FidelityScope: An Optimization Framework for Diagnosing and Improving Distributional Fidelity of Synthetic Tutoring Dialogues

Anonymous ACL submission

Abstract

Synthetic corpora are playing an increasingly critical role in AI system training and evaluation, particularly in educational settings where authentic tutor/student interaction data is difficult or costly to obtain at scale. Practitioners generating synthetic dialogue corpora lack a diagnostic tool for detecting *which* distributional properties their generator gets wrong and takng corrective action to improve the synthetic corpus quality. We introduce *FidelityScope*, an optimization framework for generating high-quality synthetic conversation corpora. Our framework decomposes fidelity into four dimensions (Surface, Content, Dynamics, Outcome), evaluated using propensity score mean-squared error (pMSE) produced by a classifier providing feature importance scores, making it possible to perform diagnosis and iterative refinement of the synthetic generation process. Applied to mathematics tutoring dialogues, the framework diagnoses why iterative prompt rewriting fails to close the distributional gap – whether done by a researcher or an automated LLM reviser – and proposes corrective action to improve corpus quality. When a single prompt controls all discourse features simultaneously, editing to fix one feature typically shifts others unpredictably. Furthermore, post-training alignment concentrates LLM outputs towards phrasings the discourse classifier systematically mislabels, making intended move rates unachievable through prompt manipulation alone.

We introduce a factored multi-phase generation architecture that separates a conversation's structure from its surface content and adds a quality control post-generation step, and show that this reduces pMSE from 50.6% to 10.54%, within 2 pp of optimal quality. We also apply our approach to a second corpus (QATD-2k), achieving a pMSE of 40.53%. FidelityScope and reference corpora are publicly available.

1 Introduction

A practitioner generating synthetic tutoring dialogue for training or simulation faces a diagnostic gap: per-sample quality metrics such as BERTScore (Zhang et al., 2020) or G-Eval (Liu et al., 2023) provide estimates that individual messages look realistic, but cannot identify *which* corpus-level discourse properties the generator gets wrong, e.g., whether students express confusion too rarely, or whether tutors over-scaffold relative to real practitioners, whether conversations resolve too quickly. These properties determine whether synthetic data produces biased training signal downstream. Without a corpus generation process that decomposes the distributional fidelity problem into actionable dimensions, practitioners have no systematic way to diagnose or improve their generator efficiently.

We introduce *FidelityScope*, a corpus generation optimization framework that incorporates four-dimensional distributional fidelity evaluation (spanning Surface, Content, Dynamics, and Outcome), scored using propensity mean-square error (pMSE) (Snoke et al., 2018). Unlike embedding-space metrics that confirm a fidelity gap exists without diagnosing its source, FidelityScope's evaluation approach identifies *which* features drive the gap, in which direction, and by how much—making diagnosis actionable rather than merely evaluative. Our optimization framework aims to be corpus-agnostic and useful for any discourse-annotated corpus with a per-message move classifier that could support the same evaluation. Within our framework, we explore the following research questions around addressing the fidelity gap in synthetic conversational data:

RQ1. Does prose generation (i.e., a single whole-conversation prompt with no imposed structure) produce distributionally faithful dialogue, and can iterative revision fix the gaps?

RQ2. Does our factored generation architecture, where each component targets a specific dimension separately, help improve corpus fidelity, and if so, by how much?

RQ3. Does our factored generation architecture transfer to a structurally distinct corpus, and what limits its effectiveness there?

Related work (Ion and Collins-Thompson, 2026) succeeded in reducing pMSE for a synthetic conversational dataset from 97.4% to 50.6% through four rounds of manual prompt revision, but was unable to achieve further iterative improvements: each revision fixed one dimension while degrading others. This oscillation appears structural to the synthesis generation problem, not a prompt-quality problem. Consider attempting to tune dozens of correlated conversation features by rewriting prose instructions. A single prompt mediates everything the LLM produces, so correcting the scaffolding rate also changes message register, length, and every other feature simultaneously. We call this phenomenon *prompt entanglement*. Compounding this, post-training alignment concentrates the LLM's outputs toward modal phrasings: a shorter positive confirmation (Confirm-Positive) move looks like an Acknowledgment move to the classifier; a Knowledge-Gap mentioning "ugh" looks like Frustration. The classifier's confusion structure is problematic to optimize directly through prose alone; we term this *typicality-classifier coupling*: alignment sharpens outputs toward modal phrasings that the classifier systematically misreads as adjacent move types, causing measured feature rates to diverge from intended rates in ways no prose revision can detect or correct.

In Section 3 we introduce a *factored corpus generation architecture* that addresses prompt entanglement directly by decomposing the fidelity optimization problem into distinct phases that do both prompt- and statistical feature-optimization jointly across distinct dimensions of fidelity. Typicality-classifier coupling is addressed with a Quality Control phase that checks what was specified vs. what was actually generated, implemented in this study using *classifier-guided rejection sampling* (CRS): for each misclassified message, the LLM generates diverse candidate phrasings and the discourse classifier selects the first labeled as the intended move type, recovering phrasings that alignment had suppressed (Zhang et al., 2025).

2 Related Work

Fidelity metrics for synthetic text. Evaluation of synthetic text corpora operates at three levels with different diagnostic power. *Per-sample lexical and semantic metrics* (BLEU, ROUGE, BERTScore (Zhang et al., 2020)) and *LLM judges* (G-Eval (Liu et al., 2023), USR (Mehri and Eskenazi, 2020)) assess whether individual utterances are realistic; they cannot detect whether the *corpus* reproduces global distributional structure. *Embedding-space distribution metrics* (e.g., kernel MMD) operate at corpus level but yield a scalar with no feature attribution: they confirm a fidelity gap exists without diagnosing its source (Gretton et al., 2012) *Feature-vector pMSE* (Snoke et al., 2018; Woo et al., 2009) operates at corpus level over an interpretable feature vector; the L1 classifier's non-zero coefficients identify which features drive discrimination, making it the only approach that is both corpus-level and diagnostically actionable. We extend pMSE from tabular privacy evaluation to conversational data, adding a four-dimensional fidelity feature decomposition, a real-vs-real null hypothesis calibration, and a public benchmark with reference corpora.

