# Agents of Diffusion: Enhancing Diffusion Language Models with Multi-Agent Reinforcement Learning for Structured Data Generation

Aja Khanal
University of Western Ontario
London, Canada
akhanal3@uwo.ca

Kaushik T. Ranade
University of Western Ontario
London, Canada
kranade@uwo.ca

Rishabh Agrawal
University of Western Ontario
London, Canada
ragrawa9@uwo.ca

Kalyan S. Basu
ICASSSD
Cambridge, Canada
ks.basu@gmail.com

Apurva Narayan
University of Western Ontario
London, Canada
apurva.narayan@uwo.ca

## ABSTRACT

Generating high-quality structured data such as JSON records, remains a fundamental challenge for large language models (LLMs), particularly when semantic richness must coexist with strict schema adherence. While autoregressive LLMs offer strong structural consistency, they often struggle with semantic variation and output diversity. In contrast, diffusion language models (DLMs) introduce powerful mechanisms for semantic richness and bidirectional decoding, yet lack the inductive biases needed for reliable structure preservation. We present **Agents of Diffusion (AoD)**, a novel framework that unifies the generative flexibility of DLMs with the reasoning capabilities of autoregressive models through language-mediated reinforcement learning. AoD frames structured text generation as a multi-agent alignment process, where a prompt optimization agent collaborates with a judge agent to iteratively guide a DLM using natural language feedback. This approach enables controllable, schema-consistent generation without modifying model parameters or relying on handcrafted constraints. AoD advances the state of controllable generation by demonstrating that diffusion models, when supervised by cooperative agents, can achieve both high semantic novelty and structural fidelity. Across multiple structured data benchmarks, AoD consistently outperforms diffusion and autoregressive baselines, establishing a new path forward for structure-aware, diversity-enhanced text synthesis. Code: https://github.com/Idsl-group/AgentsOfDiffusion. Extended Version: https://arxiv.org/abs/2601.07152

## KEYWORDS

Structured Data Generation, Synthetic Data, Multi-Agent Systems, Reinforcement Learning, Diffusion Language Models

**ACM Reference Format:**
Aja Khanal, Kaushik T. Ranade, Rishabh Agrawal, Kalyan S. Basu, and Apurva Narayan. 2026. Agents of Diffusion: Enhancing Diffusion Language Models with Multi-Agent Reinforcement Learning for Structured Data Generation.



*Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.* © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/GGJL7344

In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/GGJL7344

## 1 INTRODUCTION

**Agents of Diffusion (AoD)** is a multi-agent reinforcement learning framework for controllable data generation that pairs the semantic richness of diffusion language models (DLMs) with the structural precision of autoregressive large language model (LLM) agents. AoD explores a unique idea: use natural language feedback to supervise a DLM without fine-tuning, handcrafted rules, or scalar reward shaping. Two LLM agents (a prompt optimizer and a judge) communicate through verbal feedback to iteratively refine prompts, steering the DLM toward schema-conformant, diverse outputs.

Autoregressive LLMs are widely used in synthetic data pipelines because their inductive biases favor structure and token order [17, 31], yet these same biases can constrain diversity and trigger repetition or hallucination [18, 49]. DLMs, in contrast, generate text by iteratively denoising sequences in a non-causal, bidirectional manner [16, 26], which encourages broader semantic variation. However, they lack positional priors for format preservation, which makes them poorly suited for structure-sensitive tasks such as nested JSON synthesis [22]. AoD is designed to combine these strengths while compensating for their weaknesses.

Recent advances in prompt tuning, reinforcement learning, and agent-based coordination have improved autoregressive controllability [25, 29, 34], but comparable methods remain largely unexplored for DLMs due to their recent emergence. AoD closes this gap by enabling agentic supervision of DLMs through verbal alignment alone. Our optimization loop is parameter-free: the frozen DLM (LLaDA-8B) [27] never updates its weights, and the agents interact only through natural language, which supports interpretability and model-agnostic control. To realize this, we introduce a reinforcement learning algorithm that blends proximal policy optimization (PPO) and REINFORCE principles to optimize prompt updates using natural language feedback as a surrogate reward signal.

