CAST: CONDITIONAL GENERATION AND ADVERSARIAL SELECTION FOR TABULAR DATA SYNTHESIS WITH LONG TEXT COLUMNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthesizing tabular data with long text columns requires preserving structured attributes, text semantics, and the predictive relationships between them. Existing methods improve individual records through generator fine-tuning or post-generation filtering, but high record-level quality *does not ensure that the resulting table preserves text–attribute relationships at the dataset level*. We propose **CAST**, a conditional generation and adversarial selection framework that keeps the text generator frozen and combines candidate construction under structural constraints with dataset-level selection. Structure-Aware Semantic Generation (SSG) builds multiple text candidates from a shared semantic plan and distinct narrative roles. Conditional Adversarial Selection (CAS) reselects candidates by matching task features to those of real references, with feedback from the current synthetic table and a fixed domain discriminator. Reliability-Guided Conservative Selection (RCS) gates and reweights task feedback using validation performance, then mixes selected candidates with preset-role candidates to reduce reliance on proxy scores. Across three tabular datasets with long text columns, CAST achieves higher downstream utility than the compared synthesis methods, with gains across different generator backbones and modeling strategies.

1 INTRODUCTION

Real-world tabular data often combine numerical, categorical, and free-form text fields within the same record. Text fields capture context, observations, and evaluative evidence that structured attributes alone cannot fully express (Shi et al., 2021; Zhang et al., 2026b). Such text-rich tables allow models to learn cross-field relationships between structured attributes and text. The synthesis process should therefore preserve these relationships when generating individual fields. However, conventional tabular generators are designed mainly for numerical and categorical variables and do not directly model long-form text or its semantic dependencies on structured attributes (Kotelnikov et al., 2023; Zhang et al., 2024).

Language models provide a natural way to jointly generate structured attributes and text, and existing work broadly follows two directions. *Parameter-adaptation methods* update model parameters to learn field distributions and cross-field relationships in the target table. GReaT and TapTap, for example, fine-tune language models for tabular generation (Borisov et al., 2023; Zhang et al., 2023). For tables with long-text columns, AFT-Tab adversarially fine-tunes the text generator with discriminator feedback and updates the non-text generator through sample replay (Zhang et al., 2026b). In contrast, *frozen-model methods* keep the language model fixed and improve generation through context, feedback, or sample selection. CLLM combines few-shot generation with learning-dynamics-based filtering (Seedat et al., 2024), MALLM-GAN uses multi-agent collaboration and discriminator feedback to refine generation (Ling et al., 2026), and SelPE progressively selects schema-constrained structured-text samples after generation (Zhu et al., 2026). A broader discussion of related tabular synthesis methods is provided in Appendix A.

The two directions improve synthesis in different ways. *Parameter-adaptation methods* can absorb statistical and semantic patterns from the target data, but generation is still typically carried

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

out record by record. They also require model updates for each target dataset, adding training cost and limiting the use of models available only through inference APIs. *Frozen-model methods* avoid this cost and can improve outputs through demonstrations, feedback, or sample selection, but these mechanisms usually act on prompt construction, individual generations, or sample retention. However, improving records individually does not guarantee that the resulting synthetic table preserves the relationships present in real data. A collection of plausible records can still overrepresent particular text–attribute patterns, for example texts that make a target attribute unusually easy to infer. Each record may remain internally consistent, while the dataset as a whole presents a distorted predictive structure. Such distortions matter when synthetic data are used for model training, because they change the patterns available to downstream models. Faithful synthesis therefore requires attention to both record-level quality and dataset-level text–attribute relationships.

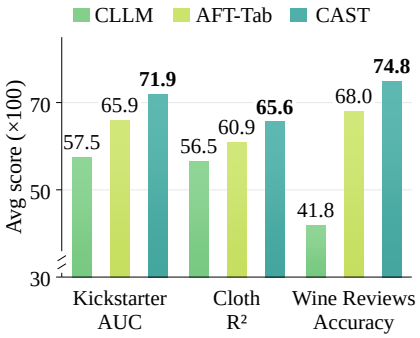


Figure 1: Average TSTR utility on real test sets with Qwen2.5-7B. Bars average All-text, Fuse-late, and Fuse-early scores.

To address this issue, we propose **CAST**, which organizes tabular synthesis with long-text columns as conditional generation under structural constraints followed by adversarial selection. Structure-Aware Semantic Generation (SSG) first synthesizes structured records and then uses a shared semantic plan and distinct narrative roles to construct multiple text candidates for each record under common content constraints. Conditional Adversarial Selection (CAS) assigns synthetic records to real references spanning different predictive-feature levels within each condition group and reselects candidates using conditional matching and domain-discrimination feedback. Before each reselection round, synthetic-data text predictors are refit on the current synthetic table, so their feedback reflects the associations induced by the current selection. Reliability-Guided Conservative Selection (RCS) gates and reweights feedback according to performance on real validation data and mixes the reselected candidates with preset-role candidates at the row level, reducing dependence on proxy feedback. The text generator remains frozen throughout.

We follow the multimodal text–tabular benchmark (Shi et al., 2021) and the experimental setup of AFT-Tab (Zhang et al., 2026b), evaluating CAST on Kickstarter (Lathwal), Cloth (Agarap, 2018), and Wine Reviews (Zynicide, 2017) under the Train-on-Synthetic, Test-on-Real (TSTR) protocol (Esteban et al., 2017). Downstream predictors are trained on synthetic data and evaluated on real test data. In the main experiments reported in Table 1, CAST achieves the highest synthetic-data utility across both generator backbones, all three tasks, and all three modeling strategies. As shown in Figure 1, with Qwen2.5-7B (Qwen et al., 2024), CAST achieves the highest average utility on all three datasets among the compared methods; on Wine Reviews, average Accuracy increases from 68.0% for AFT-Tab to 74.8%. Additional analyses across generator backbones, fixed candidate pools, reselection stages, and repeated downstream training further show that both candidate construction and candidate composition affect the training value of synthetic data.

Our main contributions are as follows:

1. Conditional generation with dataset-aware selection. We introduce CAST, which keeps the text generator frozen and builds multiple candidates for each structured record from a shared semantic plan and narrative roles, turning synthesis into candidate construction and composition under fixed structural conditions.

2. Selection guided by conditional associations. CAS assigns differentiated real references within condition groups and updates predictive feedback from the current synthetic table, with domain feedback guiding reselection. RCS gates and reweights this feedback using validation performance and applies row-level mixing to reduce reliance on proxy scores.

3. A systematic study of candidate construction and selection. Experiments across three datasets, generator backbones, and modeling strategies evaluate downstream utility. Controlled studies of candidate construction, fixed-pool selection, feedback updates, and repeated downstream training show how construction and composition jointly affect utility.

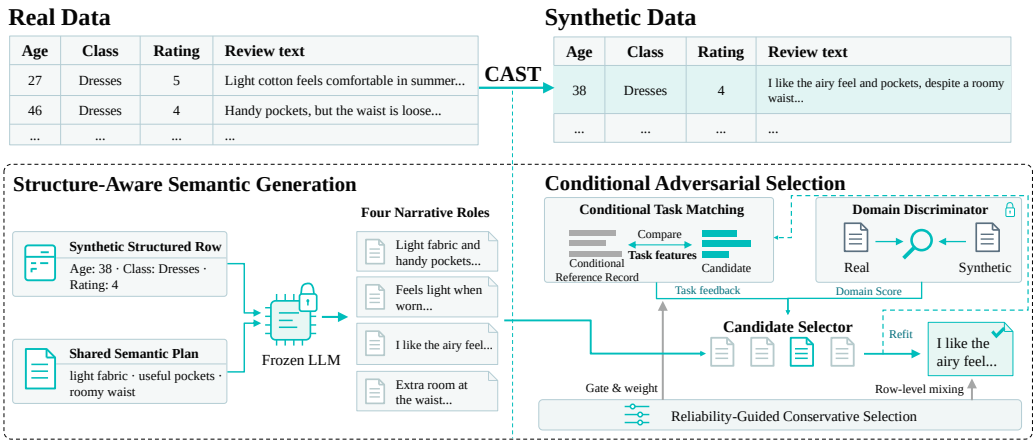


Figure 2: Overview of CAST. SSG constructs multi-role candidates under a shared semantic plan; CAS combines conditional task matching with adversarial domain feedback for candidate selection; RCS gates and weights task feedback and applies row-level mixing.

2 Preliminaries

2.1 Problem Setup

We study tabular data synthesis with one or more long-text columns. Each record consists of structured attributes $\mathbf{s}_i$, including numerical, categorical, or ordinal variables, and text fields $\mathbf{t}_i$. Given a real dataset $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{t}_i)\}$, the goal is to generate synthetic records with the same schema such that the text remains semantically compatible with the structured attributes at the record level, while the synthetic dataset preserves variation in real text–attribute associations under similar structured conditions. These properties are intended to support downstream modeling on real data. The primary target and any configured auxiliary attributes define the associations considered in this work.

2.2 Candidate Selection Formulation

To describe the selection process used later, let $\tilde{\mathbf{s}}_i$ denote the synthetic structured attributes of record i, and let $\mathbf{t}_{ij}$ denote its j-th text candidate, containing all text fields to be generated for that record. We use $\mathcal{I}$ for the set of valid record indices and $\mathcal{J}_i$ for the valid candidate indices of record i. Let j_i denote the selected candidate index and $\mathbf{j} = (j_i)_{i \in \mathcal{I}}$ the complete set of selections. The resulting synthetic table is

$$\widetilde{\mathcal{D}}_{\mathbf{j}} = \{(\tilde{\mathbf{s}}_i, \mathbf{t}_{i,j_i}) : i \in \mathcal{I}\}, \qquad j_i \in \mathcal{J}_i. \tag{1}$$

With the structured records and candidate pools fixed, the selection vector $\mathbf{j}$ directly determines the textual composition of the synthetic table. Section 3 describes how these candidates are constructed and how the selections are updated using conditional references and feedback from the current synthetic table.

3 The CAST Framework

3.1 Overview

CAST organizes tabular synthesis with long-text columns as candidate construction and selection under fixed structured conditions. As shown in Figure 2, Structure-Aware Semantic Generation (SSG) first synthesizes structured rows and constructs multiple text candidates under a shared semantic plan. Conditional Adversarial Selection (CAS) then reselects candidates through *Conditional Task Matching* and adversarial feedback from a *Domain Discriminator*. Reliability-Guided Conservative Selection (RCS) gates and weights task feedback according to validation performance and

forms the final output through row-level mixing. During selection, the structured rows and candidate pools remain fixed, and the LLM used for text generation is frozen throughout.

3.2 Structure-Aware Semantic Generation (SSG)

Structured Row Synthesis. In our implementation, the structured-synthesis stage uses a Gaussian-Copula-based generator (Patki et al., 2016) to synthesize non-text attributes, including the primary target. Subsequent candidate construction and selection operate on the resulting fixed structured rows $\tilde{\mathbf{s}}_i$ after validity checks (Appendix B.2).

Semantic Planning and Candidate Generation. For each synthetic structured row, SSG retrieves related real records and uses maximal marginal relevance (MMR) to balance structural matching and reference diversity (Carbonell & Goldstein, 1998). The retrieved records are divided into content references $\mathcal{R}_i^{\text{content}}$ and style references $\mathcal{R}_i^{\text{style}}$. Content references support the semantic content of the row, while style references provide domain-specific expression patterns.

From the structured condition and content references, the frozen LLM extracts a shared semantic plan p_i. It generates four candidates under the roles *Balanced Evidence*, *Experience First*, *Decision First*, and *Implicit Reflection*. Roles vary information order and assessment explicitness under a common content constraint: prompts instruct each role to retain the plan's supported evidence, assessment, and outcome and avoid unsupported facts. Let π_j denote candidate j's role:

$$p_i = G_\theta^{\text{plan}}\left(\tilde{\mathbf{s}}_i, \mathcal{R}_i^{\text{content}}\right), \qquad \mathbf{t}_{ij} \sim G_\theta^{\text{text}}\left(\cdot \mid \tilde{\mathbf{s}}_i, p_i, \mathcal{R}_i^{\text{style}}, \pi_j\right). \tag{2}$$

Here, G_θ^{plan} and G_θ^{text} are two calls to the same frozen LLM. Once generated, the candidate pool remains fixed; CAS and RCS only select among these candidates and do not regenerate or rewrite text. Reference retrieval, generation constraints, and validity checks are described in Appendix B.2.