Synthetic dialogue generation and evaluation. Synthetic dialogue is used for data augmentation (Mohapatra et al., 2021), task-oriented dialogue (Soudani et al., 2024), and education applications (Markel et al., 2023; Lu and Wang, 2024; Marquez-Carpintero et al., 2025), typically evaluated via BERTScore (Zhang et al., 2020) or model-based judges (Mehri and Eskenazi, 2020; Liu et al., 2023). Such per-sample metrics cannot detect whether a generated *corpus* reproduces global distributional properties; FidelityScope is intended to address this gap.

Prompt optimization. Automated prompt revision (Zhou et al., 2023; Yang et al., 2024; Khattab et al., 2024; Fernando et al., 2024; Guo et al., 2024; Yan et al., 2025) has been used to optimize a scalar metric by a search across variants of prose used to define natural language prompts. However, we find that the dialogue generation problem involves balancing a complex union of structure, content, and statistical properties that are problematic to optimize using prose-space search alone: we have observed that any prose edit can shift multiple features simultaneously in unpredictable directions and that the distributional objectives expose a dif-

ferent bottleneck—the search space itself rather than the optimizer.

3 Corpus Generation Framework

We designed a corpus generation framework that addresses two structural failure modes that our baseline benchmark analysis identified: prompt entanglement and typicality-classifier coupling.

3.1 Why Prose Revision Fails

We discuss the standard approach to corpus generation (iterative prompt prose revision) and analyze why it fails. In one example of using iterative prompt refinement, Ion and Collins-Thompson (2026) applied pMSE as a diagnostic for four manual prompt revisions (V0→V4) but were unable to obtain better than 50.6% combined pMSE on their dataset. Our work initially attempted to improve on their approach by replacing manual iteration with a reviser LLM that receives the L1 coefficients and generates conversation content using a single prompt. However, we were not successful: we believe this failure has two structural causes.

(i) Prompt entanglement. A whole-conversation prompt has the potential to control and affect all features simultaneously. A prompt change to 'Reduce scaffolding' nudges the scaffolding rate down but also changes register, message length distributions, and the overall tutor move category distribution. Each revision's side effects on features that the previous iteration fixed are unpredictable.

(ii) Typicality-classifier coupling. A move category classifier (e.g. with 24 possible tutor/student categories) has non-trivial confusion structure. A Confirm-Positive message that gets shorter and more affirming reads as Acknowledgment; a Knowledge-Gap message mentioning "ugh" looks like Frustration. Prose edits aimed at one move type drift the *measured* rate of two or three others in directions the reviser cannot anticipate. The L1 coefficients tell the reviser *which* features are off, but prose cannot navigate the classifier-confusion coupling.

The V1→V2→V3 oscillation in the original sequence is exactly this failure mode. We replicated it on the 24-move taxonomy: iteration 2 reduced Surface pMSE from 28% to 21% but doubled Content pMSE from 14% to 32%, and was rolled back. This failure is structural to prose-space optimization; tightening the loop (per-move directives, exemplar memory, iterative Monte Carlo search (Zhou et al., 2023)) does not escape it because the bottleneck is the search space itself, not the optimizer (Zhou et al., 2023; Yang et al., 2024; Khattab et al., 2024).

3.2 Corpus generation architecture

Figure 1 shows our generation optimization pipeline. Our high-level objective is to find the optimal corpus generation specification Ω that guides a workflow based on a Large Language Model (LLM) that generates synthetic tutor-student conversations. These conversations must be statistically indistinguishable from real human tutoring interactions when evaluated by a supervised discriminator model using surface, cognitive, and conversational dynamics and outcomes features.

We formalize the optimal corpus generation problem as follows. Let $\mathcal{D}_{\text{real}} = \{x_1, x_2, \ldots, x_N\}$ be a fixed dataset of N true human tutor-student conversations. We define a formal generative process $\mathcal{G}$ for creating a synthetic conversation corpus that uses generation parameters $\Omega = (\Omega_C, \Omega_M, \Omega_E)$ respectively containing generation parameters for the Conversation Corpus, Moves, and Exemplars. Let P represent a prompt configuration (or a vector of prompt parameters/templates) chosen from a feasible prompt space $\mathcal{P}$. This study applies an LLM instructed with P in the role of $\mathcal{G}$ to generate a synthetic conversation $x_{\text{synth}} \sim \mathcal{G}(\cdot \mid \Omega)$ when conditioned on generation parameters Ω. By generating N conditionally independent conversations, we construct a synthetic dataset $\mathcal{D}_{\text{synth}}(\Omega)$ such that $|\mathcal{D}_{\text{synth}}(\Omega)| = |\mathcal{D}_{\text{real}}| = N$.

Feature Extraction. We organize fidelity along four dimensions, each answering a distinct question. The dimensions are independently violable—a system can match Surface while failing Content, or Content while failing Dynamics—and require progressively richer annotation: Surface needs only tokenization; Content needs per-message discourse classification (The full 24-move taxonomy is given in Appendix B; Dynamics adds sequential structure; Outcome adds conversation-level labels.) Table 1 summarizes the complete feature set.

Following this partition, we define a feature extraction mapping $\phi : X \to \mathbb{R}^d$ that maps any conversation text x into a d-dimensional feature vector. The features are partitioned into distinct classes $\phi(x) = [\phi_{\text{S}}(x),\ \phi_{\text{C}}(x),\ \phi_{\text{D}}(x),\ \phi_{\text{O}}(x)]$ where:

- $\phi_{\text{S}}(x)$ represents structural surface text statistics (e.g., word counts, type-token ratio, read-

Table 1: Full feature set ($N = 43$) across four fidelity dimensions. †Confusion episode: Knowledge-Gap or Frustration → Inference-Understanding or Breakthrough; *phase* = equal thirds of the turn sequence.

Dim.	N	Main features	Answers
Surface	6	Per-role msg. length (mean, SD); turn count; turn ratio	Do they *look* real?
Content	24	Per-move rate (one per 24-move taxonomy)	Do the right *moves* occur?
Dynamics	10	Episode† duration; instant/quick/extended resolution rates; scaffolding density per phase	Do moves *unfold* realistically?
Outcome	3	`resolved`; `has_breakthrough`; `has_off_topic`	Do convs. *end* like real?

ability indices).