We evaluate AoD on four structured generation benchmarks that require semantic fluency and strict JSON schema adherence: **MultiWOZ**, **Super-NaturalInstructions**, **Self-Instruct**, and **TruthfulQA**. These datasets contain nested fields, varied schema formats,

and diverse linguistic styles, creating a challenging testbed for structure-aware DLM control. Across this suite, AoD achieves the highest **Task Success Rate** (0.79) and the lowest **Field Overlap** (0.29), outperforming diffusion and autoregressive baselines while indicating valid, non-memorized outputs.

The architecture is effective, reproducible, and accessible. AoD supports local open-weight models such as LLaMA 3.1 8B, Qwen-3 8B, DeepSeek-R1 8B, and Gemma-2 9B, as well as proprietary API-based models including GPT-4.1, GPT-4.1 Mini, and GPT-4.1 Nano. This flexibility enables operation in GPU-constrained environments and high-performance cloud settings alike. All experiments use a mix of consumer-grade hardware and API endpoints, showing that AoD does not require specialized infrastructure to produce high-quality, controllable structured generation.

**Contributions.** (1) We introduce **Agents of Diffusion**, the first multi-agent RL framework to guide DLMs using natural language.
(2) We propose an optimization loop where LLM agents iteratively refine prompts through verbal critique, achieving schema-aligned control without reward modeling.
(3) We demonstrate reproducible state-of-the-art results on JSON-based instruction synthesis across multiple structured datasets, establishing a foundation for controllable generation in symbolic, format-constrained domains.

## 2 BACKGROUND AND RELATED WORK

### 2.1 Structured Textual Synthetic Data

Synthetic data is increasingly important in machine learning, particularly when real data is limited, sensitive, or costly [2]. While LLMs have shown early success in freeform text generation [1, 21, 46], generating high-quality *structured* data such as tabular records or JSON outputs remains a major challenge [20]. LLMs often produce outputs that are syntactically correct but hallucinate and repeat outputs when required to generate data under nested structures. [39, 47]. Prior solutions relied on pipelines involving validation modules or latent modeling [9, 45], but these approaches are difficult to scale. Inspired by the success of diffusion models in vision [32, 33], recent work has explored their application to text [13, 48], though structure control remains limited. Our work addresses this gap by using a diffusion language model to enhance the semantic diversity of synthetic data, while ensuring structural fidelity through continuous evaluation in a multi-agent reinforcement learning setup.

### 2.2 Autoregressive Language Models

Autoregressive language models generate text by predicting each token sequentially, a paradigm that supports strong contextual coherence and structural alignment [4, 36, 38]. Their unidirectional decoding makes them effective for producing syntactically valid and schema-compliant outputs, such as JSON or tabular formats [19, 35]. However, their reliance on left-to-right generation often limits output diversity, reinforcing high-probability patterns and leading to generic or repetitive sequences [7]. While sampling strategies offer some relief, the inherent sequential bias of AR LMs constrains their generative flexibility [8, 28]. This motivates our exploration of diffusion models, which enable bidirectional and more diverse generation, while retaining structure through multi-agent control.

### 2.3 Diffusion Language Models

Diffusion language models (DLMs) generate text through iterative denoising, enabling more flexible and semantically diverse generation than autoregressive approaches. Diffusion-LM [23] introduced a continuous latent framework for controllable text synthesis, while LLaDA [27] extended masked-sequence diffusion to match or outperform AR baselines on reasoning and language tasks. DiffLM [50] applied discrete diffusion to structured tabular data. While these models show promise, they face key limitations: structure preservation remains brittle in constrained formats like JSON, the effects of prompt tuning on DLM behavior are poorly understood, and their role in multi-agent coordination is largely unexplored. We address these gaps by embedding a frozen DLM in a multi-agent reinforcement learning loop with autoregressive LLM agents, using natural language feedback and prompt-space optimization to enable schema-consistent, semantically aligned generation.

