

A Novel Pipeline for Generating Realistic Synthetic CDISC ADaM Datasets Using LLMs and Knowledge Graphs

Jaime Yan and Chao Su

Merck & Co., Inc.

Agenda

1

Introduction & Motivation

Establish the need for synthetic ADaM data and outline current challenges.

2

Methodology Overview

Provide a high-level description of the four key stages in the pipeline.

3

Schema Enhancement Process

Detail the knowledge integration and structure optimization techniques used.

4

Implementation Approaches

Compare three different methods for generating synthetic data.

This presentation will guide you through each stage of the synthetic data generation pipeline, exploring how knowledge integration and structure optimization improve data quality. The agenda will be structured to provide a clear and progressive understanding of the methodology, methodology, implementation, evaluation, and future directions of this innovative approach.

Introduction

Why Synthetic ADaM Data?

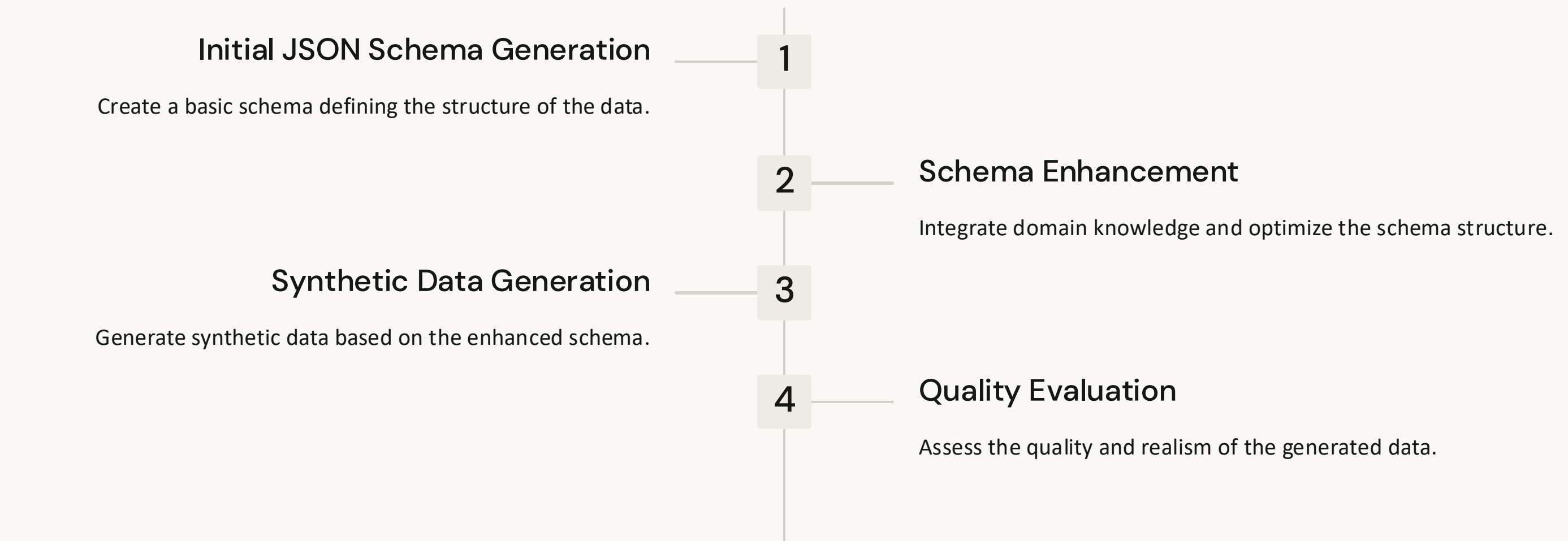
- Growing demand for data privacy protection
- Need for efficient software development processes
- Requirement for CDISC ADaM standards compliance

Current Challenges

- Complex variable relationships hard to replicate
- ADaM-specific structures difficult to preserve
- Insufficient domain knowledge incorporation

The rising concerns about data privacy, the demand for efficient software development, and the need for compliance with CDISC ADaM standards are driving the adoption of synthetic data. Traditional synthetic data generation faces challenges in replicating complex replicating complex variable relationships, preserving ADaM-specific structures, and incorporating sufficient domain knowledge. Our research knowledge. Our research addresses these challenges by introducing a novel pipeline that enhances synthetic data realism and utility. utility.