- $\phi_{\text{C}}(x)$ represents pedagogical indicators (e.g., Bloom's taxonomy levels, depth of student questions, scaffolding markers).
- $\phi_{\text{D}}(x)$ represents conversational patterns over time (e.g., turn-taking latency, speaker balance, speech act transition probabilities).
- $\phi_{\text{O}}(x)$ represents desired outcome of the conversation arc.

Propensity score objective (pMSE). We extend the application of pMSE (Snoke et al., 2018) from tabular data to conversational data via the per-conversation feature vector $\phi(\cdot)$ to use propensity score mean squared error (pMSE) as our overall objective to be optimized. To compute pMSE, the real and synthetic datasets are merged into a balanced evaluation dataset $\mathcal{D}_{\text{eval}}(\Omega) = \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}(\Omega)$ of size $2N$. Each sample i in $\mathcal{D}_{\text{eval}}(\Omega)$ consists of its extracted features $z_i = \phi(x_i)$ and a binary label $y_i \in \{0, 1\}$ indicating its origin ($y = 1$ if synthetic, 0 otherwise). The evaluation uses a K-fold cross-validation scheme ($K = 5$ for this study). For each fold $k \in \{1, \ldots, K\}$, a supervised discriminator model f_{θ_k} is trained on the training split to predict the probability that a sample is synthetic. In this study our supervised model is an L1 logistic regression classifier using the feature set $\phi(\cdot)$. For a sample i belonging to the validation split of fold k, the model outputs the estimated propensity score:

$$\hat{p}(z_i) = f_{\theta_k}(z_i) \approx P(y_i = 1 \mid z_i) \tag{1}$$

The final pMSE measures are then computed using:

$$\text{pMSE}_{\text{raw}} = \frac{1}{2N} \sum_{i=1}^{2N} (\hat{p}_i - c)^2 \tag{2}$$

$$\text{pMSE}_{\text{norm}} = \frac{\text{pMSE}_{\text{raw}}}{c(1-c)} \times 100\% \tag{3}$$

At $c = 0.5$, $\text{pMSE}_{\text{norm}} \in [0\%, 100\%]$: 0% = classifier at chance, 100% = perfect separation. The non-zero L1 coefficients $\mathbf{w}$ identify which features drive the discrimination. In computing pMSE we also compute the expected pMSE under the *empirical null distribution* that compares real data against itself. In our dataset we have $N = 500$ per side, 20 random half-splits of 2,000 held-out real conversations, giving an empirical null value using all features of 8.2% (Surface 1.42%, Content 4.42%, Dynamics 1.66%, Outcomes 0.83%). All synthetic pMSE numbers for MMDB are interpreted against this empirical null lower bound.

Unified optimization problem. Bringing together the previous components, we frame the corpus fidelity optimization task as finding an optimal generation scheme Ω^* that minimizes the expected normalized pMSE over the distribution of generated datasets:

$$\begin{aligned} \Omega^* = \arg\min_{\Omega} \mathbb{E}_{\mathcal{D}_{\text{synth}}(\Omega)} \left[\text{pMSE}\left(\mathcal{D}_{\text{real}}, \mathcal{D}_{\text{synth}}(\Omega)\right) \right] \\ \text{s.t.} \quad x_i \sim \mathcal{G}(\cdot \mid \Omega), \quad \forall x_i \in \mathcal{D}_{\text{synth}}(\Omega) \\ z_i = \phi(x_i), \quad \forall x_i \in \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}(\Omega) \\ \hat{p}(z_i) = \text{MCVD}\left(z_i; \mathcal{D}_{\text{eval}}(\Omega)\right) \end{aligned} \tag{4}$$

where MCVD is the mean cross-validated propensity score $\hat{p}$ for conversation x_i described above.

The search space for Ω is highly complex, combining components that may be natural language text, statistical parameters, and selected training exemplars. Thus, this problem does not lend itself to standard gradient-based approaches. To make solving Eq. 4 more tractable we define a structure on the generative specifications that make up Ω space by defining a conversation corpus in terms of a modular generative process, described next.