### 2.4 Prompt Tuning and Optimization

Prompt optimization is widely used to adapt LLMs without fine-tuning, with methods ranging from hand-crafted reasoning strategies like Chain-of-Thought [43] to automated approaches such as Promptbreeder [12] and EvoPrompt [14]. There equally exists self-rubric strategies for enhancing prompting as seen in CodecLM [42]. While effective, these methods treat prompt design as a static search problem, optimizing prompts offline without considering real-time feedback or generation dynamics [3]. This limits their adaptability, especially in tasks requiring structural precision or iterative refinement [10]. Our work reframes prompt optimization as a reinforcement learning problem, where a prompt agent learns to refine instructions based on natural language feedback. This approach is uniquely compatible with diffusion language models, whose iterative decoding allows prompts to guide generation over multiple denoising steps, enabling responsive and structure-aware control.

### 2.5 Multi-Agent Language-Based Coordination

Multi-agent systems (MAS) using LLMs have demonstrated strong performance on complex tasks by distributing reasoning across specialized, role-defined agents that communicate via natural language [15, 37]. Decentralized protocols such as critique, collaboration, or voting, enable robust, diverse solutions in domains like code synthesis and scientific discovery [11, 44]. However, language-based coordination suffers from interaction drift, inconsistent agent behavior, and weak memory retention in multi-turn exchanges [6, 30]. We address these challenges by anchoring agent interactions around a diffusion language model, whose iterative and bidirectional decoding creates a stable shared reference point. This structure reduces drift, enforces semantic consistency, and grounds prompt refinement and evaluation across dialogue turns, improving coordination without requiring external memory modules or supervision.

## 3 METHODOLOGY

### 3.1 Preliminaries

Let $\mathcal{X}$ be the space of structured text strings (JSON objects), and let $q_{\text{real}}$ denote an unknown probability distribution over $\mathcal{X}$. We observe $n$ independent and identically distributed (i.i.d.) samples

