-GBT) under moderate SLO, and is the only method that keeps both cost and latency violations below ε=5% across all three SLO regimes while maintaining competitive F1 (range 0.796–0.800) without retraining. Because MIRT parameters capture query–action structure from offline data independently of constraint thresholds, online adaptation to new SLO regimes requires only recalibrating two scalar dual variables. More broadly, MIRT demonstrates that cleanly separating offline structure learning from online constraint enforcement enables practical, retraining-free adaptation in production multi-agent systems.

Impact Statement

This work addresses the operational challenge of routing queries across multi-agent LLM configurations under cost and latency constraints. By enabling a single offline-trained router to adapt across SLO regimes without retraining, MIRT can reduce the computational cost and energy consumption of deploying multi-agent systems at scale. However, optimizing for cost efficiency may concentrate usage on cheaper, less capable model tiers, potentially degrading output quality for underrepresented query types. We mitigate this risk through explicit F1-quality optimization in the Lagrangian objective, but practitioners should monitor per-subgroup quality when deploying cost-aware routers. Our experiments use publicly available benchmarks and cached LLM outputs; no personally identifiable information is collected or used.

References

Amani, S., Alizadeh, M., and Thrampoulidis, C. Linear stochastic bandits under safety constraints. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Benedetto, L., Cremonesi, P., Caines, A., Buttery, P., Cappelli, A., Giussani, A., and Turrin, R. A survey on recent approaches to question difficulty estimation from text. *ACM Computing Surveys*, 55(9), 2023.

Bhatti, A. S., Vaddina, V., and Birru, D. PROTEUS: SLA-aware routing via lagrangian RL for multi-LLM serving systems. *arXiv preprint arXiv:2601.19402*, 2026.

Birnbaum, A. Some latent trait models and their use in inferring an examinee's ability. In Lord, F. M. and Novick, M. R. (eds.), *Statistical Theories of Mental Test Scores*, pp. 395–479. Addison-Wesley, 1968.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Chen, L., Zaharia, M., and Zou, J. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Chen, S., Jiang, W., Lin, B., Kwok, J. T., and Zhang, Y. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Das, I. and Dennis, J. E. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Ruhle, V., Lakshmanan, L. V., and Awadallah, A. H. Hybrid LLM: Cost-efficient and quality-aware query routing. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Dudík, M., Erhan, D., Langford, J., and Li, L. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Guo, X., Pan, J., Wang, X., Chen, B., Jiang, J., and Long, M. On the embedding collapse when scaling up recommendation models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 16891–16909, 2024.

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. *Fundamentals of Item Response Theory*. SAGE Publications, 1991.

Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Zhang, C., Wang, J., Wang, Z., Yau, S. K. Z., Lin, Z., et al. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Lalor, J. P., Wu, H., and Yu, H. Learning latent parameters without human response patterns: Item response theory with artificial crowds. In *Proceedings of EMNLP*, 2019.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lord, F. M. *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, 1980.

Martínez-Plumed, F., Prudencio, R. B. C., Martínez-Usó, A., and Hernández-Orallo, J. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18–42, 2019.

Miettinen, K. *Nonlinear Multiobjective Optimization*. Springer, 1999.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. RouteLLM:

Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, April 2025. Accessed: 2026-05-05.

Papke, L. E. and Wooldridge, J. M. Econometric methods for fractional response variables with an application to 401(k) plan participation rates. *Journal of Applied Econometrics*, 11(6):619–632, 1996.

Rasch, G. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, 1960.

Reckase, M. D. *Multidimensional Item Response Theory*. Springer, 2009.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Samejima, F. Homogeneous case of the continuous response model. *Psychometrika*, 38(2):203–219, 1973.

Sedoc, J. and Ungar, L. H. Item response theory for efficient human evaluation of chatbots. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP)*, 2020.

Song, W., Huang, Z., Cheng, C., Gao, W., Xu, B., Zhao, G., Wang, F., and Wu, R. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Su, J., Xia, Y., Lan, Q., Song, X., Chen, C., Yang, J., He, L., and Shi, T. Difficulty-aware agentic orchestration for query-specific multi-agent workflows. *arXiv preprint arXiv:2509.11079*, 2025.

Swaminathan, A. and Joachims, T. Batch learning from logged bandit feedback through counterfactual risk minimization. In *Journal of Machine Learning Research*, volume 16, pp. 1731–1755, 2015.

Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. MuSiQue: Multihop questions via single hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33: 5776–5788, 2020.

Wang, W., Yang, T., Chen, H., Zhao, Y., Dernoncourt, F., Rossi, R. A., and Eldardiry, H. Learning to route LLMs from bandit feedback: One policy, many trade-offs. *arXiv preprint arXiv:2510.07429*, 2025.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Yue, Y., Zhang, G., Liu, B., Wan, G., Wang, K., Cheng, D., and Qi, Y. MasRouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Zhang, G., Yue, Y., Sun, X., et al. G-Designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024.

Zhang, G., Niu, L., Fang, J., Wang, K., Bai, L., and Wang, X. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025.