3.3 Conditional Adversarial Selection (CAS)

CAS uses variation in attribute-related cues among real records under similar structured conditions to guide candidate selection.

Conditional Task Matching. Each synthetic row receives a differentiated *Conditional Reference Record*. Real validation records are grouped by the target and, when supported by coverage, one auxiliary categorical field. Within each group, references are ordered by a primary-target predictive feature, and synthetic rows are assigned deterministically to different positions. The reference $\text{ref}(i)$ remains fixed during reselection, allowing a condition group to cover different predictive-feature levels. Appendix B.3 defines condition construction and reference assignment.

Lightweight predictors provide *Task Features*: categorical attributes use own-label probability margins, predictive entropy, and predicted classes; numerical attributes use signed residuals; ordinal attributes use expected-level residuals, threshold uncertainty, and predicted levels. Text-view feedback covers the primary target and configured auxiliary attributes, with an additional structured–text view for the primary target.

Real-data predictors remain fixed, while synthetic-data text predictors are refit on the current synthetic table before each round. A candidate and its fixed reference are evaluated by the same predictor. Their feature differences define the conditional cost $C_{ij,h}^{(r)}$ for feedback term h at round r, with continuous differences normalized by scales estimated on real validation data. Appendix B.3 gives the feature and cost definitions.

Adversarial Candidate Selection. The *Domain Discriminator* is a lightweight TF-IDF logistic classifier trained on balanced real and candidate texts. Its *Domain Score* measures resemblance to the real-data domain; both the discriminator and its scores remain fixed during reselection.

Initialization matches primary-target task features within target-based reference groups. Its term C_{ij}^{init} is the normalized within-row rank of the raw matching cost; C_{ij}^{dom} is one minus the normalized within-row Domain Score rank. Because C_{ij}^{dom} uses only within-row score ranks, absolute

probability calibration is not required for this term. Lower costs favor closer initial task matching or more real-like text. The Candidate Selector combines these fixed terms with conditional task feedback:

$$j_i^{(r+1)} = \arg\min_{j \in \mathcal{J}_i} \left[\lambda_{\text{init}} C_{ij}^{\text{init}} + \lambda_{\text{dom}} C_{ij}^{\text{dom}} + \sum_h w_h^{(r)} C_{ij,h}^{(r)} \right]. \tag{3}$$

Here, λ_{init} and λ_{dom} weight the fixed costs, and RCS determines $w_h^{(r)}$. Starting from $j_i^{(0)}$, CAST performs at most two rounds. Each round updates task features and costs after refitting, then applies Equation (3) row by row. Selections are indirectly coupled through the shared predictor fitted on the selected table; structured rows and candidate pools remain fixed.

Here, *adversarial* means fixed-discriminator feedback guiding discrete selection. CAST neither alternates generator–discriminator training nor optimizes a global min–max objective.

3.4 Reliability-Guided Conservative Selection (RCS)

RCS evaluates feedback on real validation data to *gate and weight* its contribution. Effective reliability $\eta_h^{(r)}$ is zero when term h fails gating; otherwise, it reflects normalized improvement over a task-specific baseline. For attribute a, RCS allocates a fixed text-feedback budget W_a among its terms $\mathcal{H}_a$:

$$w_h^{(r)} = \begin{cases} W_a \dfrac{\eta_h^{(r)}}{\sum_{h' \in \mathcal{H}_a} \eta_{h'}^{(r)}}, & \text{if } \sum_{h' \in \mathcal{H}_a} \eta_{h'}^{(r)} > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad h \in \mathcal{H}_a. \tag{4}$$

This is a *gate-then-redistribute* design: validation determines which terms are trusted, and relative reliability distributes W_a among valid terms. A sole valid term receives all of W_a; reliability does not multiply the total budget. With no valid term, the budget is zero. Reliability is updated after each synthetic-data predictor refit, and reselection stops if all primary-target text feedback is invalid. Appendix B.4 gives the gating rules.

RCS then applies *row-level mixing* to reduce reliance on proxy-driven selection. The preset *Experience First* role begins with concrete observations and delays the overall assessment, providing a fixed fallback intended to reduce overt label cues. After R completed rounds, let $j_i^{(R)}$ and j_i^{preset} denote the final CAS and preset-role selections. The output index is

$$j_i^{\text{out}} = \begin{cases} j_i^{(R)}, & z_i < \rho, \\ j_i^{\text{preset}}, & z_i \geq \rho, \end{cases} \tag{5}$$

Here, $z_i \in [0, 1)$ comes from a stable row-identifier hash and fixed seed; ρ is a fixed threshold, not a per-row reliability estimate. If the preset role is unavailable, $j_i^{(0)}$ is the fallback. Substituting j_i^{out} into Equation (1) yields the output table.

4 Experiments

We organize the experiments around three questions. **RQ1: Downstream Utility** asks whether CAST produces useful synthetic training data across tasks, input strategies, and generator backbones (Sections 4.2–4.3). **RQ2: Distributional Quality** examines how structured and textual distributional properties relate to downstream utility through aggregate metrics, an alternative structured-generator control, and joint-representation visualization (Sections 4.4–4.5). **RQ3: Component Contributions** studies how candidate construction, adversarial selection, and reliability control contribute under controlled structured rows and candidate pools (Section 4.6).

4.1 EXPERIMENTAL SETUP

Data and evaluation. Following AFT-Tab (Zhang et al., 2026b) and the multimodal text–tabular benchmark (Shi et al., 2021), we evaluate campaign-success prediction on Kickstarter (AUC), rating regression on Cloth (R^2), and wine-variety classification on Wine Reviews (Accuracy). Downstream utility follows Train-on-Synthetic, Test-on-Real (TSTR) (Esteban et al., 2017), where predictors are trained on synthetic data and evaluated on real test data. We report three input strategies in A/L/E order (Shi et al., 2021): All-text serializes structured attributes and models them jointly with the original text, Fuse-late encodes the two modalities separately before combining their representations, and Fuse-early maps structured attributes into the text embedding space for joint modeling. Distributional metrics are introduced in Section 4.4; dataset splits, fields, and full evaluation details are provided in Appendix C.

Baselines and generators. We compare against CLLM (Seedat et al., 2024), TabDDPM+LLM (Kotelnikov et al., 2023; Zhang et al., 2026b), MALLM-GAN (Ling et al., 2026), SelPE (Zhu et al., 2026) with $\varepsilon = \infty$, and AFT-Tab (Zhang et al., 2026b). The main experiments use Qwen2.5-7B (Qwen et al., 2024) and Qwen3.8-27B (Qwen Team, 2026); on Cloth, we additionally evaluate DeepSeek-V4-Flash (DeepSeek-AI et al., 2026), Claude Opus 5 (Anthropic), and GPT-5.6 Sol (OpenAI). AFT-Tab is evaluated only with Qwen2.5-7B due to its fine-tuning cost, whereas CAST keeps the text generator frozen throughout.

4.2 DOWNSTREAM UTILITY

CAST improves downstream utility consistently across tasks and input organizations. Table 1 shows the strongest synthetic-data utility under both main backbones across all tasks and A/L/E strategies. CAST improves on AFT-Tab, the strongest Qwen2.5-7B competitor overall, under every input organization. The same ordering holds with Qwen3.8-27B despite changes in the strongest baseline across datasets. These tasks span classification and regression, while A/L/E expose the records to different downstream representations. The gains therefore reflect training value across several uses of the synthetic table, extending beyond any one input architecture.

Table 1: TSTR utility ($\times 100$). Cells report A/L/E = All-text/Fuse-late/Fuse-early; Kickstarter, Cloth, and Wine Reviews use AUC, R^2, and Accuracy, respectively. Bold marks the best synthetic result per generator and task; † denotes quoted results. Negative R^2 means worse than constant mean prediction on the real test set.

Generator	Method	Kickstarter A / L / E	Cloth A / L / E	Wine Reviews A / L / E
—	Original training data†	80.2 / 77.1 / 77.7	74.8 / 74.3 / 74.6	82.4 / 82.1 / 82.2
Qwen2.5-7B	CLLM†	57.0 / 58.5 / 56.9	54.4 / 55.3 / 59.9	40.9 / 42.3 / 42.3
	TabDDPM+LLM†	59.1 / 56.9 / 56.7	18.3 / 16.0 / 5.6	24.1 / 26.4 / 24.0
	MALLM-GAN	56.1 / 56.9 / 55.2	49.4 / 55.8 / 45.0	42.8 / 43.1 / 42.9
	SelPE ($\varepsilon = \infty$)	48.4 / 50.2 / 48.5	-21.9 / 15.1 / -19.3	28.7 / 27.8 / 28.4
	AFT-Tab†	66.3 / 64.8 / 66.5	61.0 / 58.9 / 62.8	67.4 / 68.6 / 68.0
	CAST	**72.9 / 71.7 / 71.0**	**63.5 / 64.4 / 69.0**	**74.7 / 75.6 / 74.0**
Qwen3.8-27B	CLLM	59.1 / 57.6 / 58.2	62.6 / 60.8 / 64.0	49.1 / 46.7 / 42.8
	TabDDPM+LLM	60.7 / 58.6 / 58.0	16.9 / 34.4 / 15.4	22.9 / 21.6 / 25.8
	MALLM-GAN	53.4 / 54.4 / 54.3	52.0 / 48.2 / 46.7	63.0 / 63.4 / 63.4
	SelPE ($\varepsilon = \infty$)	52.3 / 53.6 / 49.3	35.2 / 35.5 / 32.1	31.7 / 32.7 / 28.3
	CAST	**75.0 / 74.3 / 73.2**	**69.6 / 65.2 / 68.9**	**77.0 / 77.4 / 77.2**

Generator scale alone does not determine synthetic training utility. Qwen2.5-7B CAST exceeds larger-generator baselines across all three datasets. This is consistent with its frozen-generator design: training value depends on which condition-compatible candidates are available and how selection assembles them across rows. Its remaining gap to original-data training also shows that synthetic records do not recover all real-data signal.

4.3 Utility Across Generator Backbones

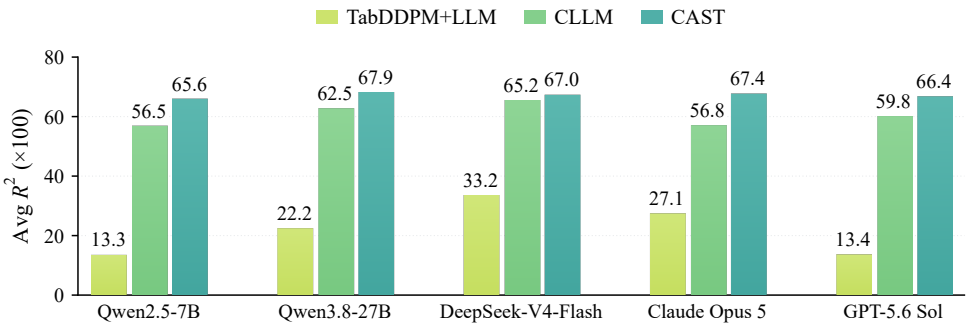


Figure 3: Average TSTR utility across generator backbones on Cloth, averaged over All-text, Fuse-late, and Fuse-early ($R^2 \times 100$).

CAST is comparatively stable across the evaluated generator backbones. Figure 3 shows a relatively narrow utility range across five generators, whereas CLLM and especially TabDDPM+LLM are more sensitive to backbone choice. Both baselines peak with DeepSeek-V4-Flash, yet CAST retains its advantage. This stability as the text generator changes extends beyond a gain obtained with one especially favorable backbone. It is consistent with CAST's division of labor: the backbone supplies candidate texts, while conditional references, task feedback, and Domain Scores guide their composition into the final table. Independent complete-synthesis realizations in Appendix D.5 separately check variation across synthesis realizations on Cloth.