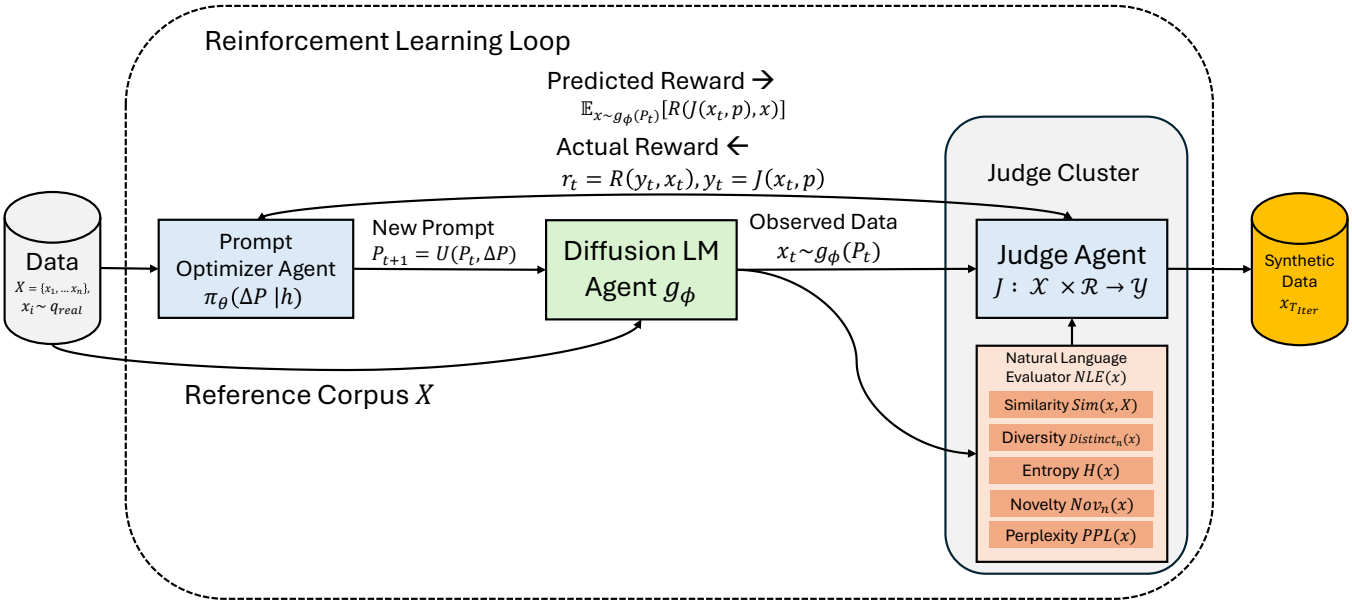


**Figure 1: Agents of Diffusion: Overview of the multi-agent training framework.**

$X = \{x_1, \dots, x_n\}$ with $x_i \sim q_{\text{real}}$. A schema $S$ defines a valid subset $\mathcal{L}(S) \subseteq \mathcal{X}$, and a binary validator $V_S : \mathcal{X} \to \{0,1\}$ returns $V_S(x) = 1$ if $x \in \mathcal{L}(S)$ and 0 otherwise. Next, let $\mathcal{P}$ be the prompt space and $\mathcal{P}_\Delta$ the set of allowable prompt edits. A diffusion language model (DLM) with parameters $\phi$ defines reverse-time Markov kernels $g_{\phi,\tau}(z_{\tau-1} \mid z_\tau, P)$ for $\tau = 1, \dots, T$ and a noise prior $\nu_T$ on a latent token space $\mathcal{Z}$ (identified with $\mathcal{X}$ after decoding). To sample $x \sim g_\phi(P)$, one draws $z_T \sim \nu_T$, then iteratively denoises via $z_{\tau-1} \sim g_{\phi,\tau}(z_{\tau-1} \mid z_\tau, P)$ until $x = z_0$. The forward corruption process, defined by kernels $k_\tau(z_\tau \mid z_{\tau-1})$, is standard and omitted here.

The framework includes four components $\mathcal{A} = \{\text{DLM}, J, R, \pi\}$. The DLM $g_\phi$ produces text samples. The judge $J : \mathcal{X} \times \mathcal{R} \to \mathcal{Y}$ maps a candidate $x$ and rubric $\rho \in \mathcal{R}$ to feedback $y = J(x, \rho) \in \mathcal{Y}$. The scorer $R : \mathcal{Y} \times \mathcal{X} \to \mathbb{R} \times \mathbb{R}^k$ returns a scalar reward $r = R(y, x)$ and a subreward vector $\mathbf{s} = R_{\text{vec}}(y, x) = (s_1, \dots, s_k)$, where each $s_i$ captures a measurable quality such as semantic similarity, diversity, novelty, perplexity, or entropy.

The prompt optimizer is a stochastic policy $\pi_\theta(\Delta P \mid h)$ parameterized by $\theta \in \Theta \subset \mathbb{R}^d$, where $\Theta$ is the set of admissible parameter vectors (e.g., neural-network weights). Given a history $h \in \mathcal{H}$ summarizing previous prompts, samples, feedback, and scores, $\pi_\theta$ outputs a distribution over edits $\Delta P \in \mathcal{P}_\Delta$. The edit operator $U : \mathcal{P} \times \mathcal{P}_\Delta \to \mathcal{P}$ then deterministically updates prompts as $P^+ = U(P, \Delta P)$. Outer optimization iterations are indexed by $t = 1, \dots, T_{\text{iter}}$, while diffusion steps are indexed by $\tau = 1, \dots, T$.

Finally, for any random variable $Z$, $\mathbb{E}[Z]$ denotes expectation with respect to the randomness from $q_{\text{real}}$, the DLM sampling process $g_\phi$, and the policy $\pi_\theta$, unless specified otherwise.

## 3.2 Problem Formulation

As shown in Figure 1, AoD formalizes the synthesis of textual JSON data as a controllable structured generation problem within a multi-agent reinforcement learning framework. The objective is to integrate the semantic diversity of diffusion language models with the structural precision of autoregressive agents. In this setup, a diffusion model proposes diverse JSON candidates, while two LLM agents iteratively refine the conditioning prompts through natural-language feedback to ensure schema conformity and semantic consistency. This cooperative feedback loop enables controllable generation of structured text without modifying model parameters.