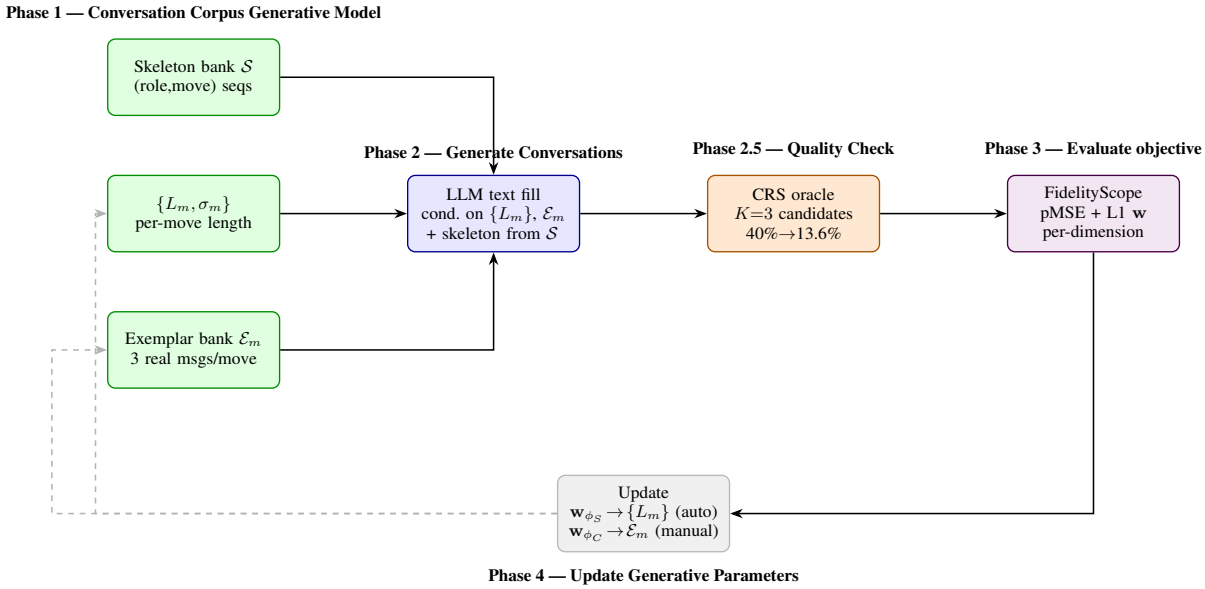

Figure 1: Factored generation pipeline. **Phase 1** defines a generative model for a corpus of conversations; the three Phase 1 components are the controllers for each feature group (Table 2): skeleton bank $\mathcal{S}$ pins Content move rates (C1); per-move length targets $\{L_m\}$ control Surface (C2); exemplar bank $\mathcal{E}_m$ anchors phrasing. **Phase 2** assembles each synthetic conversation using an LLM conditioned on $\{L_m\}$, $\mathcal{E}_m$, and a skeleton sampled from $\mathcal{S}$. **Phase 2.5** Quality Check applies CRS—the Content residual controller (C3)—re-sampling misclassified messages up to K=3 times. **Phase 3** evaluates the corpus against the pMSE objective and outputs per-dimension L1 weights $\mathbf{w}$. **Phase 4** routes each non-zero weight w_k to its dedicated controller (Table 2): Surface weights update $\{L_m\}$ automatically (dashed); Content weights flag exemplar bank updates (dashed).

3.2.1 Phase 1: Conversation Corpus Generative Model

Algorithm 1 Synthetic Corpus Generation Process

Require: Generation process $\mathcal{G}$, move feature mapping $\mathcal{F}$
Ensure: Synthetic dialogue corpus $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$
Initialize empty corpus $\mathcal{C} \leftarrow \emptyset$
Sample a total count N of target conversations to generate

for each conversation $i \in \{1, 2, \ldots, N\}$ **do**
  Sample a target number of turns T_i
  Initialize empty turn skeleton sequence $S_i \leftarrow \emptyset$
  Initialize empty conversation text block $c_i \leftarrow \emptyset$
  for each turn $t \in \{1, 2, \ldots, T_i\}$ **do**
    Generate a role, move pair $(r_{i,t}, m_{i,t})$ and a move length $\ell_{i,t}$ stored with the pair
    Append $(r_{i,t}, m_{i,t})$ to the conversation skeleton: $S_i \leftarrow S_i \cup \{(r_{i,t}, m_{i,t})\}$
    Generate move feature vector $f_{i,t} = \mathcal{F}(m_{i,t})$
  end for
  for each move $m_{i,t} \in S_i$ **do**
    Generate full move text: $r_{i,t} \leftarrow \mathcal{G}(S_i, \mathbf{f}_i)$
    Add to conversation: $c_i \leftarrow c_i \cup \{r_{i,t}\}$
  end for
  Add to corpus: $\mathcal{C} \leftarrow \mathcal{C} \cup \{c_i\}$
end for
return $\mathcal{C}$

As a starting point, we define: (1) a generative model for creating a corpus $\mathcal{C}$ of conversations as shown in Alg. 1; and (2) a structural skeleton $\mathcal{S}$ as a sequence of length T turns defining a conversation's turn structure as a series of move types but without surface text realization.

For this study, we set $N = |\mathcal{D}_{real}|$, and then as a simple way to preserve overall conversation arcs and turn-taking dynamics, each skeleton $\mathcal{S}_i$ is drawn from an empirical non-parametric distribution over all skeletons extracted from the training subset of the real data. This ensures correct turn structure, correct skeleton-level move rates (each move's intended frequency matches the real distribution exactly), and correct long-range conversation arcs. With 5,000 unique skeletons, sampling with replacement provides ample diversity. Investigating more refined sampling methods is a topic for future work.

In sum, this generative model decomposes a conversation into a structural skeleton (*what moves to make, in what order*) and a per-message text fill (*how to phrase a given move*).

Move length sampling. For each move m, we compute from real training data the mean length L_m and lognormal scale σ_m:

$$\text{length}(m) \sim \text{LogN}\big(\log L_m - \tfrac{1}{2}\sigma_m^2,\ \sigma_m\big)\,,$$
$$\mathbb{E}[\text{length}(m)] = L_m.$$

Table 2: FidelityScope feature-partition: each ϕ component assigned to its natural non-prose controller. Prose controllable = whether iterative prose revision can close this gap (Section 3.1).

Dimension	Feat.	Controller	Mechanism	Automated?
Surface (ϕ_S)	6	$\{L_m, \sigma_m\}$	Lognormal length conditioning (C2)	Phase 4
Content: rates (ϕ_C)	24	Skeleton $\mathcal{S}$	Real sequence sampling (C1)	Yes
Content: resid. (ϕ_C^+)	—	CRS oracle	Classifier-gated resampling (C3)	Yes
Dynamics (ϕ_D)	10	Skeleton $\mathcal{S}$	Sequence structure preservation	Manual
Outcome (ϕ_O)	3	Skeleton $\mathcal{S}$	Conversation arc preservation	Manual

At generation time, the text-fill prompt specifies "write approximately ℓ words" where $\ell \sim \text{Lognormal}(L_m, \sigma_m)$. Phase 4 updates $\{L_m\}$ iteratively via proportional feedback (see 3); The CRS quality check (C3) runs automatically per message; exemplar bank updates are performed manually between runs. Real per-move means span from 3 (Confirm-Positive) to 13 words (Explain-Problem). Without conditional targets, LLMs fall back to a uniform 5-10-word default that mismatches roughly two-thirds of move types.

3.2.2 Phase 2: Generate Conversations

For each skeleton slot, the LLM receives as metadata for the move creation prompt: the desired move label, three real exemplars from the move's exemplar bank, the per-move target length ℓ, and optional negative exemplars from a confusable move (e.g., for Confirm-Positive, Encouragement messages tagged "do NOT write like these").

3.2.3 Phase 2.5: Quality Check

The Quality Check step compares the generative specification output by Phase 1 against the results from generator $\mathcal{G}$ in Phase 2, and does error correction of the output as needed. In fact, on a single-pass run using MMDB (Table 3) 40% of messages received a different label than the skeleton S_i specified, inflating Content pMSE to 28.56% despite the intended distribution being correct. Our approach is based on related work by Zhang et al. (2025) who showed that post-training alignment concentrates LLM output mass at modal phrasings, and that prompting the model to generate K diverse candidates in a single call recovers non-modal outputs from the distribution tail at low cost. We adapt this insight for a different objective: rather than using verbalized probabilities for diversity, we use a discourse classifier as the acceptance criterion. For each misclassified message, we request $K{=}3$ diverse candidates in a single LLM call and accept the first one the classifier labels as the skeleton's intended move type. The choice of $K{=}3$ balances cost against resolution rate: empirically 82.8% of mismatches resolve within three candidates, with diminishing returns for $K{>}3$. The classifier serves as a correction tool: if the LLM's modal output falls near a decision boundary, a non-modal candidate often does not. To provide context for the correction, CRS makes use of the prior messages generated in the conversation history.