4.4 Synthetic Data Quality

Table 2: Synthetic-data quality with Qwen2.5-7B. Lower Avg. WD/JSD and higher similarities are better. † denotes quoted results.

Method	Avg. WD ↓	Avg. JSD ↓	Text Sim. ↑	Overall Sim. ↑
Kickstarter				
CLLM†	6230.8	0.1894	0.5503	0.8524
TabDDPM+LLM†	140988.4	0.0136	0.6059	0.8439
MALLM-GAN	6256.9	0.2004	0.5658	**0.8886**
SelPE ($\varepsilon = \infty$)	7086.03	0.1067	0.6041	0.8499
AFT-Tab†	4930.8	0.1626	0.5492	0.8717
CAST	**284.8**	**0.0028**	**0.6559**	0.8594
Cloth				
CLLM†	248.77	0.1733	0.6018	0.8626
TabDDPM+LLM†	27.012	0.2615	0.5073	0.8641
MALLM-GAN	3.666	0.2531	0.6221	0.9052
SelPE ($\varepsilon = \infty$)	2.2057	0.3521	0.6092	0.9031
AFT-Tab†	4.112	0.1339	0.5316	0.8950
CAST	**0.1715**	**0.0130**	**0.7242**	**0.9438**
Wine Reviews				
CLLM†	11.291	0.2946	0.7698	0.9200
TabDDPM+LLM†	3.513	0.2104	0.7310	0.9193
MALLM-GAN	8.552	0.3350	0.7692	0.9238
SelPE ($\varepsilon = \infty$)	6.3745	0.3438	0.7528	0.9206
AFT-Tab†	2.639	0.1377	0.7779	**0.9387**
CAST	**0.0482**	**0.0073**	**0.7783**	0.9337

Distributional fidelity and downstream utility capture different properties of synthetic data. Table 2 shows that CAST performs strongly on structured marginals and text similarity across all three datasets, but Overall Similarity gives a different ranking. MALLM-GAN on Kickstarter and AFT-Tab on Wine Reviews provide stronger overall representation matches while remaining weaker in TSTR. Global representation matching therefore does not alone determine training value. Once structured rows and retained rows are fixed, CAS and RCS can reorganize text but cannot alter Avg.

WD/JSD. These metrics characterize structured generation and retention, whereas downstream and conditional evaluations respond to textual composition.

Table 3: Structured-generator control with reference baselines on Cloth with Qwen2.5-7B. Utility is average $R^2 \times 100$ over A/L/E. CAST configurations use the same subsequent construction and selection rules. Lower WD/JSD and higher utility are better. † denotes quoted results.

Configuration	Avg. WD ↓	Avg. JSD ↓	Avg. Utility ↑
Reference baselines			
AFT-Tab†	4.112	0.1339	60.90
TabDDPM+LLM†	27.012	0.2615	13.30
Structured-generator control			
Gaussian Copula + CAST	0.171452	0.013045	65.63
CTGAN + CAST	5.230125	0.152567	65.23

The structured-generator control further separates marginal fidelity from training value. Table 3 provides two contextual references: AFT-Tab is the strongest Qwen2.5-7B utility baseline on Cloth, while TabDDPM+LLM represents staged structured–text synthesis. CTGAN + CAST exceeds both in utility, although its reported WD/JSD values are higher than AFT-Tab's.

The two CAST configurations provide the controlled comparison. Replacing Gaussian Copula with CTGAN substantially changes structured marginal fidelity while leaving average utility close. The subsequent candidate-construction and selection rules are unchanged, so the strong Cloth utility is not explained solely by the favorable marginal fidelity of the Gaussian-Copula realization. Together with Table 2, this result supports treating structured marginals and training utility as complementary measures. Appendix E.4 provides the full strategy-level results.

4.5 Visualization of Synthetic Distributions

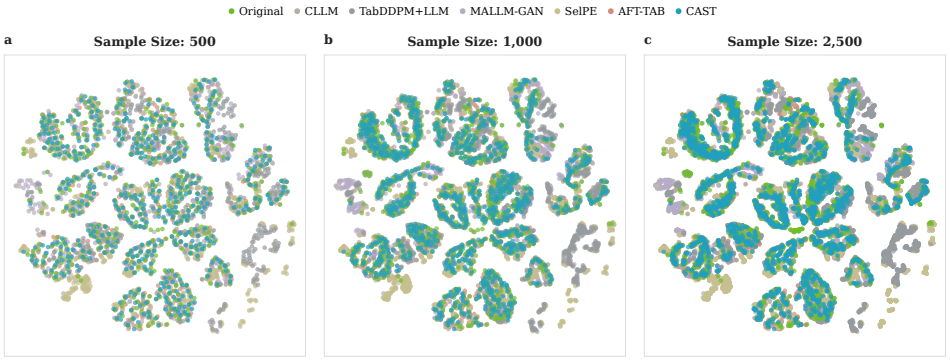


Figure 4: t-SNE visualization of real and synthetic joint representations on Cloth with Qwen2.5-7B. Panels (a)–(c) show 500, 1,000, and 2,500 samples per source under the same projection.

The joint-representation visualization provides a local view of the distributional patterns. Across the three display densities in Figure 4, CAST remains interleaved with real samples across several major regions of the shared t-SNE projection (van der Maaten & Hinton, 2008). AFT-Tab also occupies many of the same local regions, whereas TabDDPM+LLM forms a more visibly separated region and some SelPE samples lie near the boundaries of real-data clusters. The projection therefore provides qualitative evidence that CAST covers major regions of the real representation rather than forming a distinct synthetic island. However, the overlap pattern also reinforces the quantitative result in Section 4.4: methods with visually similar global coverage can still differ substantially in downstream training value. We therefore interpret the projection as a local qualitative complement to the distributional metrics and TSTR evaluation rather than as a standalone measure of synthetic-data quality.

4.6 Ablation Study

Candidate construction and selection both contribute to utility. Table 4 shows the largest observed decrease for w/o SSG (no mixing), while removing CAS or RCS from the complete pipeline also reduces average utility. The gains vary by input strategy: w/o RCS is strongest on Fuse-late, whereas Full CAST leads on All-text, Fuse-early, and the average. Thus, the improvement in the assembled table is not a uniform increase under every downstream representation. The w/o SSG comparison also removes final mixing; the controlled no-mixing study in Appendix E.1 separates the role of candidate construction. Both the shared semantic plan and narrative roles improve utility under matched no-mixing settings, and their combination gives the strongest pre-mixing result. The same structured rows and candidate budget are used throughout these comparisons, placing the observed differences in the text-construction and selection stages. This supports the role of SSG as more than simple prompt diversification: it creates alternatives that share a common content constraint while differing in how that content is expressed. Such a pool is particularly important for CAST because the later selector cannot create new information; it can only organize the candidates supplied by SSG.

Table 4: Cloth ablations ($R^2 \times 100$; Qwen2.5-7B). Avg. averages A/L/E.

Method	All-text	Fuse-late	Fuse-early	Avg.
Full CAST	**63.48**	64.43	**68.99**	**65.63**
w/o SSG (no mixing)	59.36	60.91	58.79	59.69
w/o CAS	61.38	65.48	66.51	64.46
w/o RCS	61.03	**67.28**	66.07	64.79

The component effects exhibit complementarity rather than simple additivity. On the candidate pool generated without SSG, CAS shows little average difference from random selection, whereas its contribution becomes clearer on the shared-plan candidate pool (Appendix E.1). The fixed-pool study in Appendix E.2 examines selection independently of regeneration: with structured rows and candidate texts held fixed, CAST outperforms target-margin maximization both before and after the matched final-mixing step, while CAST also has a lower conditional predictive-feature distance measured using an independent evaluation predictor. Final mixing improves CAST's utility but reduces the margin baseline's utility, even though the latter's conditional predictive-feature distance decreases. The independent evaluator therefore complements TSTR by measuring conditional text–attribute associations, while the mixing contrast shows why a smaller predictive-feature distance alone cannot determine the value of a synthetic training table. Together, these results match the intended construction-and-composition view of CAST: useful candidates must first be created, CAS must organize them using conditional and domain information, and RCS acts conservatively on the resulting selection state rather than as an independently beneficial post-processing rule. With each synthetic table fixed, CAST's pre-mixing advantage over target-margin selection and its gains from mixing and reliability control remain positive across the evaluated downstream-training seeds (Appendix D.5).

5 Conclusion

We introduced CAST, which organizes tabular synthesis with long-text columns as candidate construction under structural constraints followed by dataset-aware selection. CAST combines shared semantic planning and role-based candidate generation with conditional real references, adversarial domain feedback, reliability control, and row-level mixing, while keeping the text generator frozen. Across three datasets, CAST achieves higher downstream utility than the compared synthesis methods, and the ablations show that both candidate construction and selection contribute to this gain. Our analyses further show that stronger label cues or higher overall distributional similarity do not necessarily translate into better training utility. Synthetic-data quality therefore depends on both the quality of individual records and how candidate textual variations are organized across the dataset. The structured-generator control further distinguishes marginal fidelity from training value. Future work can strengthen semantic validation and privacy protection for specialized document structures.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets as part of the CAST pipeline, assist with translation, provide feedback on experimental design, and assist with the interpretation and organization of experimental results. We did not use generative AI tools to develop the core theoretical or conceptual framework, formulate mathematical claims, implement the proposed methods, or clean and reformat the datasets; qualitative and thematic data analysis is not applicable to this work. We reviewed all AI-assisted work: synthetic data were evaluated using the reported experimental protocols, translations and manuscript text were checked by the authors, and AI-assisted experimental discussion was verified against the underlying results and sources. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available benchmark datasets and does not collect new human-subject data. Because CAST retrieves real text as content and style references, synthetic outputs may reproduce or closely paraphrase source text. Appendix D.3 reports exact-match and near-duplicate diagnostics, but these checks do not provide formal privacy guarantees. Additional privacy and domain-specific safeguards would therefore be required before applying CAST to sensitive or proprietary data.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B describe the CAST framework, algorithm, and implementation settings, while Appendix C documents the datasets and evaluation protocol and Appendices D–E provide supplementary analyses and controlled experiments. We will additionally provide supplementary source code, the original experimental data used in this work, and the evaluation code to support reproduction of the reported results.

REFERENCES

Abien Fred Agarap. Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (RNN), 2018. URL https://arxiv.org/abs/1805.03687.

Anthropic. What's new in Claude Opus 5. Claude Platform documentation. URL https://platform.claude.com/docs/en/models/opus-5/whats-new-opus-5. Accessed 2026-09-11.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223, 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cEygmQNOeI.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336. Association for Computing Machinery, 1998. doi: 10.1145/290941.291025.

Sonia Cromp, Satya Sai Srinath Namburi GNVV, Mohammed Alkhudhayri, Catherine Cao, Samuel Guo, Nicholas Roberts, and Frederic Sala. Tabby: A language model architecture for tabular and structured data synthesis, 2026. URL https://arxiv.org/abs/2503.02152.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs, 2017. URL https://arxiv.org/abs/1706.02633.

Liancheng Fang, Aiwei Liu, Hengrui Zhang, Henry Peng Zou, Weizhi Zhang, and Philip S. Yu. TABGEN-ICL: Residual-aware in-context example selection for tabular data generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20027–20041, 2025. doi: 10.18653/v1/2025.findings-acl.1027.

Eibe Frank and Mark Hall. A simple approach to ordinal classification. In *Machine Learning: ECML 2001*, volume 2167 of *Lecture Notes in Computer Science*, pp. 145–156. Springer, 2001. doi: 10.1007/3-540-44795-4_13.

Jingtong Gao, Zhaocheng Du, Xiaopeng Li, Yichao Wang, Xiangyang Li, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. SampleLLM: Optimizing tabular data synthesis in recommendations. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 211–220, 2025. doi: 10.1145/3701716.3715253.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://www.jmlr.org/papers/v13/gretton12a.html.

Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabStruct: Measuring structural fidelity of tabular data. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/447ac93bf22099aa346a45577376492d-Abstract-Conference.html.