At each outer iteration $t \in \{1, \dots, T_{iter}\}$, the system maintains a prompt $P_t \in \mathcal{P}$ and a summary history $h_t \in \mathcal{H}$. The DLM $g_\phi$ generates a structured candidate $x_t \sim g_\phi(P_t)$. The judge $J$ evaluates $x_t$ under rubric $\rho \in \mathcal{R}$, producing feedback $y_t = J(x_t, \rho)$, and the scorer $R$ converts this into numerical signals $r_t = R(y_t, x_t)$ and $\mathbf{s}_t = R_{vec}(y_t, x_t)$. The prompt optimizer, parameterized by $\theta$, samples an edit $\Delta P_t \sim \pi_\theta(\Delta P \mid h_t)$ and updates the prompt via

$$P_{t+1} = U(P_t, \Delta P_t). \tag{1}$$

This defines an episodic Markov decision process (MDP) with state $z_t = (P_t, h_t)$, action $a_t = \Delta P_t$, transition kernel induced by $(g_\phi, J, R, U)$, and reward $r_t$. The policy parameters $\theta$ are optimized to maximize the expected discounted return:

$$\max_\theta \mathbb{E}\left[\sum_{t=1}^{T_{iter}} \gamma^{t-1} r_t\right] \tag{2}$$

subject to the resource constraints $Tokens \le B_{tok}$, $Calls \le B_{calls}$, $T_{iter} \le B_{iter}$, $\gamma \in [0, 1)$.

**Algorithm 1** Multi-Agent Reinforcement Learning Loop in Agents of Diffusion (AoD)

**Input:** schema $S$, rubric $\rho$, diffusion model $g_\phi$, policy $\pi_\theta$, edit operator $U$
**Initialize:** prompt $P_1 \in \mathcal{P}$, summary history $h_1 \in \mathcal{H}$
**for** $t = 1$ **to** $T_{iter}$ **do**
  Sample structured candidate $x_t \sim g_\phi(P_t)$
  Judge provides feedback $y_t = J(x_t, \rho)$
  Compute reward $r_t = R(y_t, x_t)$ and subrewards $\mathbf{s}_t = R_{vec}(y_t, x_t)$
  Update history summary $h_t = f(P_t, x_t, y_t, \mathbf{s}_t)$
  Sample prompt edit $\Delta P_t \sim \pi_\theta(\Delta P \mid h_t)$
  Apply edit: $P_{t+1} = U(P_t, \Delta P_t)$
  Update policy parameters: $\theta \leftarrow \theta + \eta \widehat{\nabla}_\theta \mathbb{E}_{\pi_\theta}[r_t]$
**end for**
**Return:** final prompt $P_{T_{iter}}$, final candidate $x_{T_{iter}}$

At convergence ($T_{iter}$), performance is evaluated using terminal objectives that capture structural validity, semantic relevance, and diversity:

$$\max_\theta \ \alpha\,\mathbb{E}[V_S(x_{T_{iter}})] + \beta\,\mathbb{E}[Sim(x_{T_{iter}}, X)] + \delta\,\mathbb{E}[Distinct_n(x_{T_{iter}})],$$

or equivalently in constrained form,

$$\max_\theta \ \mathbb{E}[Distinct_n(x_{T_{iter}})] \tag{3}$$

such that $\mathbb{E}[V_S(x_{T_{iter}})] \geq \tau_{valid},\ \mathbb{E}[Sim(x_{T_{iter}}, X)] \geq \tau_{sim}$.

The diffusion parameters $\phi$ remain fixed; controllability arises solely through the autoregressive policy $\pi_\theta$ acting on prompts.