Pipeline Overview



Our pipeline consists of four key stages, starting with initial JSON schema generation, where a basic data structure is defined. The schema enhancement stage integrates domain knowledge and optimizes the data structure for ADaM compliance. Synthetic data generation then uses the enhanced schema to create realistic datasets, followed by a quality evaluation stage to assess the generated data's utility and realism. This structured approach ensures high-quality synthetic data that meets specific needs.

Json Format of Extracted Spec

Example Faker-Based Data Generation for ADSL

```
from faker import Faker
fake = Faker()

def generate_adsl(num_subjects):
    data = []
    for _ in range(num_subjects):
        record = {
            "STUDYID": fake.pystr(max_chars=8),
            "USUBJID": fake.pystr(max_chars=20),
            "AGE": fake.random_int(min=18, max=90),
            "SEX": fake.random_element(elements=("M", "F", "U"))
        }
        data.append(record)
    return data
```

Faker Code Example

Extracted Jason structure from Spec

```
{
  "Dataset Name": {
    "Structure": "str",
    "Key Variables": "str",
    "Variables": [
      {
        "VARIABLE": "str",
        "LABEL": "str",
        "DATA TYPE": "str",
        "Codelist": "NoneType",
        "Code": "str",
        "Terms": [
          "... "
        ]
      }
    ]
  }
}
```

Implementation Methods

1 Direct JSON Generation

Basic variable-level generation with limited relationship preservation.

2 Enhanced JSON Generation

Knowledge-enriched schemas for improved relationship handling.

3 Template-Based Generation

Pre-designed structural patterns centered around ADSL.

We explored three distinct implementation approaches for synthetic data generation. Direct JSON generation provides a basic variable-level generation but has limitations in relationship preservation. Enhanced JSON generation leverages knowledge-enriched schemas to improve relationship handling. Template-based generation utilizes pre-designed structural patterns centered around ADSL, offering a more structured and ADaM-compliant approach. Comparing these methods highlights trade-offs between simplicity and realism.

Schema Enhancement

Knowledge Query System

- Uses clinical trial documentation
- Builds structured knowledge graphs
- Extracts variable ranges and relationships

LLM-Based Enhancement

- Synthesizes domain knowledge
- Generates realistic value distributions
- Identifies variable relationships

The knowledge integration approach leverages both a knowledge query system and LLM-based enhancement. The query system utilizes clinical system utilizes clinical trial documentation to construct knowledge graphs, extracting variable ranges and relationships. Complementing this, LLMs synthesize domain knowledge, generating realistic value distributions and identifying intricate variable relationships. This two-pronged approach ensures a robust and comprehensive enhancement of the schema, leading to more realistic to more realistic synthetic data.

Schema Requirement

Dataset Type	Structure	Enhancement Approach
ADSL	One Record Per Subject	Ensure subject-level integrity (one record per USUBJID). Maintain hierarchical relationships between key identifiers (USUBJID, STUDYID, SITEID). Align treatment variables (TRT01P, TRT01A) and clearly structure demographic/baseline covariates (AGE, SEX, RACE).
BDS	One Record Per Parameter Per Visit	Align PARAMCD with AVAL, BASE, CHG, and other related variables within each parameter and visit. Ensure each parameter has a clear mapping to corresponding analysis variables.
OCCDS	One Record Per Occurrence	Preserve event-based relationships, ensuring each record represents a distinct occurrence (e.g., adverse events, concomitant medications). Clarify relationships between variables like AEDECOD, AESTDY, AEENDY, and severity gradings.