Jinhee Kim, Taesung Kim, and Jaegul Choo. EPIC: Effective prompting for imbalanced-class data synthesis in tabular data classification via large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 31504–31542, 2024.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17564–17579, 2023. URL https://proceedings.mlr.press/v202/kotelnikov23a.html.

Lathwal. Funding successful projects on kickstarter. Kaggle dataset. URL https://www.kaggle.com/datasets/codename007/funding-successful-projects. Accessed 2026-09-18.

Anton D. Lautrup, Tobias Hyrup, Arthur Zimek, and Peter Schneider-Kamp. SynthEval: A framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Mining and Knowledge Discovery*, 39:6, 2025. doi: 10.1007/s10618-024-01081-4.

Yaobin Ling, Xiaoqian Jiang, and Yejin Kim. MALLM-GAN: Multi-agent large language model as generative adversarial network for synthesizing tabular data. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 120–136, 2026. doi: 10.18653/v1/2026.findings-acl.7.

María Luisa Menéndez, Julio Angel Pardo, Leandro Pardo, and María del C. Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. doi: 10.1016/S0016-0032(96)00063-4.

Dang Nguyen, Sunil Gupta, Kien Do, Thin Nguyen, and Svetha Venkatesh. Generating realistic tabular data with large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 330–339. IEEE, 2024. doi: 10.1109/ICDM59182.2024.00040.

OpenAI. GPT-5.6 Sol model. OpenAI API documentation. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed 2026-09-11.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399–410. IEEE, 2016. doi: 10.1109/DSAA.2016.49.

Qwen et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.8-27B model card. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed 2026-09-11.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated LLM: Synergy of LLMs and data curation for tabular augmentation in low-data regimes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44060–44092, 2024. URL https://proceedings.mlr.press/v235/seedat24a.html.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. TabDiff: A mixed-type diffusion model for tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5c882988ce5fac487974ee4f415b96a9-Paper-Conference.pdf.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alexander J. Smola. Benchmarking multimodal AutoML for tabular data with text fields. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/9bf31c7ff062936a96d3c8bd1f8f2ff3-Abstract-round2.html.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL https://www.jmlr.org/papers/v9/vandermaaten08a.html.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html.

June Yong Yang, Geondo Park, Joowon Kim, Hyeongwon Jang, and Eunho Yang. Language-interfaced tabular oversampling via progressive imputation and self-authentication. In *The Twelfth International Conference on Learning Representations*, 2024.

Congjing Zhang, Ryan Feng Lin, Ruoxuan Bao, and Shuai Huang. Team, then trim: An assembly-line LLM framework for high-quality tabular data generation, 2026a. URL https://arxiv.org/abs/2602.04785.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9750610639c3e7a849cff746bf60dbd-Abstract-Conference.html.

Tianping Zhang, Shaowen Wang, Shuicheng Yan, Jian Li, and Qian Liu. Generative table pre-training empowers models for tabular prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14836–14854, 2023. doi: 10.18653/v1/2023.emnlp-main.917.

Yuhao Zhang, Liang Yan, Shaoming Duan, Xinyu Zha, Jinhang Su, Peiyi Han, and Chuanyi Liu. AFT-Tab: Adversarial fine-tuning for tabular data synthesis with long text columns. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4581–4594, 2026b. doi: 10.18653/v1/2026.acl-long.209.

Xuancheng Zhu, Guoshun Nan, Han Zhang, Ben Niu, Yang Yue, Zixu Wang, Yilian Liu, Min Lei, and Xiaofeng Tao. SelPE: Progressive selection for private structured text synthesis. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3817858.

Zynicide. Wine reviews. Kaggle dataset, 2017. URL https://www.kaggle.com/datasets/zynicide/wine-reviews. Accessed 2026-09-18.

Appendix

CAST: Conditional Generation and Adversarial Selection for Tabular Data Synthesis with Long Text Columns

Table of Contents

A Additional Related Work . . . 14
- A.1 Mixed-Type Tabular Data Generation . . . 14
- A.2 Fine-Tuning-Based Tabular Synthesis . . . 14
- A.3 Frozen-Model Tabular Synthesis . . . 14

B Algorithm and Implementation Details . . . 15
- B.1 CAST Algorithm . . . 15
- B.2 Structure-Aware Semantic Generation (SSG) . . . 15
- B.3 Conditional Adversarial Selection (CAS) . . . 16
- B.4 Reliability-Guided Conservative Selection (RCS) . . . 17

C Data and Evaluation Setup . . . 18
- C.1 Datasets and Splits . . . 18
- C.2 Evaluation Protocol and Result Sources . . . 19

D Additional Results and Analysis . . . 21
- D.1 GPU Total Time . . . 21
- D.2 Utility Across Generator Backbones . . . 21
- D.3 Supplementary Checks and Visualization . . . 21
- D.4 Candidate Construction and Conditional Selection Cases . . . 22
- D.5 Repeated Evaluation . . . 25

E Ablation Studies . . . 26
- E.1 Module Ablations and Candidate Construction . . . 26
- E.2 Fixed-Pool Selection and Reselection Stages . . . 28
- E.3 Reliability Control and Final Mixing . . . 29
- E.4 Alternative Structured-Generator Control . . . 30

F Discussion and Scope . . . 30

A Additional Related Work

A.1 Mixed-Type Tabular Data Generation

Mixed-type tabular synthesis requires modeling the distributions and dependencies of numerical and categorical fields. CTGAN handles multimodal continuous distributions and class imbalance through mode-specific normalization and conditional sampling (Xu et al., 2019). Diffusion-based methods use different mechanisms for heterogeneous fields. TabDDPM combines Gaussian and multinomial diffusion for numerical and categorical variables, respectively (Kotelnikov et al., 2023). TabSyn first maps mixed-type fields into a continuous latent space with a variational autoencoder and then learns the latent distribution with a diffusion model (Zhang et al., 2024). TabDiff instead uses a joint continuous-time diffusion process with feature-wise learnable noise schedules to adapt to different field distributions (Shi et al., 2025).

Structured tabular generators can also be combined with language models to synthesize tables containing long-text columns, as in our TabDDPM+LLM baseline. However, structured generators alone do not directly model long-form textual content or its semantic dependencies on structured attributes.

A.2 Fine-Tuning-Based Tabular Synthesis

Language models provide a unified sequential representation of field names, attribute values, and natural-language content. GReaT serializes tabular records and fine-tunes an autoregressive language model, using randomized feature orderings to support flexible conditional generation (Borisov et al., 2023). TapTap performs table-oriented pretraining over multiple datasets before fine-tuning and sampling on the target dataset (Zhang et al., 2023). Tabby introduces field-specific expert parameters to capture differences across features during fine-tuning (Cromp et al., 2026). Nguyen et al. (2024) combine a fine-tuning permutation strategy with feature-conditional sampling and label generation conditioned on sampled features to improve feature–target relationships in synthetic tables.

For tables with long-text columns, AFT-Tab adversarially fine-tunes a language-model-based text generator using discriminator feedback and replays selected samples to optimize the non-text generator, thereby coordinating structured distributions with textual semantics (Zhang et al., 2026b). It also introduces an association-fidelity metric based on how well text predicts structured attributes in real and synthetic data.

These methods require updating generator parameters for each target dataset, which adds training cost and limits their use when a model is available only through an inference API. CAST instead keeps the text-generation model frozen and incorporates predictive associations into candidate selection, using predictive features from real condition groups to guide reselection.

A.3 Frozen-Model Tabular Synthesis

Frozen-model approaches keep the generative language model fixed and improve synthesis through in-context demonstrations, iterative feedback, or post-generation selection. CLLM combines few-shot generation with learning-dynamics-based data filtering to improve the quality of synthetic training data (Seedat et al., 2024). EPIC improves prompting for imbalanced tabular classification through balanced group-wise demonstrations, consistent formatting, and variable mapping (Kim et al., 2024). Language-Interfaced Tabular Oversampling uses conditional generation through progressive imputation and filters insufficiently transformed samples through self-authentication (Yang et al., 2024). TABGEN-ICL iteratively selects in-context examples according to differences between synthetic and real distributions (Fang et al., 2025). MALLM-GAN uses a supervised discriminator to evaluate synthetic records and feeds the resulting feedback to language-model agents that refine generation (Ling et al., 2026). SampleLLM applies importance-aware sampling after example-guided generation to reduce distributional bias while accounting for feature interactions (Gao et al., 2025). For differentially private structured-text synthesis, SelPE constructs semantic context, generates schema-constrained records, and progressively selects samples using distances over heterogeneous fields (Zhu et al., 2026). Team-then-Trim combines sequential generation by specialized

LLM workers with post-generation checks of validity, task-related costs, and diversity (Zhang et al., 2026a).

These methods improve synthesis through contextual guidance, distributional feedback, or sample selection. CAST instead constructs multiple text candidates for each fixed structured row and reselects them using differentiated real references within condition groups together with feedback from the current synthetic table. Its focus is therefore on how alternative textual expressions under fixed structured conditions are assembled into a synthetic table with downstream training value.

B Algorithm and Implementation Details

B.1 CAST Algorithm

Algorithm 1 CAST synthesis

Require: Real data $\mathcal{D}$, frozen generator G_θ, schema and feedback configuration, maximum rounds $R_{\max}$
Ensure: Synthetic table $\widetilde{\mathcal{D}}$

```
Synthesize and validate structured rows {s̃_i}, including the target
for each structured row i do
    Retrieve disjoint content and style references
    Extract shared plan p_i; generate and validate four role-based candidates
end for
Fit initialization predictors and the fixed Domain Discriminator
Form valid pools {J_i}_{i∈I} and initial selections j^(0)
Fit fixed real-data feedback predictors; assign references ref(i)
Initialize and gate joint feedback once; set R ← 0
for r = 0, ..., R_max − 1 do
    Refit synthetic-data text predictors on D̃_{j^(r)}
    Gate and weight text feedback using Equations (9) and (4)
    if all primary-target text feedback is invalid then
        break
    end if
    Compute conditional task costs using Equation (6)
    Reselect candidates using Equation (3); set R ← r + 1
end for
Apply row-level mixing to j^(R) using Equation (5)
return the synthetic table defined by Equation (1)
```

B.2 Structure-Aware Semantic Generation (SSG)

Structured row synthesis. We fit a Gaussian-Copula-based generator to the non-text fields, including the primary target. Numerical marginals are modeled empirically, while categorical combinations that must remain valid are modeled jointly. Generated rows are checked for marginal distributions, cross-column associations, field constraints, and temporal consistency. Accepted structured rows remain fixed throughout text generation and selection.

Reference retrieval and candidate generation. For each structured row, SSG retrieves references from three sources: records with the same target, records from nearby target groups, and records matching the other structured attributes, using approximate quotas of $1/2$, $1/4$, and $1/4$. Numerical targets use exact-value groups when low-cardinality and quantile groups otherwise. Within each source, maximal marginal relevance (MMR) (Carbonell & Goldstein, 1998) balances structural relevance and textual diversity. Content and style references are selected as disjoint sets.

The frozen generator extracts a shared semantic plan from the structured condition and content references, then generates four candidates using that plan and the style references, as in Equation (2). Content references support the evidence to be expressed; style references provide domain-specific expression patterns.

Narrative roles. All roles are instructed to retain the same supported evidence, assessment, and outcome in the shared plan, without adding unsupported facts. The roles in Table 5 change discourse order and emphasis under the same structured condition, shared plan, and output-field constraints.

Table 5: Narrative roles used by SSG under a shared structured condition and semantic plan.

Role	Generation emphasis
Balanced Evidence	Present favorable and unfavorable evidence in a balanced progression.
Experience First	Begin with concrete observations or experience and place the overall assessment later.
Decision First	State the decision or outcome first, when present, then support it with evidence.
Implicit Reflection	Let the assessment emerge from evidence without explicit label-like wording.

Validity checks. Failed plan extraction is retried with simplified prompt formats; failed role generation is retried under the same role constraint. Candidates must satisfy field-completeness and length constraints and must not exactly copy their own style references after normalization. Candidates with non-finite initialization costs are removed. The resulting candidate pool is fixed for CAS and RCS. Saved plans and candidate examples appear in Appendix D.4.