## 3.3 DLM Integrated Multi-Agent RL

*Prompt Optimization Agent.* The prompt optimization agent governs controllability in text-based structured JSON synthesis by steering the diffusion language model toward schema-valid, semantically coherent generations. It is instantiated as an autoregressive large language model because prompt edits are sequential and token-dependent, making autoregressive architectures ideal for learning discrete edit trajectories in text space. The agent defines a stochastic policy $\pi_\theta(\Delta P \mid h)$ parameterized by $\theta$, generating contextually guided prompt updates that modulate the conditional output distribution of $g_\phi$. Its objective is to maximize the expected reward provided by the judge–scorer pair $(J, R)$:

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{x \sim g_\phi(P)}[\, R(J(x, \rho), x)\,], \tag{4}$$

where $\rho$ is a task-specific rubric capturing schema and semantic fidelity. Since $g_\phi$ is non-differentiable, $\pi_\theta$ serves as a surrogate functional optimizer that approximates the gradient of $\mathcal{T}(P) = \mathbb{E}_{x \sim g_\phi(P)}[R(J(x, \rho), x)]$ through discrete, language-conditioned updates rather than backpropagation.

Theorem 3.1. *Let $\mathcal{T}(P) = \mathbb{E}_{x \sim g_\phi(P)}[R(J(x, \rho), x)]$ as in (4). If $\mathcal{T}(P)$ is locally Lipschitz and prompt edits $\Delta P_t$ sampled from $\pi_\theta(\Delta P \mid h_t)$ are bounded, then the iterative update $P_{t+1} = U(P_t, \Delta P_t)$ constitutes a contraction mapping in expectation for sufficiently small step size. Thus, the sequence $\{P_t\}$ converges to a fixed point $P^*$ satisfying*

$$\mathcal{T}(P^*) = \max_P \mathcal{T}(P),$$

*ensuring stable convergence toward reward-aligned, schema-consistent prompts.*

This agent operationalizes discrete autoregressive reasoning as a control layer over diffusion dynamics, transforming natural-language feedback into token-level schema alignment steps. For instance, under a booking schema $S$ with fields `departure_city`, `arrival_city`, and `date`, an initial prompt such as "Generate travel details" may elicit feedback like "Use JSON format and include all fields." Through iterative updates $\Delta P_t$, the agent refines this into `"Generate a JSON object with fields {origin, destination, date} using YYYY-MM-DD format."` Each refinement incrementally aligns the DLM's conditional distribution $p_\phi(x \mid P_t)$ with the valid schema subset $\mathcal{L}(S)$ while preserving semantic diversity. Theoretically, this mechanism bridges discrete symbolic reasoning and stochastic generation, enabling reinforcement-driven adaptation in structured text synthesis.

*Diffusion Language Model Agent.* The diffusion language model (DLM) $g_\phi(z_{\tau-1} \mid z_\tau, P)$ serves as the generative backbone for synthesizing text-based structured JSON data. It defines a reverse-time Markov process that reconstructs text tokens from gradually denoised latent representations, yielding the conditional distribution $p_\phi(x \mid P)$. Unlike autoregressive models that factorize $p(x \mid P)$ sequentially as $\prod_i p(x_i \mid x_{<i}, P)$, the DLM estimates $p_\phi(x \mid P)$ implicitly through iterative denoising. This non-causal, bidirectional formulation allows each denoising step to condition on global context rather than local token dependencies, resulting in broader coverage of valid schema-conformant configurations.

Proposition 3.2. *Let $p_\phi(x \mid P)$ and $p_{AR}(x \mid P)$ denote diffusion and autoregressive conditional distributions trained on the same structured dataset with schema $S$. If both minimize divergence from the real data distribution $q_{\text{real}}(x)$ under bounded reconstruction error and finite diffusion horizon $T$, then*

$$\mathrm{KL}(q_{\text{real}}(x) \,\|\, p_\phi(x \mid P)) \leq \mathrm{KL}(q_{\text{real}}(x) \,\|\, p_{AR}(x \mid P)),$$

*indicating that diffusion better approximates the real data manifold and captures a wider set of semantically valid configurations.*

Theoretically, this property positions the DLM as the diversity-preserving agent within AoD. It expands the support of $p_\phi(x \mid P)$ across multiple valid schema realizations, enabling the synthesis of varied yet coherent JSON structures. The autoregressive agents, in contrast, provide the constraint mechanism that ensures syntactic and semantic adherence to $\mathcal{L}(S)$. Together, they form a complementary system: diffusion drives diversity, while autoregression enforces structure. This achieves controlled, schema-aligned generation of structured text data.