3.2.4 Phase 3: Evaluate objective

The corrected set of conversations is evaluated with cross-validated pMSE as described in Sec.3.2. The output from the evaluate objective step comprises: (1) for each conversation, the pMSE score and predicted label; and (2) feature weights for the fitted model $\mathcal{F}$ that indicate how the real and synthetic data are different in feature space.

3.2.5 Phase 4: Update generative parameters

The overall goal of this update phase is to perform an adjustment at iteration t:

$$\Omega^{(t+1)} \leftarrow \Omega(t) + \Delta(t)$$

based on a correction $\Delta(t)$ derived from the output of the evaluate objective step. This is a complex search space and there are many possible strategies for finding and performing the generative process update. As a starting point, we begin with simple, restricted search strategies to see how these succeed or fail, with a longer-term goal of further refinement in future work.

In this study, length features are currently the only features updated automatically in the update loop. The L1 coefficients $\mathbf{w}$ select which *length* knobs to update (moves where $|w_{f(m)}| > \tau$). We

expect to explore further updates to other components of Ω in future work. Full computation details are given in Appendix A: Algorithm 2 (single-pass synthesis, Phases 1–2.5) and Algorithm 3 (iterative length update, Phase 4).

4 Experiments

Datasets. **MMDB** (MathMentorDB) is a corpus of volunteer math tutoring sessions on Discord. We use 5,000 conversations for training (exemplar bank, skeleton sampling, knob fitting) and 2,000 held-out conversations for pMSE evaluation. All discourse labels (24-move taxonomy + resolution) are assigned by Gemini 2.5 Flash using few-shot prompting; the same model is used for generation. Using one model for both generation and discourse classification introduces a potential circularity, but pMSE is computed by a separate L1 logistic regression on 43-dimensional feature vectors, not by the discourse classifier directly. Systematic label biases affect real and synthetic corpora identically and cancel in the difference; the 8.20% real-vs-real null bounds the noise floor empirically.

QATD-2k (Eedi, 2024) (N=1,971 dialogues, 46K turns) is a large open-source 1:1 math tutoring dataset, anchored to Diagnostic Questions on the UK Eedi platform. The two corpora are structurally distinct: on the same features, the propensity classifier separates them at 77% pMSE. QATD-2k tutors scaffold heavily (15.1% vs. 5.6% in MMDB) and give direct answers rarely (16.4% vs. 50.2%); students answer in shorter bursts (mean 2.77 words vs. 8). We use 500 conversations for cross-corpus knob fitting and 200 held-out for pMSE evaluation.

Evaluation protocol. We used N=200 synthetic vs. N=200 held-out real conversations, $c = 0.5$. The "Combined-43" run uses all 43 features jointly; per-dimension scores use dimension subsets. Table 3 reports all results; subsections below address each research question in turn.

4.1 RQ1 — Prose Generation Fails, and Revision Cannot Fix It

Prose-only generation—a single whole-conversation prompt with no skeleton, length conditioning, or exemplar bank—scores a very poor pMSE of 96.96% using Combined-43 features on MMDB (Table 3): the propensity classifier can distinguish synthetic from real conversations with near-certainty. All four dimensions fail—Surface (61.26%), Content (94.24%), Dynamics (65.43%)—because a single prompt mediates all features simultaneously, with no mechanism to anchor any one of them independently.

FidelityScope reveals that the failure is not uniform, which matters for diagnosis. Consider the pre-CRS factored baseline (row C, +Exemplar bank): after adding length conditioning and an exemplar bank, Surface reaches near-null (2.74%) while Content remains at 28.56%. A scalar quality metric might report these conversations as realistic; FidelityScope identifies that the problem is entirely in Content—the generator over-scaffolds and under-corrects relative to real tutoring (Appendix D). Knowing which dimension is still off points directly to which component to fix next.

Iterative prose revision cannot close this Content gap. Because one prompt mediates all 43 features at once, adjusting scaffold phrasing shifts move frequencies, message lengths, and turn-taking patterns together. In the optimization loop, iteration 2 reduced Surface from 28% to 21% but simultaneously doubled Content from 14% to 32%, triggering a rollback—replicating the oscillation previously observed by Ion and Collins-Thompson (2026).

4.2 RQ2 — Factored Generation Achieves Near-Null Fidelity

Table 3 (MMDB rows A–D) shows the full ablation. Starting from the prose-only baseline (96.96%), the factored architecture with CRS reaches 10.54%—a 9.2× reduction, within 2.3 pp of the 8.20% null. In terms of the pMSE ratio with the null, the factored result (1.3× null) approaches the theoretical ideal of 1.0, versus 11.8× for prose-only generation.

Each step in the ablation targets a specific dimension. **A→B** (skeleton + length conditioning): Surface collapses from 27.32% to 3.66% with almost no change in Content (33.40%→30.30%). Conditioning each message on its lognormal length target closes the surface gap without affecting the discourse move distribution—exactly the independence the factored design intends. **B→C** (+Exemplar bank): Content improves modestly (30.30%→28.56%), but Dynamics increases slightly (3.79%→6.60%). Providing real example messages anchors phrasing, but LLM alignment still concentrates generation at modal phrasings that the classifier does not read as the intended move. **C→D** (+CRS): Content drops sharply from 28.56% to 9.84%. CRS replaces over-represented modal phrasings with alternatives

Table 3: Fidelity results across generation approaches and corpora. $\text{pMSE}_{\text{norm}}$ (Eq. 3): lower is better. N=200 vs. N=200 held-out real, c=0.5. Rows A–D add factored components in sequence (§3.2).