Schema Enhancement

Enhancement Type	Method	Implementation Details
Domain Knowledge	Knowledge Query System	Extracts structured information from clinical trial documents (protocols, SAPs, CRFs) using NLP and pattern matching
	LLM Enhancement	Generates realistic value ranges and distributions based on variable context and relationships
	Knowledge Graph	Maps relationships between variables, datasets, and domain concepts
Structure Optimization	ADSL Templates	One-record-per-subject structure with demographic and treatment patterns
	BDS Templates	Parameter-value relationships with visit patterns and change calculations
	OCCDS Templates	Event-based temporal relationships with sequence validation
Data Generation	Faker Code Generation	Direct translation of enhanced schemas to Faker code with relationship preservation
	Template Application	Specialized templates for each ADaM structure with built-in validations
	Cross-Dataset Integration	ADSL-first approach with consistent subject propagation

Schema Enhancement

Example: ADSL Schema Enhancement *Problem: The corresponding variables for the same subject might be treated as individual variables, such as the subject identifier USUBJID and STUDY, the USUBJID always contains the info from STUDY.*

JSON Structure (Before Enhancement)

```
{
  "Variables": {
    "USUBJID": { "Label": "Unique Subject Identifier", "Type": "Char", "Length": 20 },
    "SITEID": { "Label": "Site Identifier", "Type": "Char", "Length": 10 },
    "TRT01P": { "Label": "Planned Treatment", "Type": "Char", "Length": 20 }
  }
}
```



Example JSON Schema for ADSL

```
{
  "Dataset": "ADSL",
  "Structure": "ONE_RECORD_PER_SUBJECT",
  "Key Variables": ["STUDYID", "USUBJID"],
  "Variables": [
    { "Name": "STUDYID", "Type": "Char", "Length": 8 },
    { "Name": "USUBJID", "Type": "Char", "Length": 20 },
    { "Name": "AGE", "Type": "Num", "Length": 8 },
    { "Name": "SEX", "Type": "Char", "Length": 3, "Codelist": "SEX" }
  ]
}
```

Schema Enhancement

Example: BDS (Basic Data Structure) Schema Enhancement *Problem: PARAMCD is a list of codes, but there is no explicit mapping between a PARAMCD and the corresponding AVAL, BASE, CHG. The LLM cannot understand how each PARAMCD relates to different values.*

Original JSON Structure (Before Enhancement)

```
{
  "PARAMCD": { "Values": ["SYSBP", "DIABP"], "Label": "Parameter Code" },
  "AVAL": { "Label": "Analysis Value", "Type": "Num" },
  "BASE": { "Label": "Baseline Value", "Type": "Num" }
}
```

Enhanced JSON Structure (After Reorganization)

```
{
  "Dataset": "BDS",
  "Structure": "ONE_RECORD_PER_SUBJECT_PER_PARAMETER_PER_VISIT",
  "Parameters": [
    {
      "PARAMCD": "SYSBP",
      "PARAM": "Systolic Blood Pressure",
      "Analysis Variables": [
        { "Name": "AVAL", "Label": "Analysis Value", "Type": "Num" },
        { "Name": "BASE", "Label": "Baseline Value", "Type": "Num" },
        { "Name": "CHG", "Label": "Change from Baseline", "Type": "Num" }
      ]
    },
    {
      "PARAMCD": "DIABP",
      "PARAM": "Diastolic Blood Pressure",
      "Analysis Variables": [
        { "Name": "AVAL", "Label": "Analysis Value", "Type": "Num" },
        { "Name": "BASE", "Label": "Baseline Value", "Type": "Num" },
        { "Name": "CHG", "Label": "Change from Baseline", "Type": "Num" }
      ]
    }
  ]
}
```



Schema Enhancement

Example: OCCDS (Occurrence Data Structure) Schema Enhancement *Problem: The original schema did not show the occurrence relationship between the occurrence variables.*

Original JSON Structure (Before Enhancement)

```
{
  "AEDECOD": { "Label": "Adverse Event", "Type": "Char", "Length": 40 },
  "AESTDY": { "Label": "Start Day", "Type": "Num" },
  "AEENDY": { "Label": "End Day", "Type": "Num" }
}
```

Enhanced JSON Structure (After Reorganization)

```
{
  "Dataset": "OCCDS",
  "Structure": "ONE_RECORD_PER_SUBJECT_PER_OCCURRENCE",
  "Occurrence Variables": [
    { "Name": "AEDECOD", "Label": "Adverse Event", "Type": "Char", "Length": 40 },
    { "Name": "AESTDY", "Label": "Start Day", "Type": "Num" },
    { "Name": "AEENDY", "Label": "End Day", "Type": "Num" },
    { "Name": "AESER", "Label": "Serious Event Flag", "Type": "Char", "Allowed Values": ["Y", "N"] }
  ]
}
```