*Judge Agent.* The judge cluster combines the LLM-based judge $J$ and the Natural Language Evaluator (NLE), forming the evaluation subsystem responsible for interpreting and supervising outputs from the diffusion language model $g_\phi$. The NLE receives each generated JSON sample $x \in \mathcal{X}$ and computes five quantitative metrics defined in the preliminaries: semantic similarity $\mathrm{Sim}(x, X)$, diversity Distinct-$n(x)$, entropy $H(x)$, novelty Nov-$n(x)$, and perplexity $\mathrm{PPL}(x)$. These metrics jointly describe how well $x$ aligns with the reference dataset $X$, how varied and fluent it is, and whether it generalizes beyond seen examples. The NLE then converts these

numeric values into structured natural language statements, which summarize deviations and attributions in an interpretable form. The LLM judge $J$, instantiated as an autoregressive model, consumes this structured textual feedback together with the rubric $\rho$ and evaluates each sample through a fixed set of rubric-aligned yes/no questions (e.g., "Is the JSON structurally complete?", "Are all required fields present?", "Is the text semantically faithful?"). Based on both the quantitative assessments and its own contextual reasoning, $J$ generates the final critique $y = J(x, \rho)$, which the scorer translates into a scalar reward $r = R(y, x)$ and subreward vector $\mathbf{s} = R_{\text{vec}}(y, x)$. For example, the cluster might produce: "*The JSON is fluent but missing* `arrival_city`*; the date format should be YYYY-MM-DD.*" This feedback then guides the prompt optimization policy $\pi_\theta(\Delta P \mid h)$, completing the reinforcement loop.

Correspondingly, the NLE isolates measurable properties of the generated text, providing low-variance, disentangled feedback signals that prevent noisy gradients and improve the reliability of downstream optimization. The LLM judge $J$ transforms these discrete measurements into a smooth, natural-language surrogate of the underlying reward landscape, enabling gradient-free optimization while maintaining semantic transparency. The autoregressive formulation of $J$ is particularly important for structured JSON synthesis, as evaluating conformance, field ordering, and key dependencies requires sequential reasoning over tokens. By processing feedback in a left-to-right manner, $J$ preserves causal consistency in its critiques and ensures alignment with how $\pi_\theta$ performs token-level edits to $P$. Thus, autoregressive reasoning enforces structured coherence while maintaining semantic flexibility, allowing the system to generalize across diverse schemas without manual rule engineering.

Theoretically, the cluster defines an expected feedback operator

$$\mathcal{T}(P) = \mathbb{E}_{x \sim g_\phi(P)}[R(J(x, \rho), x)] \tag{5}$$

that stabilizes learning in the non-differentiable environment $(g_\phi, J, R)$. The NLE grounds $\mathcal{T}(P)$ in verifiable quantitative signals, while the autoregressive judge smooths discontinuities by mapping discrete validation outcomes to continuous linguistic explanations. This synergy reduces variance in the policy-gradient estimate, improves credit assignment, and enforces reward monotonicity with respect to schema-conformant and semantically faithful outputs. Together, they form a theoretically consistent bridge between numeric supervision and symbolic prompt control, balancing the structured precision of autoregressive models with the generative diversity of diffusion-based synthesis.

Proposition 3.3. *Let $\mathcal{T}(P) = \mathbb{E}_{x \sim g_\phi(P)}[R(J(x, \rho), x)]$. Suppose each component in $R_{vec}(y, x)$ is bounded, the mapping from the NLE's metric vector to textual feedback $y = J(x, \rho)$ is Lipschitz with constant $L_J$ with respect to* $(\text{Sim}(x, X), \text{Distinct-}n(x), H(x), \text{Nov-}n(x), \text{PPL}(x))$, *and $R$ is monotone in these components. Then $\mathcal{T}(P)$ is locally Lipschitz in $P$ and preserves ordering with respect to semantic fidelity: if $P_1, P_2$ induce samples such that $\mathbb{E}[\text{Sim}(x, X) \mid P_1] > \mathbb{E}[\text{Sim}(x, X) \mid P_2]$, then $\mathcal{T}(P_1) > \mathcal{T}(P_2)$. Consequently, policy updates driven by $\mathcal{T}(P)$ are stable in expectation and prioritize semantically faithful prompts.*