B.3 Conditional Adversarial Selection (CAS)

Feedback predictors and task features. Text-view feedback covers the primary target and configured auxiliary attributes; the condition field used below is selected separately. Let $\mathcal{D}_{\mathrm{fit}}$ and $\mathcal{D}_{\mathrm{dev}}$ denote the sampled real training and validation data. Predictors use TF-IDF features with linear classifiers or Ridge regression according to attribute type. Real-data predictors are fitted once on $\mathcal{D}_{\mathrm{fit}}$ and remain fixed; synthetic-data text predictors are refit on the current synthetic table before each round. A joint predictor for the primary target additionally combines non-target structured fields, text, and their interactions. It follows the downstream task: nominal classification or Ridge regression on the original numerical target. Its real-data model provides fixed matching features, while its synthetic-data counterpart is fitted only on the initial table for one-time transfer gating.

For feedback term h, which identifies the attribute, input view, prediction mode, and predictor source, ϕ_h^{main}, ϕ_h^{aux}, and $\hat{c}_h$ denote its main feature, auxiliary feature, and predicted class or level. Categorical attributes use the own-label probability minus the largest alternative-class probability, predictive entropy, and predicted class. Numerical attributes use only the signed observed-minus-predicted residual on the predictor's target scale. Ordinal attributes use cumulative classification (Frank & Hall, 2001): the observed-minus-expected level, mean binary entropy across thresholds, and rounded, clipped expected level. A nominal-classification term also contributes to an ordinal attribute's shared text-feedback weight.

Conditional matching cost. Candidate (i, j) and its fixed Conditional Reference Record $\mathrm{ref}(i)$ are evaluated by the same predictor:

$$
\begin{aligned}
C_{ij,h}^{(r)} = & \frac{\left|\phi_h^{\mathrm{main}}(i,j) - \phi_h^{\mathrm{main}}(\mathrm{ref}(i))\right|}{\sigma_h^{\mathrm{main}}} \\
& + \alpha \frac{|\phi_h^{\mathrm{aux}}(i,j) - \phi_h^{\mathrm{aux}}(\mathrm{ref}(i))|}{\sigma_h^{\mathrm{aux}}} \\
& + \beta\, \mathbf{1}\left[\hat{c}_h(i,j) \neq \hat{c}_h(\mathrm{ref}(i))\right].
\end{aligned} \tag{6}
$$

Here, $\alpha = 0.25$, $\beta = 0.35$, and inapplicable terms are omitted. Each continuous-feature scale σ_h is computed from real validation features as $\max(\mathrm{IQR}, 0.5\,\mathrm{std}, 10^{-3})$. Round indices on features and scales are omitted. Real-data features and scales remain fixed; synthetic-data features and scales are recomputed after refitting, with reference records unchanged. For attributes outside the condition key, candidate and reference labels may differ; the last term still compares their predicted classes.

Conditional reference assignment. Condition keys use the exact primary-target value y and at most one non-identifier categorical field c, which may be specified or selected automatically. Let κ_c

be its category count and γ_c the fraction of synthetic rows whose (y, c) combination occurs in $\mathcal{D}_{\text{dev}}$. Automatic selection maximizes

$$S(c) = \text{NMI}_{\mathcal{D}_{\text{fit}}}(y, c) \frac{\sqrt{\gamma_c}}{\sqrt{\max\left(\log_2(\kappa_c + 1), 1\right)}} \tag{7}$$

among fields with $\gamma_c \geq 0.95$ and non-negative score; otherwise, only y is used. NMI denotes normalized mutual information over discrete field values. Missing complete conditions fall back to the primary-target group and, if necessary, the full validation set.

Within each group, real references are sorted in ascending order of the primary feature from the real-data predictor; ordinal targets use the nominal-classification margin. Synthetic rows are ordered by a fixed-seed hash of row and group identifiers. For a group with n_g synthetic rows and m_g real references, zero-based synthetic position k receives reference position

$$q_g(k) = \left\lfloor \frac{(k + 1/2)m_g}{n_g} \right\rfloor. \tag{8}$$

The assigned record $\text{ref}(i)$ is shared by all feedback terms and remains fixed throughout reselection. This distributes references across the group's observed predictive-feature range.

Initialization and adversarial selection. Initialization uses separately fitted primary-target predictors and target-based reference groups: classes for classification, and exact values or quantile groups for numerical and ordinal targets. Corresponding task-feature differences are normalized within each reference group. The valid candidate minimizing the sum of raw task-matching cost and a small domain-cost term defines $j_i^{(0)}$. Only valid candidates with finite initialization costs enter reselection; their raw task costs are converted to normalized within-row ranks to obtain C_{ij}^{init}.

The Domain Discriminator uses TF-IDF features and a linear logistic model trained on equal numbers of real and candidate texts. Its Domain Scores remain fixed; C_{ij}^{dom} is one minus the normalized within-row score rank, so more real-like candidates have lower domain cost. Equation (3) then reselects candidates row by row using RCS weights. Updated selections determine the next synthetic-data text-predictor fit, coupling rows through their shared predictor. The candidate pool, structured rows, real-data predictors, conditional references, and Domain Discriminator remain fixed.

B.4 Reliability-Guided Conservative Selection (RCS)

Reliability gating. Let $M_h^{(r)}$ be the validation performance of feedback predictor h after r completed reselection rounds, and M_h^{base} its baseline. Classification feedback uses Accuracy with the majority-class proportion as baseline; regression and cumulative-ordinal feedback use R^2 with baseline zero. A feedback term passes gating when its finite validation performance exceeds its baseline by more than δ. For a synthetic-data predictor, if the corresponding real-data predictor is valid, its performance must additionally be no lower than the real-data performance minus τ.

Effective reliability is

$$\eta_h^{(r)} = \begin{cases} \text{clip}\left(\dfrac{M_h^{(r)} - M_h^{\text{base}}}{\max(1 - M_h^{\text{base}}, \epsilon)}, 0, 1 \right), & \text{if the feedback passes gating,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here, $\text{clip}(\cdot, 0, 1)$ clips to $[0, 1]$ and ϵ ensures numerical stability. Real-data reliability remains fixed; synthetic-data text reliability is updated after each refit. Equation (4) distributes the attribute-level weight W_a across its valid real-data and synthetic-data text terms, including both prediction modes for ordinal attributes. This gate-then-redistribute rule keeps W_a fixed whenever valid feedback exists; reliability controls its allocation, not its absolute magnitude. An attribute receives zero text-feedback weight if none of its terms is valid.

Joint feedback and stopping. For the primary target, joint feedback requires both the real-data predictor h and the synthetic-data predictor h' fitted on the initial table to pass the baseline and relative-tolerance checks. Its weight is $0.35\sqrt{\eta_h^{(0)}\eta_{h'}^{(0)}}$, or zero if either check fails. This gate and weight are computed once; the fixed real-data joint predictor supplies the matching features. Reselection stops if all primary-target text feedback becomes invalid, even when joint feedback remains valid. If this occurs before the first round, the initial selections are retained.

Shared parameters. Table 6 lists the shared empirical settings, including the default maximum of two reselection rounds. These values were chosen heuristically and shared across datasets.

Table 6: Shared CAST parameters and default settings used across experiments.

Parameter	Default Setting
Number of candidate roles / max. reselection rounds	$4/2$
Initial matching weight λ_{init} / domain weight λ_{dom}	$0.20/0.35$
Primary-target text-feedback weight / default auxiliary text-feedback weight	$1.0/0.35$
Baseline margin δ / synthetic-predictor tolerance τ	$0.01/0.15$
Joint-feedback coefficient	0.35
Numerical-stability constant ϵ	10^{-8}
Mixing threshold ρ / preset narrative role	0.75 / Experience First

Final mixing. Equation (5) mixes the last CAS selections with the preset Experience First candidates using a fixed threshold ρ and stable row-identifier hashes. If no round is completed, $R = 0$ and the CAS branch uses $j_i^{(0)}$; if the preset role is unavailable, it also falls back to $j_i^{(0)}$. The threshold does not depend on per-row prediction reliability. Mixing selects all text fields together and leaves the structured row unchanged.

C Data and Evaluation Setup

C.1 Datasets and Splits

We use the fixed training and test data released with AFT-Tab (Zhang et al., 2026b), whose tasks are drawn from the multimodal text–tabular benchmark (Shi et al., 2021). The public sources for Kickstarter, Cloth, and Wine Reviews are given by (Lathwal), (Agarap, 2018), and (Zynicide, 2017), respectively.

The original training set is split into $\mathcal{D}_{\text{fit}}$ and $\mathcal{D}_{\text{dev}}$. The former is used to train real-data feedback predictors, while the latter is used to construct conditional references, compute feature scales and condition coverage, and evaluate feedback reliability. Both are available to the synthesis procedure. The real test set is reserved for TSTR evaluation.

Each synthesis run targets the size of the corresponding original training set. The final output contains only records that pass validity filtering, so the retained number of rows may be smaller than the target generation size.

Table 7: Dataset sizes and prediction tasks. All sizes are numbers of records.

Dataset	Train	$\mathcal{D}_{\text{fit}}$	$\mathcal{D}_{\text{dev}}$	Test	Target	Task
Kickstarter	86,502	69,201	17,301	21,626	`final_status`	Binary cls.
Cloth	18,788	15,030	3,758	4,698	`Rating`	Regression
Wine Reviews	84,123	67,298	16,825	21,031	`variety`	Multiclass cls.

The field sets, field types, and downstream inputs follow the corresponding AFT-Tab configurations (Zhang et al., 2026b). Table 8 lists the prediction target, primary text field, and other main fields for each dataset.

Table 8: Dataset fields. Bold indicates the prediction target. The primary text field corresponds to the text-length statistics in Table 9.

Dataset	Target	Primary Text	Other Main Fields
Kickstarter	**final_status**	desc	goal, disable_communication, country, currency, deadline, created_at, name, keywords
Cloth	**Rating**	Review Text	Age, Positive Feedback Count, Division Name, Department Name, Class Name, Title
Wine Reviews	**variety**	description	country, points, price, province

We follow the task names used by the long-text tabular benchmark. The datasets range from short descriptions to paragraph-level reviews. Table 9 summarizes whitespace-separated word counts for non-empty primary-text fields in the original training sets. For comparison, the median lengths of Kickstarter `name`, Cloth `Title`, and Wine Reviews `province` are substantially shorter.

Table 9: Primary-text length statistics in the original training sets. Word counts use whitespace tokenization and include only non-empty texts; missing or empty texts are counted separately. P10, P50, and P95 denote the 10th, 50th, and 95th percentiles.

Dataset	Text Field	Non-empty	Missing	Mean	P10	P50	P95 / Max
Kickstarter	desc	86,496	6	19.5	13	20	26 / 35
Cloth	Review Text	18,121	667	60.3	22	59	102 / 115
Wine Reviews	description	84,123	0	40.5	27	40	60 / 135

Kickstarter has an imbalanced binary target, with more unsuccessful than successful projects. Cloth uses ratings from 1 to 5, with samples concentrated at higher ratings, and is evaluated as a regression task. Wine Reviews contains 30 wine varieties with an imbalanced class distribution.

C.2 Evaluation Protocol and Result Sources

We use the same evaluation environment and protocol as AFT-Tab to keep comparisons consistent (Zhang et al., 2026b). Dataset fields, field types, and evaluation inputs follow the corresponding benchmark configurations. Previously reported values are quoted directly, while settings not reported in prior work are run locally under the same evaluation setup.

The original-training-data reference and the Qwen2.5-7B results for CLLM, TabDDPM+LLM, and AFT-Tab in Table 1 are quoted from prior reports. The corresponding synthetic-data quality results in Table 2, as well as the reported AFT-Tab GPU time in Appendix D.1, are also quoted directly. Values reused in Figure 1, Figure 3, and Appendix D.2 follow the same sources. Results for CAST, the ablations, and the supplementary analyses are obtained from local runs. Quoted values are marked with †.

Downstream utility. TSTR (Esteban et al., 2017) trains downstream predictors on synthetic data and evaluates them on the real test set. Kickstarter, Cloth, and Wine Reviews use AUC, R^2, and Accuracy, respectively, all reported after multiplying by 100. Unless otherwise stated, locally evaluated downstream models use training seed 2026. Appendix D.5 additionally reports repeated downstream training with seeds 2026, 2027, and 2028 while keeping the synthetic data fixed.