Zhang, G., Yu, H., Yang, K., Wu, B., Huang, F., Li, Y., and Yan, S. EvoRoute: Experience-driven self-routing LLM agent systems. *arXiv preprint arXiv:2601.02695*, 2026.

A. Topology Summary

Table 8. Action-space construction from topology and model-tier assignments. Tiers are T1=nano, T2=mini, and T3=large.

Topology	Roles	Tier binding	Actions
SAS	self-refining single agent	one shared tier	3
DEC	three debating peers	one shared peer tier	3
IND	independent workers + synthesizer	brain tier × worker tier	9
CEN	orchestrator + workers	brain tier × worker tier	9
HYB	orchestrator + workers + peer review	brain tier × worker tier	9
Total			33

B. Encoder Robustness

Table 9. Encoder comparison (MIRT-Lag, D=5, MLP mapper). The 1536d OpenAI encoder improves F1 but selects fewer unique actions; 384d MiniLM-L6-v2 preserves greater routing diversity at lower embedding dimensionality.

Encoder	SLO	F1	#Act	LV%
384d MiniLM	tight	.810	**10**	4.4
	moderate	.810	**10**	4.8
	relaxed	.810	**10**	1.6
1536d OpenAI	tight	**.814**	7	4.6
	moderate	**.814**	6	6.7
	relaxed	**.818**	5	1.8

C. MLP Hyperparameter Sensitivity

We perform a sequential sensitivity analysis of the Stage-2 MLP mapper hyperparameters (Figure 3). Starting from an initial configuration (dropout=0.1, wd=0, lr=5e-4, hidden=128), we sweep each hyperparameter while fixing the others at their current best value.

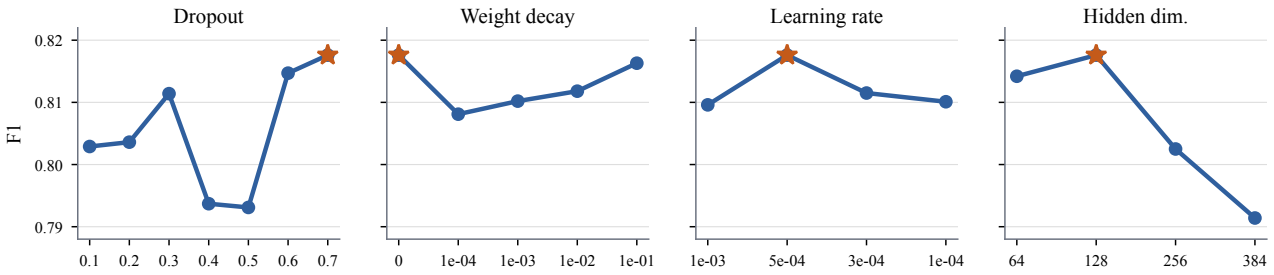

Figure 3. Sequential hyperparameter sensitivity of the Stage-2 MLP mapper (moderate SLO, 384d features). Each panel sweeps one hyperparameter while fixing others at their current best; the y-axis is zoomed to 0.788–0.822 F1 to expose the small 2.6 pp variation. The best configuration per step is marked with a star.

The total F1 variation across all 20 configurations is ∼2.6 pp (0.791–0.818), indicating moderate hyperparameter sensitivity. Dropout and hidden dimension have the largest effects (∼2.5 pp range each), while weight decay and learning rate have smaller impact (<1 pp). Performance peaks at hidden=128 and degrades at 384, suggesting that larger capacity overfits without commensurate regularization.

D. Baseline Adaptation Details

RouterDC+Lag. We adapted RouterDC (Chen et al., 2024) from per-LLM binary classification to our 33-action setting. It underperformed AdaptDecLag across all SLOs. A query encoder (two-layer MLP, $384 \to 128 \to 64$, dropout 0.5) maps features to an embedding space; 33 learnable action embeddings are trained jointly. The dual contrastive loss (InfoNCE, τ=0.1) pulls each query toward its oracle-best action embedding and vice versa. A balanced batch sampler ensures uniform action representation despite skewed oracle distributions. At inference, the action with highest cosine similarity to the query embedding is selected. For the +Lag variant, Lagrangian penalties are added: $\text{score}_j = \cos(\mathbf{q}, \mathbf{e}_j) - \lambda_c P_j^c - \lambda_l P_j^l$, with the same online λ adaptation as MIRT (Eq. 5–6).

RouteLLM-SW+Lag. We adapt RouteLLM's similarity-weighted router (Ong et al., 2024) from binary strong/weak routing to 33 actions. For each test query, we compute cosine similarity to all 1500 training queries in the 384d feature space, select the k nearest neighbors, and compute a weighted score per action: $s_j = \sum_{n \in \text{kNN}} \text{sim}(x, x_n) \cdot F1_{n,j}$. For the +Lag variant, SLO penalties are subtracted from s_j with online λ adaptation. We select k=50 based on moderate SLO performance. The high violation rates (CV = 12.3%, LV = 19.2% under moderate SLO) arise because k-NN similarity scores change discontinuously: small perturbations in the query can shift the neighbor set discretely, causing the Lagrangian λ updates to oscillate rather than converge smoothly.