Schema Enhancement

ADSL

- One record per subject structure
- Demographic/baseline patterns
- Treatment variable alignment

BDS

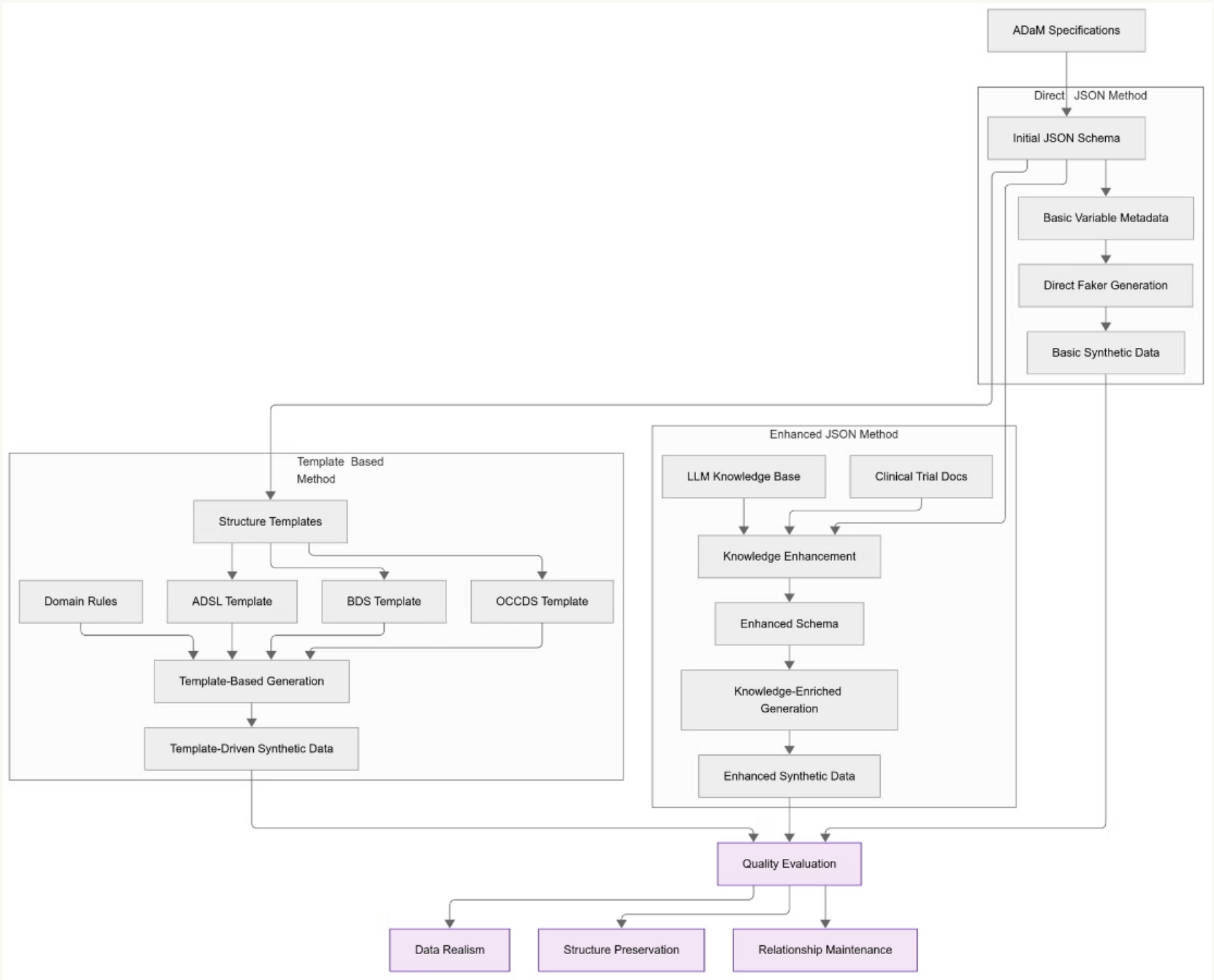
- Parameter-value relationships
- Visit pattern consistency
- Analysis variable mapping