All-text, Fuse-late, and Fuse-early follow the three modeling strategies defined in the multimodal text–tabular benchmark (Shi et al., 2021). Avg. denotes their arithmetic mean.

Table 10: Shared evaluation settings. Dataset splits are given in Appendix C.1, and CAST parameters are listed in Table 6.

Item	Setting
Downstream tasks	Kickstarter: `final_status`, AUC; Cloth: `Rating`, R^2; Wine Reviews: `variety`, Accuracy. All utility values are multiplied by 100.
Input organization	All-text, Fuse-late, and Fuse-early follow the multimodal text–tabular benchmark; Avg. is their arithmetic mean.
Ablation controls	All variants use the same data splits, downstream-training setup, random seed, and evaluation protocol, while following their own selection trajectories.
Text length	Ablations use the same data-adaptive length rule, with the effective truncation length determined from the corresponding training data.
Distributional evaluation	Table 2 retains reported structured-marginal, text-representation, and overall-table metrics. Joint-MMD^2 is used in Appendix E as an additional diagnostic of joint structured–text distribution differences.
GPU time	Appendix D.1 reports GPU total time under the same accounting convention for the four-GPU setting. Reported minute values are retained as given and are not multiplied by the number of GPUs.

Distributional metrics. Avg. WD and Avg. JSD in Table 2 are based on Wasserstein distance (Arjovsky et al., 2017) and Jensen–Shannon distance (Menéndez et al., 1997), respectively. For consistency with the benchmark terminology, we retain the label Avg. JSD; the evaluation implementation computes Jensen–Shannon distance, i.e., the square root of Jensen–Shannon divergence. WD uses original field units without scale normalization; per-field distances are averaged. Lower values indicate closer marginals under the same field scope.

Text Similarity compares real and synthetic text representations, while Overall Similarity compares joint structured–text representations; higher values indicate greater representation-level similarity. Once the structured rows and retained-row set are fixed, text reselection does not change Avg. WD or Avg. JSD. These metrics, conditional predictive-feature distances, and TSTR measure different aspects of synthetic-data quality. Broader evaluation frameworks likewise separate complementary dimensions: SynthEval combines fidelity, utility, and privacy-risk metrics, while TabStruct studies structural fidelity alongside conventional distributional and predictive evaluations (Lautrup et al., 2025; Jiang et al., 2026).

Conditional predictive-feature evaluation. For Cloth with Qwen2.5-7B, we use a fixed text predictor that is not involved in candidate selection, referred to as the *conditional evaluation predictor*. It extracts predictive residuals and target-label probability margins from real and synthetic text. The label margin is defined as the probability of the target label minus the largest probability among the remaining classes, following the same form used in Appendix B.3 but computed by this independent evaluator.

Real and synthetic predictive features are compared within `Rating` × `Class Name` groups against cached real-test features. Eligible groups contain at least 30 records in each compared population. Distances are weighted by real-reference counts within the eligible support and then averaged over five attributes. The Full CAST entry in Table 24 uses 33 groups; the fixed-pool and reliability-control results in Appendices E.2–E.3 use the historical common support of 31 groups. These results share the frozen predictor and feature definitions, but rounded values from different supports should not be treated as identical evaluations. Lower conditional W1 indicates closer measured predictive-feature distributions within the specified support.

Joint-MMD^2. For the controlled ablations in Appendix E, we additionally report Joint-MMD^2 (Gretton et al., 2012) over joint structured–text representations using the same source rows and real references across variants. Lower values indicate a smaller discrepancy in the measured joint representation space. This metric is used as a diagnostic rather than as a substitute for TSTR utility.

D Additional Results and Analysis

D.1 GPU Total Time

Table 11 compares reported GPU total time and downstream utility on Wine Reviews with Qwen2.5-7B. CAST requires 102 minutes, about 35% less than the 157 minutes reported for AFT-Tab, while achieving higher average Accuracy. Its reported time is also close to that of MALLM-GAN, with substantially higher downstream utility.

Table 11: GPU total time and downstream utility on Wine Reviews with Qwen2.5-7B. Avg. Accuracy averages the three modeling strategies and is multiplied by 100. † denotes a quoted result.

Method	GPU Total Time (min) ↓	Avg. Accuracy ↑
CLLM	84	41.83
TabDDPM+LLM	**75**	24.83
MALLM-GAN	95	42.93
SelPE ($\varepsilon = \infty$)	214	28.30
AFT-Tab†	157	68.00
CAST	102	**74.77**

D.2 Utility Across Generator Backbones

Table 12: Full values for Figure 3: average utility across generator backbones on Cloth, averaged over All-text, Fuse-late, and Fuse-early ($R^2 \times 100$). † denotes quoted results.

Generator	TabDDPM+LLM	CLLM	CAST
Qwen2.5-7B	13.3†	56.53†	**65.63**
Qwen3.8-27B	22.2	62.47	**67.90**
DeepSeek-V4-Flash	33.2	65.16	**67.02**
Claude Opus 5	27.1	56.78	**67.41**
GPT-5.6 Sol	13.4	59.81	**66.44**

D.3 Supplementary Checks and Visualization

Condition coverage. Following the condition-key and coverage definitions in Appendix B.3, we measure the fraction of synthetic rows whose complete condition combination is present in the real validation data. All three full runs achieve more than 97% complete-condition coverage; unmatched rows fall back to the primary-target group.

Table 13: Complete-condition coverage in the three full CAST runs.

Dataset	Complete-Condition Coverage
Kickstarter	99.88%
Cloth	99.86%
Wine Reviews	97.79%

Text overlap and near-duplicate diagnostics. We compare Full CAST outputs with Qwen2.5-7B against each dataset's real training-text library. Text is normalized with Unicode NFKC, lowercasing, and whitespace collapse. Exact matching compares complete textual-field combinations, while near-duplicate diagnostics and distance to the closest record (DCR) use the main text field.

The near-duplicate threshold is defined separately for each dataset by the 99.9th percentile of leave-one-out nearest-neighbor similarity among deduplicated real texts, before evaluating synthetic outputs. Similarity uses a frozen all-MiniLM-L6-v2 encoder with full-text chunk aggregation. DCR is one minus the nearest-real-text cosine similarity, and DCR P5 is its fifth percentile. Larger DCR indicates greater embedding distance, not a privacy guarantee.

Table 14: Text-overlap diagnostics for Full CAST with Qwen2.5-7B. The near-duplicate threshold is determined separately from each deduplicated real-text library. Lower match rates indicate less observed overlap; higher DCR P5 indicates greater distance from the closest real text.

Dataset	Synthetic Rows	Exact Match	Near-Duplicate Rate ↓	DCR P5 ↑
Cloth	18,784	0 / 18,784	0.2555%	0.128797
Wine Reviews	84,122	0 / 84,122	0.0666%	0.098129
Kickstarter	86,301	0 / 86,301	0.0035%	0.174791

No exact matches are observed for complete textual-field combinations, and near-duplicate rates are low. Near-duplicate rates are 0.2555% on Cloth, 0.0666% on Wine Reviews, and 0.0035% on Kickstarter. These diagnostics characterize observed overlap; they do not rule out partial copying or establish formal privacy guarantees.

Historical content and style references cover all Cloth rows, with zero exact matches against either reference type. The original per-row reference caches for Wine Reviews and Kickstarter were not preserved in the archived outputs. Reference-specific analysis is therefore limited to Cloth, for which the historical per-row reference caches are available.

Visualization. CAST covers several major local clusters of the real data in the shared joint-representation projection. As the displayed sample size increases in Figure 4, CAST remains interleaved with real samples across multiple regions of the projection. AFT-Tab also overlaps with real data in many regions. In contrast, TabDDPM+LLM forms a clearer separated region on the right, while some SelPE samples lie near the outer edges of real-data clusters. These local patterns provide a qualitative complement to the joint-representation comparison in Table 2.

D.4 Candidate Construction and Conditional Selection Cases

We examine two synthetic records from the same condition group (`Rating`=4, `Class Name`=`Dresses`) in the full Cloth × Qwen2.5-7B run. The two records differ in their assigned reference positions and selection trajectories. This condition group contains 1,058 retained synthetic records and 198 conditional references from the real validation set $\mathcal{D}_{\mathrm{dev}}$.

The examples are selected to illustrate different reference positions and actual selection trajectories. Case A shows the full process, while Case B provides a comparison under the same condition group. All English plan excerpts, reference excerpts, titles, and candidate texts below are taken directly from saved outputs and are not regenerated or edited.

Case A: Structured condition, shared plan, and candidates. Structured condition. Selected structured fields for Case A are shown below.

Table 15: Selected structured fields for Case A.

Age	Rating	Division Name	Department Name	Class Name
35	4	General Petite	Dresses	Dresses

The SSG content reference is **Great transition dress**, while the style reference is **Nice dress, fit is ok**. The former is used to construct the shared semantic plan and the latter provides an expression reference. Neither is the conditional reference later assigned by CAS.

The relevant content-reference text is:

> Very charming, well made and comfortable. it is a heavier material with a slip so it's a great spring dress if you have cooler days but i could see in the summer if you have 80f + temps it could be a little warm.

Saved shared semantic plan. Table 16 shows excerpts from the actual saved plan.

Table 16: Saved shared-plan excerpts for Case A.

Plan Entry	Saved Text
Size / fit	"didn't find it too snug"
Fabric / structure	"heavier material with a slip"
Comfort	"comfortable"
Positive evidence	"well made"; "comfortable"; "charming"; "pretty tts for maeve"
Use context	"great spring dress if you have cooler days"
Conditional drawback	"could be a little warm in summer if you have 80f + temps"
Trade-off	"heavier material with a slip, but great for spring with cooler days"

The conditional drawback is stored in both the use-context and negative-evidence fields. The separate `uncertainty`, `overall_assessment`, and `decision_or_outcome` fields are empty. The four candidates are generated under the same structured condition and shared plan, while differing in emphasis, context ordering, and placement of the conclusion.

Four saved candidates.

Balanced Evidence

Title: Charming Dress for Spring

> I love the charming style of this dress, perfect for my petite frame. It's a bit heavy but well-made with a slip underneath. Comfortable to wear, though it could get a bit warm on warmer spring days. Fits true to size, just right for cooler weather.

Experience First

Title: Pretty Dress for the Season

> This dress is pretty and charming, suitable for spring. It feels comfortable and has a nice fit, but it gets a bit warm in warmer weather. I like the heavier material with a slip layer, making it great for transitional seasons.

Decision First

Title: Great Dress for Spring

> Decided to try this dress for spring and it turned out to be a great choice. It fits perfectly and is comfortable to wear. However, it could be a bit warm in summer when the temperature hits the 80s. Overall, a charming and well-made dress for the season.

Implicit Reflection

Title: Spring Dress Fit

> I'm happy with this dress. It's comfortable and fits well, though it does run a bit heavy. Suitable for cool spring weather but might be too warm in hotter months. The charming style and quality make it a good pick.

Case A: Conditional reference, candidate scores, and reselection trajectory. Conditional Reference Record. The CAS reference assigned to this row comes from $\mathcal{D}_{\text{dev}}$ and has title **Adorable!**. Its structured attributes include `Rating`=4, `Class Name`=`Dresses`, `Age`=43, and `Division Name`=`General Petite`.

The reference text is:

> **Title:** Adorable!
>
> Was so glad i gave this dress a chance - it's adorable! it does run large. i am 5'6", 125lb and the 2 fits perfectly. it is super comfortable. i am taking away a star only because it doesn't have pockets - with all of the fabric - how were pockets left out?

Among the 198 real records in this condition group, the reference is ranked 100th by the `Rating` label margin, corresponding to a midpoint percentile of 50.25%, with margin -0.1814. It provides a target for conditional predictive-feature matching and is not a source used to construct the shared semantic plan.

Candidate scores. The label margin is computed by the fixed real-data feedback predictor used in the full run. It is defined as the probability of the target `Rating` minus the highest probability among the remaining classes and corresponds to the nominal-classification feedback for `Rating`. The downstream task itself is still evaluated as rating regression.