	Per dimension				
	Surf. (6)	Cont. (24)	Dyn. (10)	Out. (3)	Full 43
MMDB					
Null (real vs. real)	1.42	4.42	1.66	0.83	8.20
Prose baseline	61.26	94.24	65.43	13.74	96.96
Factored — pre-CRS					
A. Skeleton only	27.32	33.40	6.85	1.24	45.89
B. +Length conditioning	3.66	30.30	3.79	2.47	31.34
C. +Exemplar bank	2.74	28.56	6.60	1.41	29.73
Factored — post-CRS					
D. +CRS	3.94	**9.84**	5.17	**0.97**	**10.54**
QATD-2k (transfer: parameters re-fitted)					
Null (real vs. real)	1.64	7.30	3.78	1.31	12.54
Prose baseline	80.71	89.70	35.12	0.14	96.20
Factored — pre-CRS					
Factored, no CRS	9.34	41.11	7.73	6.65	40.53
Factored — post-CRS					
Factored +CRS	13.53	36.16	12.77	3.47	43.84
Factored +CRS +exemplar	12.18	35.04	13.96	4.09	45.48

from the LLM's output tail, directly breaking the alignment-classifier feedback loop. After CRS, the remaining gap (10.54% vs. 8.20% null) is concentrated in a small set of discourse move errors where the classifier has limited resolution; see RQ3.

4.3 RQ3 — Transfer to QATD-2k

We applied the same architecture to QATD-2k by re-fitting only the per-move length parameters, skeleton bank, and exemplar bank on QATD-2k training data. The algorithm itself was unchanged.

The architecture generalizes. The prose-only baseline on QATD-2k is 96.20%—nearly the same as on MMDB (96.96%)—showing that structural differences in the two corpora are also invisible to a generic prose prompt. The factored single-pass reduces this to 40.53%, a 2.4× improvement, similar to MMDB's pre-CRS trajectory (3.3× from prose to row C). The two corpora differ structurally (77% pMSE separating them on the same feature set), yet re-fitting only the numerical features is enough to give evidence that the skeleton-plus-fill structure generalizes without retraining.

CRS does not help on QATD-2k. Adding CRS regresses slightly (40.53%→45.48%), with only a small Content gain (41.11%→35.04%) offset by a Dynamics increase (7.73%→13.96%). The per-dimension breakdown explains why this differs from MMDB. On MMDB, CRS succeeds because the misclassified move pairs have clear classifier boundaries—sampling the LLM's output tail finds alternatives the classifier accepts. On QATD-2k, the largest Content gap is Guidance-Direct vs. Guidance-Scaffolding, two high-frequency moves (50.2% and 15.1% of tutor turns) that the classifier cannot reliably distinguish in a scaffold-heavy register. Within K=3 sampled candidates, 17.7% of Guidance-Direct mismatches found no alternative the classifier accepted. The ceiling is set by classifier boundary quality, not by the generator. Improving the discourse classifier at this specific move pair is the direct path to CRS working on QATD-2k.

5 Conclusion

We presented and analyzed FidelityScope, a reusable and extensible corpus generation diagnostic and optimization tool. The benchmark and generation architecture are designed to generalize: any conversational corpus with a discrete-move classifier and per-message statistical features could readily adapt all phases of our framework. Our choices for how each phase was implemented are flexible and could be replaced with more refined methods. Additional experimental extensions include computing bootstrap confidence intervals with larger sample sizes to quantify uncertainty across ablation conditions; cross-LLM evaluation to test whether the architecture's gains are LLM-agnostic; improving the discourse classifier for Guidance-Direct / Scaffolding in scaffold-heavy corpora, which is the direct path to CRS effectiveness on QATD-2k; and closing the remaining manual steps in the optimization architecture by extending Phase 4 to skeleton-sampling weights (Dynamics, Outcome) and automated exemplar selection (Content).

Limitations

Single domain. All experiments are on math tutoring from two corpora. The architecture is designed to generalize to any classifier-in-the-loop generation setting, but confirming this with domain validation beyond math tutoring is open work.

Classifier-quality dependence. The residual 10.54% Combined-43 is partly set by the 13.6% post-CRS mismatch rate and the classifier's 62.9% aggregate move accuracy. Improvements to the discourse classifier would lower the Content pMSE floor independently of generation quality. Additionally, the CRS step selects for phrasings Gemini 2.5 Flash classifies unambiguously, which may inflate measured fidelity under Gemini evaluation relative to an independent rater: CRS corrects typicality-classifier coupling for this classifier, not for discourse structure in general.

CRS effectiveness is bounded by classifier boundary quality. On QATD-2k, CRS with the exemplar bank (all 24 moves, 10 exemplars each) causes net regression: Combined-43 moves from 40.53% (single-pass) to 45.48% (CRS + bank), with marginal Content improvement (41.11% $\rightarrow$ 35.04%) offset by Dynamics regression (7.73% $\rightarrow$ 13.96%). This diverges from the MMDB result (28.56% $\rightarrow$ 9.84% Content) because the failure mode differs. In MMDB, the dominant mismatches are typicality-bias cases: the LLM's modal output lands near a classifier decision boundary, and the correct phrasing exists in the distribution tail (82.8% of mismatches resolve within $K{=}3$). In QATD-2k, the dominant pair is Guidance-Direct / Guidance-Scaffolding—the two highest-frequency tutor moves in a scaffold-heavy corpus—where the classifier itself has intrinsic boundary ambiguity: 17.7% of Guidance-Direct mismatches were unresolvable within $K{=}3$ candidates. CRS is a targeted fix for typicality-classifier coupling, not for classifier boundary quality. When the two are conflated, CRS is incorrectly blamed for a problem it was not designed to solve. The direct fix for QATD-2k is upstream: improving the discourse classifier's Guidance-Direct / Guidance-Scaffolding boundary, after which CRS should recover similar resolution rates to MMDB.

Partial automation of the feedback loop. Phase 4 automates the feedback loop for Surface features only: the L1 coefficients select which per-move length targets to update, and the update is a directly observable ratio requiring no gradient through the LLM or classifier. Dynamics and Outcome signals from the L1 coefficients are currently informative but not automatically actionable. Extending to these dimensions is mechanically feasible: skeleton-sampling weights can be adjusted by computing each skeleton's feature values and reweighting toward the target distribution, but the loop is not yet implemented. Exemplar bank updates are partially automatable (add exemplars for moves with low classifier acceptance rates) but hit a hard floor for move pairs with blurry classifier boundaries: Student-Confirm has a 29.5% classifier accuracy (Table 5), reflecting a classifier floor for short affirmative messages that additional exemplars cannot fully overcome. These residuals are a classifier quality floor, not a generation ceiling.