OCCDS

- Event-based temporal relationships
- Occurrence pattern preservation
- Severity grading connections

Structure optimization is critical for different dataset types within ADaM. For ADSL, the focus is on maintaining a one-record-per-subject per-subject structure and capturing demographic/baseline patterns. For BDS, the focus is on ensuring parameter-value relationships and visit relationships and visit pattern consistency. For OCCDS, structure optimization involves preserving event-based temporal relationships and relationships and capturing occurrence patterns, including severity grading connections. Aligning these structures by dataset type significantly type significantly enhances the realism of the synthetic data.

Workflow for Synthetic ADaM Data Generation



Methods Comparison

Aspect	Direct JSON Spec	Enhanced JSON Spec	Template-Based
Knowledge Source	Basic variable metadata only	Domain knowledge + LLM suggestions	Predefined templates + Domain rules
Structural Info	Limited variable relationships	Explicit ADaM structure	Built-in structural patterns
Variable Relationships	Individual variable constraints	Cross-variable dependencies	Pre-designed relationship flows
Dataset Integration	Independent generation	Knowledge graph-based links	ADSL-centered generation
Scalability	High	Medium	Medium
Maintenance	Simple updates	Requires knowledge base updates	Template updates needed
Use Cases	Simple datasets	Complex relationships	Production environments

Evaluation Framework



Data Structure Score (DS)

Measures the fidelity of the data structure to ADaM standards.



Subject-level Score (SS)

Assesses the realism of subject-level data patterns.



Relationship Score (RS)

Evaluates the preservation of variable relationships.

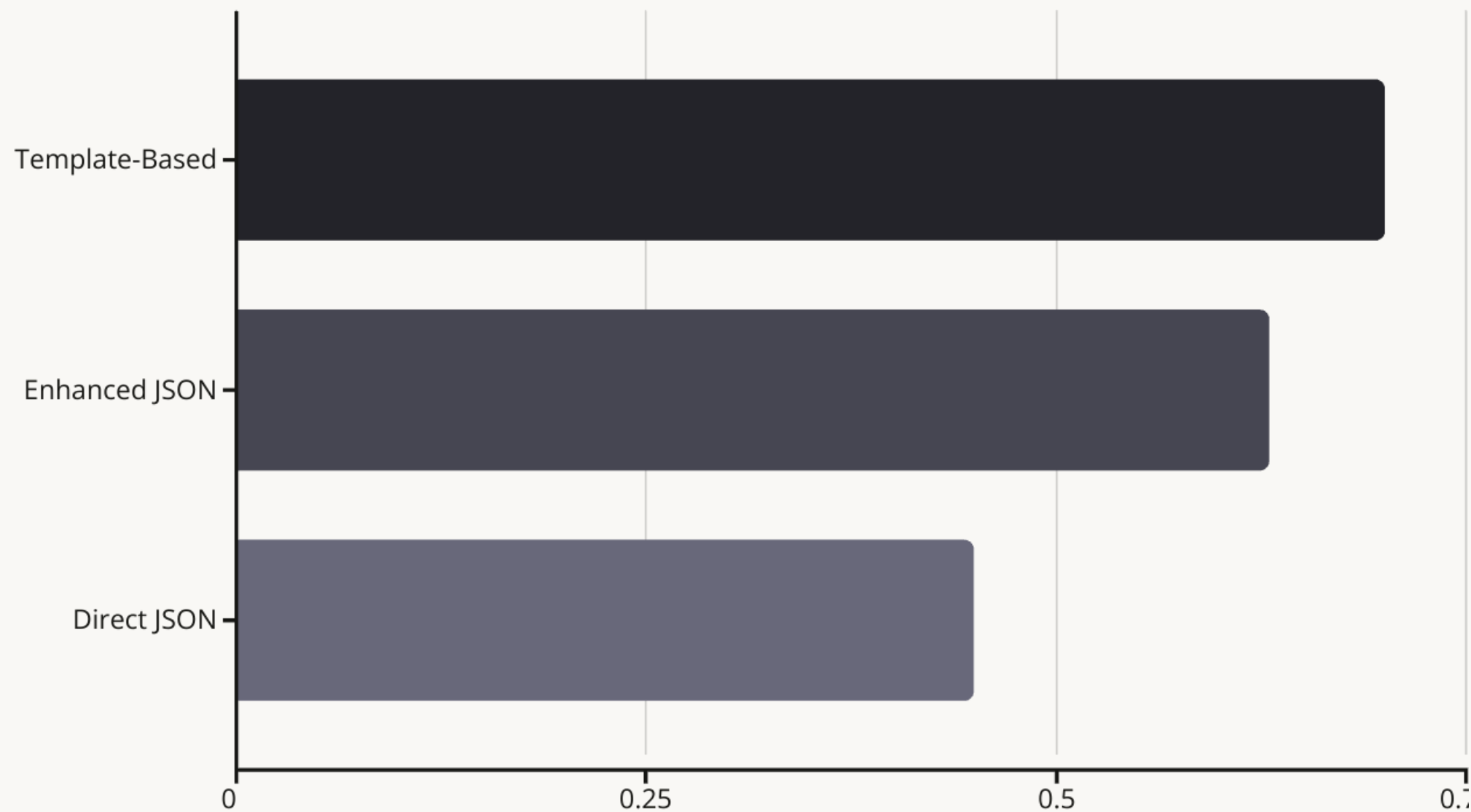
The evaluation framework utilizes three key quality metrics to assess the generated synthetic data: Data Structure Score (DS), Subject-level Subject-level Score (SS), and Relationship Score (RS). These metrics are combined into an overall quality score (Q) using weighted averaging. This weighted averaging. This framework enables a comprehensive assessment of data structure fidelity, subject-level realism, and the preservation the preservation of variable relationships, ensuring the generated synthetic data meets stringent quality standards.