The second-round total cost combines the initial-matching term, Domain Score term, and reliability-weighted multi-attribute conditional feedback from Equation (3).

Table 17: Candidate scores and actual selections for Case A. Lower second-round total cost is better. Margins and costs are reported on their original scales.

Candidate Role	Label Margin	Round-2 Cost ↓	Selection
Balanced Evidence	-0.4460	1.6493	–
Experience First	+0.3452	1.6702	Margin max.; initial
Decision First	-0.0017	1.4669	Round 1
Implicit Reflection	+0.1072	**1.3274**	Round 2; Full CAST

The target-margin baseline selects Experience First, whereas the second CAST reselection selects Implicit Reflection, which has the lowest total cost but not the largest label margin. Both reselection rounds operate on the same fixed four-candidate pool.

Cost changes and reliability gating. Table 18 reports the cost change within each reselection round. The "before" and "after" values in the same row are evaluated by the same round-specific scorer.

Table 18: Within-round selection-cost changes for Case A.

Round	Entering	Selected	Before	After	Rel. Drop
1	Experience First	Decision First	1.7807	1.1372	36.14%
2	Decision First	Implicit Reflection	1.4669	1.3274	9.51%

As one example of RCS feedback, the real-data cumulative-ordinal predictor for `Rating` achieves validation $R^2 = 0.4828$ on $\mathcal{D}_{\text{dev}}$. The synthetic-data predictor, refit along the current synthetic-table trajectory, has the validation performance and weights shown in Table 19.

Table 19: Validation performance and gating of the synthetic-data ordinal `Rating` feedback in Case A.

Round	Synthetic R^2	Gap to Real	Weight	Enabled
1	0.3188	0.1640	0.0000	No
2	0.3516	0.1312	0.3343	Yes

In the second round, the weighted initial-matching term decreases from 0.1500 to 0.1000 and the weighted domain term from 0.2625 to 0, while the sum of task-feedback terms increases from 1.0544 to 1.2274. The decreases in the first two terms outweigh the increase in task feedback, reducing the total cost from 1.4669 to 1.3274. This illustrates that the selection reflects a trade-off among multiple feedback terms rather than optimizing any single score.

For the ordinal `Rating` feedback, the round-1 performance gap of 0.1640 exceeds the tolerance of 0.15, so the synthetic feedback receives zero weight. In round 2, the gap decreases to 0.1312 and passes the gating rule, so the feedback participates with weight 0.3343. These values describe the validation-based gating state of the feedback model, not a reliability label assigned to an individual candidate.

Final row-level mixing follows Equation (5). The stable hash for this row is 0.4360, which is below the threshold 0.75, so Full CAST retains the second-round Implicit Reflection candidate.

Text observation. The initial CAST candidate describes the heat-related drawback as "it gets a bit warm in warmer weather," while the Full CAST candidate uses "might be too warm in hotter months." The latter preserves the tentative character of the content reference and shared plan together with the descriptions of comfort, fit, suitability for cooler spring weather, and a generally positive assessment. The phrase "too warm" is nevertheless stronger than the reference phrase "a little warm." This observation concerns the generated text itself; the actual selection is determined by the combined cost reported above.

Reference positions and selections within the same condition group. The two cases differ in age, division, shared semantic plan, reference position, and selection trajectory. Reference position refers to the location of the real record in the within-group predictive-feature ordering, not to a quality or sentiment rank. We abbreviate the four roles as BE, EF, DF, and IR for Balanced Evidence, Experience First, Decision First, and Implicit Reflection.

Table 20: Reference positions and selection trajectories for two records from the same condition group. Values in parentheses are `Rating` label margins. The final column reports initial selection $\rightarrow$ round 1 $\rightarrow$ round 2 $\rightarrow$ final output.

Case	Age / Division	Reference Position	Margin-Max Selection	CAST Round-2 Selection	Selection Trajectory
A	35 / General Petite	50.25% (-0.1814)	EF ($+0.3452$)	IR ($+0.1072$)	EF $\rightarrow$ DF $\rightarrow$ IR $\rightarrow$ IR
B	36 / General	7.32% (-0.6798)	IR (-0.1782)	DF (-0.8522)	EF $\rightarrow$ BE $\rightarrow$ DF $\rightarrow$ DF

For both Cases A and B, the second CAST reselection chooses a different candidate from the target-margin baseline. The selections combine conditional references, multi-attribute task feedback, and the Domain Score rather than following a fixed narrative role or always reducing the label margin.

Case A illustrates the trade-off among the combined costs in a fixed candidate pool, while Case B illustrates a different reference position and selection trajectory within the same condition group. Together, these examples provide concrete selection processes corresponding to the fixed-pool comparison in Appendix E.2. Aggregate results for reliability control and final mixing are reported in Appendix E.3.

D.5 Repeated Evaluation

Downstream-training randomness. For Cloth with Qwen2.5-7B, we fix the synthetic table, retained rows, and candidate selections for each configuration in Table 21. The corresponding selection and reliability-control configurations are defined in Appendices E.2–E.3.

The data split, text processing, and downstream-training configuration remain unchanged; only the downstream training seed is varied among 2026, 2027, and 2028. Each run reports the mean $R^2 \times 100$ over All-text, Fuse-late, and Fuse-early. Cross-seed standard deviations use the sample standard deviation with denominator $n - 1 = 2$.

Table 21: Repeated downstream training with fixed synthetic data. The final column reports mean $\pm$ sample standard deviation over three training seeds. Bold indicates the highest utility in each column.

Configuration	2026	2027	2028	Mean $\pm$ Std.
Full CAST	**65.63**	**65.18**	**65.99**	**65.600 $\pm$ 0.406**
CAST Round 2	65.12	64.76	65.23	65.037 $\pm$ 0.246
w/o RCS + final mixing	65.08	64.26	65.37	64.903 $\pm$ 0.576
Target-margin max. (no mixing)	63.615	62.75	62.67	63.012 $\pm$ 0.524

Table 22 reports paired differences under the same downstream-training seed.

Table 22: Paired utility differences under the same downstream-training seeds ($R^2 \times 100$). Positive values favor the configuration on the left. The final column reports mean $\pm$ sample standard deviation.

Comparison: A − B	2026	2027	2028	Mean ± Std.
CAST Round 2 − target-margin max.	+1.505	+2.010	+2.560	**+2.025 ± 0.528**
Full CAST − CAST Round 2	+0.510	+0.420	+0.760	**+0.563 ± 0.176**
Full CAST − (w/o RCS + final mixing)	+0.550	+0.920	+0.620	**+0.697 ± 0.197**

Results. Full CAST achieves the highest average utility among the four configurations for all three downstream-training seeds. The pre-mixing advantage of CAST over target-margin maximization, as well as the paired gains from final mixing and feedback reliability control, is positive under every seed. These results show that the observed selection gains remain consistent under the downstream-training randomness tested here.

This first experiment keeps the synthetic data fixed and varies only the downstream-training seed. It evaluates downstream-training randomness; the independent synthesis experiment below varies the complete synthesis realization.

Independent synthesis replicates. We evaluate three complete Full CAST synthesis realizations on Cloth with Qwen2.5-7B: the main synthesis and two independent replicates. Each realization runs structured generation, reference retrieval, shared-plan extraction, four-role candidate generation, validity filtering, CAS, and final mixing. The data split, method configuration, and downstream-training seed remain fixed across the three realizations.

Table 23: Complete Full CAST synthesis realizations on Cloth with Qwen2.5-7B. Utility is $R^2 \times 100$. Avg. averages A/L/E using full-precision scores for the main synthesis and reported one-decimal scores for the two replicates.

Synthesis Realization	Avg. WD ↓	Avg. JSD ↓	All-text	Fuse-late	Fuse-early	Avg.
Main synthesis	0.171452	0.013045	63.5	64.4	69.0	65.63
Independent replicate A	0.150772	0.012709	63.6	64.2	68.7	65.50
Independent replicate B	0.234008	0.012916	62.3	67.3	66.4	65.33

The three complete synthesis realizations yield similar average downstream utility. Their averages range from 65.33 to 65.63, with a mean and sample standard deviation of 65.49 ± 0.15. These summaries use the per-synthesis averages before display rounding, with denominator $n - 1 = 2$ for the sample standard deviation. The synthesis realization is the unit of repetition. This analysis evaluates synthesis variability on Cloth under Qwen2.5-7B with a fixed downstream-training seed; broader cross-dataset replication is left for future evaluation.

E Ablation Studies

E.1 Module Ablations and Candidate Construction

Common setup. All experiments in this section use Cloth with Qwen2.5-7B and treat `Rating` as the downstream target. The variants use the same structured data, fields, validity checks, and four-candidate budget per row. Downstream evaluation follows the common setup in Table 10 with training seed 2026. All variants use the same data-adaptive text-length rule, with the effective truncation length determined from the corresponding training data. The preset-role fallback and two-round reselection rule of Full CAST are described in Appendix B.4, and the conditional W1 metric is defined in Appendix C.2. These controls hold structured rows fixed while varying candidate construction or selection, separating the studied effects from differences in structured-data synthesis.

Module ablations. Full CAST applies SSG, CAS, and RCS and corresponds to the main result. The three ablations differ only in the operations described below.

w/o SSG (no mixing) keeps the structured input and uses the same source content and style references, but replaces the shared semantic plan and four narrative-role prompts with four independent generations from the same base prompt. No narrative or preset role is assigned. CAS initialization, reselection, and feedback reliability control are retained, while final mixing is disabled.

w/o CAS reuses the complete Full CAST candidate pool and the same valid rows and candidate sets, but replaces task- and domain-guided initialization and iterative reselection with uniform random selection within each row. Feedback-dependent reliability gating is therefore not applied. Basic validity checks and final preset-role mixing are retained.

w/o RCS also reuses the complete candidate pool and retains CAS, but removes performance gating, reliability-based weight allocation, and final mixing. Numerically valid feedback terms share the attribute-level weight W_a uniformly. Synthetic-data predictors are still refit along the selection trajectory of this variant.

Table 24: Full CAST and module ablations on Cloth with Qwen2.5-7B. Utility is $R^2 \times 100$, and Avg. averages All-text, Fuse-late, and Fuse-early. Lower Joint-MMD^2 (Gretton et al., 2012) and conditional W1 are better. Bold indicates the best value in each column; "–" denotes an unreported conditional W1.

Method	All-text	Fuse-late	Fuse-early	Avg.	Joint-MMD^2 ↓	Cond. W1 ↓
Full CAST	**63.48**	64.43	**68.99**	**65.63**	0.0084	**0.107054**
w/o SSG (no mixing)	59.36	60.91	58.79	59.69	0.0175	–
w/o CAS	61.38	65.48	66.51	64.46	0.0090	0.107900
w/o RCS	61.03	**67.28**	66.07	64.79	**0.0081**	0.107751

w/o SSG is lower than Full CAST under all three downstream strategies, with the largest gap under Fuse-early. With the complete candidate pool fixed, Full CAST improves average utility over w/o CAS by 1.17 points. The average gains from CAS and RCS mainly come from All-text and Fuse-early, while Fuse-late decreases. Full CAST also has lower conditional W1 than w/o CAS and w/o RCS. In contrast, w/o RCS has a slightly lower Joint-MMD^2 than Full CAST, showing that rankings by global distributional discrepancy and downstream utility need not agree.

Shared plan and narrative-role comparison. To separate the effects of the shared semantic plan and narrative roles, we construct four configurations A–D using these two components as factors. All configurations use the same structured rows, content-reference sources, style references, text-length targets, candidate-generation calls, sampling settings, output limits, and format-repair rules.

Configurations without a plan use the original content references directly, whereas configurations with a plan use the shared semantic plan extracted from those references. Configurations without roles independently generate four candidates from the same base prompt, whereas configurations with roles use the four narrative-role prompts. Plan-based configurations require an additional plan-extraction call; the remaining candidate-generation settings are unchanged.