Bootstrap CIs. All reported numbers are $N{=}200$ point estimates without bootstrap confidence intervals. Wider evaluation ($N{=}600$ with bootstrap CIs) would quantify uncertainty across ablation conditions and is left for future work.

LLM-classified labels and classifier drift. Both training and held-out conversations use Gemini 2.5 Flash labels; Section 4 addresses the circularity concern directly. An additional issue is longitudinal comparability: if the classifier model is updated between experiments, systematic label changes would affect pMSE scores across time. Pinning the classifier version and re-running the null baseline when the model changes is the practical mitigation.

References

Eedi. 2024. QATD-2k: A Dataset of Question-Answering Tutoring Dialogues. Publicly available dataset, CC-BY-NC-SA.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Proceedings of ICML 2024*.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *Proceedings of ICLR 2024*.

Michael Ion and Kevyn Collins-Thompson. 2026. Measuring Simulation Fidelity via Statistical Detectability: A Diagnostic Framework for AI-Generated Tutor-

ing Conversations. In *Proceedings of the Thirteenth ACM Conference on Learning @ Scale (L@S '26)*, Seoul, South Korea. ACM.

Omar Khattab and 1 others. 2024. Compiling declarative language model calls into self-improving pipelines. In *Proceedings of ICLR 2024*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. *Proceedings of EMNLP 2023*.

Xinyi Lu and Xu Wang. 2024. Generative students: Using LLM-simulated student profiles to support question item evaluation. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, pages 1–12. ACM.

Julia M Markel, Steven G Opferman, James A Landay, and Chris Piech. 2023. Gpteach: Interactive TA training with GPT-based students. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, pages 226–236, New York, NY, USA. ACM.

Luis Marquez-Carpintero, Alberto Lopez-Sellers, and Miguel Cazorla. 2025. Simulating students with large language models: A review. *arXiv preprint arXiv:2511.06078*.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog. In *Proceedings of the 58th Annual Meeting of the ACL*, pages 681–707. ACL.

Biswesh Mohapatra, Gaurav Pandey, Danish Sui, and Tanya Umapathi. 2021. Simulated chats for building dialog systems: Learning to generate conversations from instructions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203. ACL.

Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. 2018. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):663–688.

Heydar Soudani, Roxana Petcu, Evangelos Kanoulas, and Faegheh Hasibi. 2024. A survey on recent advances in conversational data generation. *arXiv preprint arXiv:2405.13003*.

Mi-Ja Woo, Jerome P Reiter, Anna Oganian, and Alan F Karr. 2009. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1):111–124.

Cilin Yan, Jingyun Wang, Lin Zhang, Ruihui Zhao, Xiaopu Wu, Kai Xiong, Qingsong Liu, Guoliang Kang, and Yangyang Kang. 2025. Efficient and Accurate Prompt Optimization: the Benefit of Memory in Exemplar-Guided Reflection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, volume Volume 1: Long Papers, pages 753–779.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *Proceedings of ICLR 2024*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTscore: Evaluating text generation with BERT. *Proceedings of ICLR 2020*.

Zhaolin Zhang, Zhenfang Chen, Tianmin Shu, Kent Harber, Utkarsh Bhatt, and Chuang Gan. 2025. Verbalized Sampling: How to Mitigate Mode Collapse and Unlock LLM Diversity. *Preprint*, arXiv:2510.01171.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *Proceedings of ICLR 2023*.

A Full Algorithms

The factored architecture separates into two algorithms. Algorithm 2 is the core synthesis procedure (Phases 1–2.5 plus evaluation); it is complete as a single-pass system and produces the overall 10.54% pMSE result (Table 3) on the MMDB dataset. Algorithm 3 is an optional iterative controller that loops over Algorithm 2, updating only the per-move length knobs via proportional feedback; it requires no gradient through the LLM or classifier. The split makes explicit that Phase 4 is a feedback loop *on top of* an otherwise complete synthesis procedure, not part of generation itself.

B 24-Move Discourse Taxonomy

The complete move category taxonomy used for the MMDB dataset is given in Table 4.

C Per-Move Classifier Accuracy

Table 5 shows the gap between skeleton intent and classifier label for the N=200 single-pass run before CRS. Moves with distinctive surface forms (Platform-Command: 89.7%, Confirm-Positive: 84.5%) have high accuracy; moves that lack distinctive markers overlap with adjacent move types.

D Benchmark Diagnostic Example

Table 6 shows the top FidelityScope L1 coefficients from the single-pass factored baseline (Condition C, 29.73% Combined-43) evaluated against MMDB held-out real conversations (N=200 each). Positive coefficients indicate features the synthetic corpus over-produces relative to real; negative indicate under-production.

Algorithm 2 Single-Pass Factored Synthesis (Phases 1–2–2.5–3)

Require: Real corpus R_{train}, $R_{\text{held-out}}$; batch size N; 24-move taxonomy $\mathcal{M}$

Ensure: Minimize $\text{pMSE}(X_{\text{real}},\ X_{\text{synth}};\ \Phi_{43})$ (Eq. 4) via decomposition (Figure 2): **(C1)** skeleton (role, m) sampled from R_{train} pins Content by construction; **(C2)** $\ell \sim \text{LogN}(L_m, \sigma_m)$ gives Surface an independent controller per move; **(C3)** classifier label must match skeleton intent (CRS: classifier oracle, not diversity sampler).