Evaluation Framework

Table 4: Detailed Evaluation Metrics

Category	Specific Metrics	Measurement Method
Structure	Record count ratios	Count matching records per structure type
	Key variable presence	Verify required variables exist
	Format compliance	Check data type and length requirements
Relationships	Value dependencies	Correlation between related variables
	Cross-dataset links	USUBJID consistency across datasets
	Temporal patterns	Visit sequence validity
Data Quality	Value distributions	KS test for continuous variables
	Category proportions	Chi-square test for categorical variables
	Missing patterns	Compare missing data patterns

Results – Performance Comparison



The performance comparison demonstrates the superior quality of the template-based generation approach, achieving an overall quality score of 0.70. Enhanced JSON generation follows with a score of 0.63, while direct JSON generation scores lower at 0.45. These results highlight the effectiveness of the template-based approach in generating high-quality synthetic data by preserving ADaM-specific structures and relationships, leading to improved data utility and realism.

Detailed Analysis

Structure Preservation

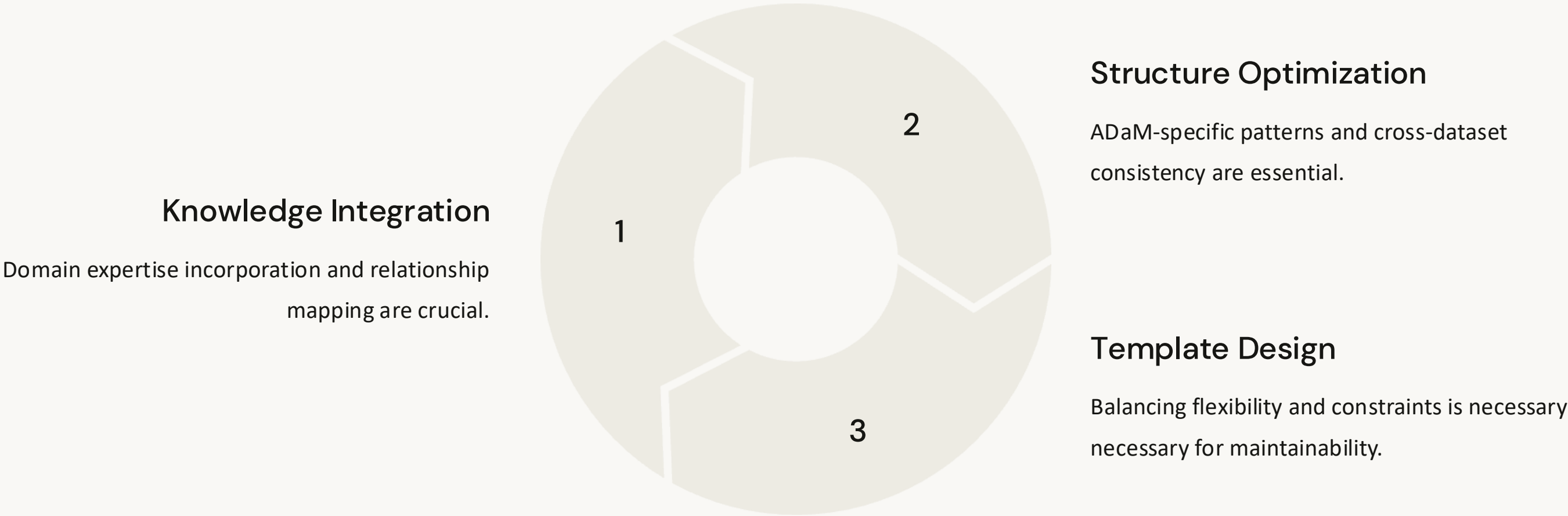
- ADSL: 98% structure maintenance
- BDS: 65-75% parameter-value validity
- OCCDS: 50-65% temporal sequence validity

Relationship Preservation

- Cross-dataset consistency: 70-75%
- Variable dependencies: 55-70%
- Temporal patterns: 45-60%

The detailed analysis reveals high structure preservation for ADSL datasets, with 98% structure maintenance. BDS datasets show 65-75% parameter-value validity, while OCCDS datasets achieve 50-65% temporal sequence validity. Relationship preservation demonstrates 70-75% cross-dataset consistency, 55-70% variable dependencies, and 45-60% temporal patterns. These results indicate the strengths and areas for improvement in the pipeline's ability to generate realistic and coherent synthetic ADaM datasets.

Implementation Considerations



Successful implementation hinges on three critical success factors: knowledge integration, structure optimization, and template design. Knowledge integration requires incorporating domain expertise and mapping variable relationships. Structure optimization necessitates adhering to ADaM-specific patterns and maintaining cross-dataset consistency. Effective template design involves balancing flexibility and constraints to ensure long-term maintenance and adaptability. By addressing these factors, organizations can maximize the benefits of synthetic data generation.