Configurations A–D use the same CAS initialization, reselection rounds, and feedback reliability control, and all disable final mixing. Their synthetic predictors are refit along their own selection trajectories. Configuration A corresponds to w/o SSG (no mixing), while D corresponds to the second-round CAST selection before final mixing. Configuration E reuses the candidate pool from A but replaces feedback-based selection with uniform random selection under a fixed seed. Full CAST uses the complete plan-and-role configuration followed by final mixing.

Table 25: Controlled comparison of the shared semantic plan and narrative roles on Cloth with Qwen2.5-7B. Utility is $R^2 \times 100$, and Avg. averages the three modeling strategies. Bold indicates the best value in each column.

Configuration	Shared Plan	Role Prompts	Final Selection	All-text	Fuse-late	Fuse-early	Avg.
A: w/o SSG	No	No	CAS, no mixing	59.36	60.91	58.79	59.69
B	Yes	No	CAS, no mixing	63.63	62.87	63.78	63.43
C	No	Yes	CAS, no mixing	61.15	63.21	64.84	63.07
D: CAST Round 2	Yes	Yes	CAS, no mixing	**63.69**	**65.52**	66.15	65.12
E: random selection from A	No	No	Random, no mixing	56.17	59.19	63.94	59.77
Full CAST	Yes	Yes	CAS + final mixing	63.48	64.43	**68.99**	**65.63**

Adding the shared plan improves average utility by 3.74 points without role prompts (B − A) and by 2.05 points with role prompts (D − C). Adding role prompts improves average utility by 3.38 points without a shared plan (C − A) and by 1.69 points with a shared plan (D − B). Thus, both components contribute to candidate quality, and their joint configuration D achieves the highest utility among A–D.

Under the same no-mixing setting, D exceeds A by 5.43 points on average. The 5.94-point difference between Full CAST and A additionally includes the effect of final mixing.

On the candidate pool without SSG, CAS does not uniformly outperform random selection across downstream strategies. Compared with E, A is higher by 3.19 points on All-text and 1.72 points on Fuse-late, but lower by 5.15 points on Fuse-early, yielding a small average difference of -0.08 points. This comparison concerns the candidate pool produced without SSG. The Full CAST versus w/o CAS comparison in Table 24 instead evaluates selection on the complete CAST candidate pool with final mixing retained. The effect of final mixing on configuration D is analyzed separately in Appendix E.3.

This pattern is consistent with stronger complementarity when CAS operates on shared-plan-constrained, compositionally diverse candidates. We therefore view SSG and CAS as complementary stages whose effects depend on the candidate pool and selection configuration.

E.2 Fixed-Pool Selection and Reselection Stages

Selection rules. All comparisons reuse the same complete Full CAST candidate pool, structured rows, and retained rows. Their differences isolate the candidate-selection rule, including final mixing where specified, from text generation and structured synthesis.

Experience First-Only directly selects the Experience First candidate. *CAST Initial* uses $j_i^{(0)}$. *CAST Round 1* and *CAST Round 2* use the selections after the first and second reselection rounds, respectively. These three CAST stages come from the same full run and are evaluated before final mixing. *Full CAST* denotes the final output after row-level mixing.

We additionally compare against *target-margin maximization*, which selects the candidate that maximizes the probability margin of the target label relative to the other classes. We evaluate this rule both without mixing and with the same final mixing used by Full CAST.

Target-margin predictor. The target-margin baseline uses the real-data nominal-classification text predictor for `Rating` from the Full CAST reselection stage (TF-IDF + logistic SGDClassifier). The classifier is trained on the 15,030 real fitting records with seed 341409. Its vocabulary is the same fixed vocabulary used by Full CAST and is constructed from real fitting text and the complete candidate pool.

Let p_{real} denote the class probabilities and y_i the `Rating` of row i. For candidate $j \in \mathcal{J}_i$, define

$$m_{ij} = p_{\text{real}}(y_i \mid \mathbf{t}_{ij}) - \max_{c \neq y_i} p_{\text{real}}(c \mid \mathbf{t}_{ij}),$$
$$j_i^{\text{margin}} = \min \left\{ j \in \mathcal{J}_i : m_{ij} = \max_{k \in \mathcal{J}_i} m_{ik} \right\}. \tag{10}$$

Candidate indices are the original candidate slots, and ties are broken by the smallest slot index. This baseline performs a single row-wise selection and does not use iteration, Domain Scores, or conditional-reference distances. Although the scoring model treats `Rating` as a nominal classification target, downstream utility is still evaluated using the original rating-regression task.

Matched final mixing. The mixed target-margin baseline reuses j_i^{margin} without refitting any predictor and applies the same final-mixing rule as Full CAST. It uses the Experience First candidate, $\rho = 0.75$, the same source-row identifiers, the same stable-hash rule, and mask seed 341411. If $z_i < 0.75$, the margin-maximizing candidate is used; otherwise, the preset candidate is used. All text fields are selected together and the structured attributes remain unchanged. Every row in this candidate pool has a valid preset-role candidate, so no fallback is triggered.

Table 26: Fixed-pool selection rules and reselection stages on Cloth with Qwen2.5-7B. Utility is the average $R^2 \times 100$ over the three modeling strategies. Conditional W1 uses a fixed evaluation predictor independent of selection on the common support of 31 `Rating` × `Class Name` groups (Appendix C.2); lower values are better. Bold indicates the best reported value.

Selection / Stage	Final Mixing	Avg. Utility ↑	Cond. W1 ↓
Experience First-Only	N/A	63.5	0.114
Target-margin max.	No	63.615	0.114
Target-margin max.	Yes	62.577	0.112
CAST Initial	No	64.11	0.110
CAST Round 1	No	64.34	0.110
CAST Round 2	No	65.12	0.108
Full CAST	Yes	**65.63**	**0.107**

CAST Round 2 and Full CAST exceed the corresponding target-margin baselines by about 1.51 and 3.05 utility points, respectively. Their conditional W1 values are also lower: 0.108 versus 0.114 before mixing, and 0.107 versus 0.112 after mixing. Full CAST further exceeds Experience First-Only and CAST Initial by about 2.1 and 1.52 utility points.

After the first reselection round, CAST refits the synthetic-data text predictors on the current synthetic table, updates the relevant predictive features, scales, and reliability weights, and then performs the second reselection. Average utility increases from 64.34 to 65.12, while conditional W1 decreases from 0.110 to 0.108. Relative to the initial selection, the two reselection rounds provide a cumulative utility gain of 1.01 points. Conditional W1 is unchanged at the reported precision between the initial state and the first round.

For the target-margin baseline, final mixing lowers conditional W1 from 0.114 to 0.112 but also reduces utility by 1.038 points. Thus, the effect of mixing depends on the preceding selection rule, and a smaller measured conditional predictive-feature distance does not necessarily imply greater downstream utility.

E.3 Reliability Control and Final Mixing

We compare four combinations of feedback reliability control and final mixing while keeping the CAS selection procedure, structured rows, candidate pool, and evaluation protocol fixed.

w/o RCS disables both reliability control and final mixing. *CAST Round 2* is the pre-mixing output of the complete CAST run and retains reliability control. *w/o RCS + final mixing* restores the Full CAST preset-role and row-level mixing rule on top of the w/o RCS selection result, while keeping reliability gating and reliability-based weighting disabled. CAS and synthetic-predictor refitting are still retained. *Full CAST* uses both reliability control and final mixing.

Table 27: Reliability control and final mixing on Cloth with Qwen2.5-7B. Utility is average $R^2 \times 100$. Lower conditional W1 is better. Bold indicates the best value at the displayed precision.

Configuration	Reliability Control	Final Mixing	Avg. Utility ↑	Cond. W1 ↓
w/o RCS	No	No	64.79	0.108
CAST Round 2	Yes	No	65.12	0.108
w/o RCS + final mixing	No	Yes	65.08	**0.107**
Full CAST	Yes	Yes	**65.63**	**0.107**

Without final mixing, reliability control improves average utility from 64.79 to 65.12, a gain of 0.33 points. Under the same final-mixing rule, it improves utility from 65.08 to 65.63, a gain of 0.55 points. Final mixing improves utility by 0.29 points when reliability control is disabled and by 0.51 points when it is enabled. Full CAST therefore achieves the highest average utility among the four configurations.

At the displayed precision, both mixed configurations achieve conditional W1 of 0.107, compared with 0.108 for the two non-mixed configurations.

The per-strategy results in Table 25 further show that final mixing does not improve every downstream strategy uniformly. Relative to configuration D (CAST Round 2), Full CAST decreases All-text by 0.21 points and Fuse-late by 1.09 points, while increasing Fuse-early by 2.84 points. The average gain from final mixing therefore comes mainly from Fuse-early.

E.4 Alternative Structured-Generator Control

We compare Gaussian Copula and CTGAN (Xu et al., 2019) as structured generators on Cloth with Qwen2.5-7B. Both configurations apply the complete CAST pipeline after structured synthesis, with the same subsequent candidate-construction and selection rules. The Gaussian-Copula configuration corresponds to the main synthesis in Table 23.

Table 28: Alternative structured-generator control on Cloth with Qwen2.5-7B. Both configurations use Full CAST after structured synthesis. Utility is $R^2 \times 100$. Avg. averages A/L/E using full-precision scores for Gaussian Copula and reported one-decimal scores for CTGAN.

Structured Generator	Avg. WD ↓	Avg. JSD ↓	All-text	Fuse-late	Fuse-early	Avg.
Gaussian Copula	**0.171452**	**0.013045**	**63.5**	64.4	**69.0**	**65.63**
CTGAN	5.230125	0.152567	61.0	**66.1**	68.6	65.23

CTGAN + CAST retains similar average downstream utility despite substantially larger structured-marginal distances. Avg. WD increases from 0.171452 to 5.230125 and Avg. JSD from 0.013045 to 0.152567, while average utility changes from 65.63 to 65.23, a difference of -0.40 points computed before rounding the average utilities. This control shows that the observed utility is not tied to the particularly low marginal discrepancy of the Gaussian-Copula realization. Because this comparison varies the structured generator rather than the selection rule, selection attribution is examined separately through the fixed-row and fixed-pool controls in Appendices E.1–E.2.

F Discussion and Scope

Selection interpretation. Here, *adversarial* denotes discrete selection guided by a fixed Domain Discriminator, without alternating generator–discriminator training or a global min–max objective. Selection proceeds row by row within each round; rows interact indirectly through the synthetic-data predictor refit on the current synthetic table. For text feedback, RCS first gates terms by validation performance, then distributes a fixed attribute-level budget among valid terms. Effective reliability controls this allocation; a single valid term receives the full budget. Final mixing uses a fixed threshold (Appendix B.4). These mechanisms act as empirical safeguards against over-reliance on proxy scores, and their effects remain configuration-dependent. CAST is designed as an integrated

construction-and-composition framework in which the effects of SSG, CAS, and RCS can depend on the candidate pool and preceding selection state.

Generation fidelity and scope. Candidate generation uses shared-plan prompts and basic validity checks. The shared plan serves as a prompt-level content constraint rather than a formal entailment constraint. The four narrative roles form a compact default taxonomy; domains with specialized document structures may benefit from adapted role sets. The Gaussian-Copula generator is an implementation choice for the structured component. Appendix E.4 evaluates CTGAN as one alternative on Cloth; broader generator coverage remains open. The evaluated texts range from short free-form descriptions to paragraph-level reviews. Extensions to document-scale generation and general continuous-target settings remain future work. Because reselection uses raw target values in condition keys, continuous-target extensions would require attention to group size and reference coverage.

Privacy and reproducibility boundaries. CAST retrieves real text during synthesis. The diagnostics in Appendix D.3 measure exact and near-duplicate overlap on all three datasets. These diagnostics are empirical overlap checks rather than formal privacy guarantees, and partial copying remains possible. Reference-specific analysis is limited to Cloth, for which historical per-row reference caches are available. Sensitive or proprietary text would require additional privacy controls. Appendix D.5 distinguishes downstream-training repeats on fixed tables from the main Cloth synthesis and two independent complete-synthesis replicates. The synthesis-replicate study is limited to Cloth under one backbone and one downstream-training seed. The case studies present selected observed trajectories. Parameter sensitivity and repeated synthesis under other datasets and generators remain directions for further evaluation.