Phase 1 — Fit knobs from real data *no LLM calls; statistics only*
for each move $m \in \mathcal{M}$ **do**
  $L_m \leftarrow$ mean message length for move m in R_{train}
  $\sigma_m \leftarrow \sqrt{\log(1 + (\text{std}_m/L_m)^2)}$ {MoM lognormal; $\mathbb{E}[\ell]{=}L_m$}
  $E_m \leftarrow$ {real messages of move m from R_{train}} {exemplar bank}
end for
$\mathcal{S} \leftarrow$ {real (role, m) sequences from R_{train}} {skeleton bank}
Phase 2 — Factored generation *C1 + C2 enforced here*
for $i = 1$ **to** N **do**
  skeleton$_i \leftarrow$ sample from $\mathcal{S}$ {**C1**: Content move rates pinned; removed from optimization}
  for each slot $(\text{role},\ m) \in$ skeleton$_i$ **do**
    $\ell \sim \text{Lognormal}(\log L_m - \frac{1}{2}\sigma_m^2,\ \sigma_m)$ {**C2**: Surface; independent of move rates}
    text $\leftarrow \text{LLM}(m,\ \ell,\ E_m,\ \text{neg-exemplar},\ \text{history})$ {LLM as renderer, not optimizer}
  end for
end for
Phase 2.5 — Classifier-Guided Rejection Sampling (CRS) *C3 enforced here*
for each message where clf-label $\neq$ skeleton intent **do**
  candidates $\leftarrow$ CRS($K{=}3$): LLM call requesting diverse candidates {samples output tail via classifier oracle, not verbalized probabilities}
  accept first candidate clf labels as skeleton intent {**C3**: label forced to match skeleton}
  else keep highest-probability candidate {28.56%→9.84% Content pMSE}
end for
Phase 3 — Evaluate
Re-classify all messages → extract Φ_{43} features → F_{synth}
$\text{pMSE},\ \mathbf{w} \leftarrow$ L1 logistic (5-fold CV) on $F_{R_{\text{held-out}}} \cup F_{\text{synth}}$ (Eq. 3)
return $X_{\text{synth}},\quad \text{pMSE},\quad \mathbf{w}$

Algorithm 3 Iterative Length Update — Proportional Feedback Control (Phase 4)

Require: Current knobs $\{L_m^{(0)}\}_{m\in\mathcal{M}}$ from Algorithm 2; L1 coefficients $\mathbf{w}$; selector cutoff τ; ratio clamp $[r_{\text{lo}}, r_{\text{hi}}] = [0.5, 3.0]$; max iterations T

Ensure: Surface control only (C2); no gradient through LLM or classifier. Dynamics and Outcome have no controller in this loop (see Figure 2).

$\text{pMSE}_{\text{best}} \leftarrow \text{pMSE}$ from Algorithm 2
for $t = 1$ **to** T **do**
  // select discriminating length knobs via L1 coefficients
  for each move m with $|w_{f(m)}| > \tau$ **do**
    $r \leftarrow \text{clip}(L_m^{\text{target}}/L_m^{\text{synth,(t-1)}},\ r_{\text{lo}},\ r_{\text{hi}})$ {observable ratio; no LLM/clf gradient}
    $L_m^{(t)} \leftarrow L_m^{(t-1)} \times r$ {proportional correction, η=1.0}
  end for
  Re-run Algorithm 2 with $\{L_m^{(t)}\} \rightarrow \text{pMSE}_t$, $\mathbf{w}$
  if $\text{pMSE}_t < \text{pMSE}_{\text{best}}$ **then**
    Accept: $\text{pMSE}_{\text{best}} \leftarrow \text{pMSE}_t$
  else
    Rollback: $L_m^{(t)} \leftarrow L_m^{(t-1)}$ **break**
  end if
end for
return $\{L_m^{\text{best}}\}$, $\text{pMSE}_{\text{best}}$

Cat.	Move	Example
Tutor-Academic	Guidance-Direct	"you have an error in your derivative"
	Guidance-Scaffolding	"what have you tried?"
	Tutor-Question	"which problem number?"
	Correction	"actually that should be positive"
	Confirm-Positive	"yes", "correct"
	Confirm-Negative	"no", "not quite"
	Knowledge-Check	"do you know the Pythagorean identity?"
Student-Academic	Knowledge-Gap	"idk how to do this"
	Knowledge-Recall	"yes I know the chain rule"
	Inference-Attempt	"so is it x=3?"
	Inference-Understanding	"oh ok that makes sense"
	Breakthrough	"OHHH I SEE IT NOW"
	Explain-Problem	"the question says find the derivative…"
	Explain-Reasoning	"I tried substituting but got stuck"
	Student-Question	"what do you mean by that?"
	Student-Confirm	"yes", "right?"
Socio-Em.	Encouragement	"you're close, keep going!"
	Frustration	"ugh", "I'm so lost"
	Empathy-Rapport	"no worries take your time"
	Confidence-Express	"I got this!"
Non-Acad.	Greeting-Closing	"hi", "thanks for your help"
	Platform-Command	".close", "@Helpers"
	Small-Talk	"lol", "brb"
	Acknowledgment	"ok", "gotcha"

Table 4: 24-move discourse taxonomy with example utterances.

Table 5: Per-move accuracy (skeleton intent vs. classifier label) and top confusion pairs, single-pass, N=200.

Move	Accuracy	Top confusion
Platform-Command	89.7%	—
Confirm-Positive	84.5%	—
Confirm-Negative	82.0%	—
Inference-Attempt	81.4%	—
Greeting-Closing	79.0%	—
Knowledge-Recall	78.9%	—
Knowledge-Check	77.6%	—
Frustration	76.2%	—
Guidance-Direct	71.6%	→ Confirm-Positive
Guidance-Scaffolding	67.8%	→ Tutor-Question
Encouragement	62.5%	—
Explain-Problem	58.3%	—
Knowledge-Gap	53.9%	→ Frustration
Inference-Understanding	52.1%	—
Explain-Reasoning	50.3%	→ Inference-Attempt
Tutor-Question	48.8%	→ Guidance-Scaffolding
Acknowledgment	47.3%	—
Confidence-Express	46.2%	← Breakthrough
Student-Question	43.4%	→ Inference-Attempt
Small-Talk	33.2%	→ Greeting-Closing
Breakthrough	33.7%	→ Confidence-Express
Correction	29.6%	→ Guidance-Direct
Student-Confirm	29.5%	—
Empathy-Rapport	25.0%	—
Aggregate	**62.9%**	

Table 6: Top FidelityScope L1 coefficients (single-pass baseline vs. MMDB real, N=200 each). Positive = synthetic over-produces; negative = under-produces.

Feature	Dim.	Coef.
pct_socio_emotional	Content	$+0.61$
scaffolding_rate	Content	$+0.43$
correction_rate	Content	-0.52
explain_problem_rate	Content	-0.35
n_confusion_episodes	Dynamics	-0.29
direct_rate	Content	-0.22