Future Directions

1

Template Automation

Automate the creation and maintenance of templates.

2

Temporal Pattern Enhancement

Improve the capture and preservation of temporal patterns.

3

Rare Event Generation

Develop methods for generating realistic rare events.

4

Cross-Dataset Validation

Enhance validation methods for cross-dataset consistency.

Future research opportunities include template automation to streamline the creation and maintenance of maintenance of templates, temporal pattern enhancement to improve the capture and preservation of preservation of temporal relationships, and rare event generation to develop methods for generating generating realistic rare events. Additional efforts should focus on cross-dataset validation to enhance enhance validation methods for ensuring cross-dataset consistency. These advancements will improve data improve data realism, expanding the utility of synthetic data.



Conclusion

1 Comprehensive Synthetic Data Pipeline

A structured approach for realistic ADaM data.

2 Novel Evaluation Metrics

Robust measures for assessing data quality.

3 Template-Based Generation Approach

Effective method for preserving ADaM structures.

4 Empirical Validation Framework

Systematic validation of synthetic data quality.

In conclusion, our research introduces a comprehensive synthetic data pipeline for generating realistic ADaM datasets, incorporating novel evaluation metrics and a template-based generation approach. The framework provides a robust method for preserving ADaM structures and relationships and a systematic approach to validate synthetic data quality. The results demonstrate the effectiveness of the proposed pipeline in improving data quality, improving data quality, enhancing development efficiency, and promoting privacy protection within the pharmaceutical industry.



Thank You

Thank you for your time and attention. For further information or inquiries, please contact:

- Jaime Yan: mingyu.yan1@merck.com
- Chao Su: chao.su@merck.com