# 4 EXPERIMENTS AND RESULTS

## 4.1 Experimental Setup

*Hardware.* To demonstrate the accessibility and reproducibility of AoD, all experiments were run on a consumer-grade workstation with an AMD Ryzen 9 7900X (12-core, 24-thread, 4.7 GHz base), 32 GB DDR5 RAM, and an NVIDIA RTX 4080 SUPER GPU (16 GB VRAM). This setup reflects hardware that is widely available to individual researchers and developers, ensuring that AoD does not rely on specialized infrastructure or large-scale compute clusters.

*Models.* Our experiments use both open-source and API-based language models to highlight AoD's flexibility and hardware independence. Eight autoregressive models were used for the prompt optimizer and LLM judge roles: **LLaMA-3.1 8B** (32 layers, 40 heads, 4-bit quantization; temperature 0.7 for prompting, 0.2 for judgment), **Qwen-3 8B** (multilingual, LoRA-enabled, nucleus sampling with $p = 0.9$), **DeepSeek-R1 8B** (NTK-aware, top-$k$ sampling with $k = 40$), **Gemma-2 9B** (beam search width 3), **Mistral 7B** (grouped-query attention, 8-bit decoding), and three API-based models: **GPT-4.1 Nano**, **Mini**, and **GPT-4.1**, which support lightweight to high-fidelity generation. All experiments used the same autoregressive model for both the prompt optimizer and LLM judge for consistency and computational simplicity. The generator agent, by contrast, was exclusively **LLaDA 8B**, a discrete diffusion language model with 32 layers, sinusoidal embeddings, 1024-token input, and $T = 12$ denoising steps. LLaDA operates in FP16 mode and disables classifier-free guidance to preserve sampling diversity. Across both local and API models, AoD demonstrates consistent, high-quality performance without reliance on specialized hardware.

*Datasets.* We evaluate our framework to synthesize JSON data on four publicly available datasets chosen for their diversity in structure, semantics, and generation objectives. **MultiWOZ** [5] is a multi-domain dialogue dataset with rich slot-filling annotations, useful for testing structure preservation and schema alignment. **Super-NaturalInstructions** [41] contains diverse instruction-response pairs across hundreds of tasks, enabling generalization over prompt types. **TruthfulQA** [24] provides factuality-challenging questions, useful for evaluating hallucination and semantic precision. **Self-Instruct** [40] consists of instruction-based examples distilled from LLMs, supporting experiments on prompt-response synthesis in alignment-critical tasks. Each dataset is randomly subsampled prior to training and evaluation to reduce computational overhead and mitigate memorization. By exposing the model to only a small, randomly selected portion of the full dataset, we minimize the risk of copying specific examples during generation and ensure that performance reflects generalization to unseen instructions.

*Baselines.* We compare AoD against six complementary baselines that represent the main paradigms of structured text generation. **Diffusion-LM** [23] and **DiffLM** [50] are diffusion-based models emphasizing semantic diversity and multimodal coverage, serving as diversity-oriented references. **CodecLM** [42], **PromptBreeder** [12], and **EvoPrompt** [14] are autoregressive prompt-optimization methods that provide strong control-oriented baselines, reflecting existing strategies for structured prompting and

language-driven refinement. Finally, **UniGen** [45] embodies validation-based synthesis, enforcing symbolic and schema-level constraints at generation time. Together, these six baselines span the design space of *structure control* (AR-based), *diversity* (diffusion-based), and *constraint enforcement* (validation-based), providing a comprehensive comparison framework for AoD's contribution: unifying all three through a multi-agent reinforcement-learning mechanism that produces schema-conformant yet semantically diverse structured JSON data.

*Evaluation Metrics.* Our evaluation framework separates metrics used for agent feedback from those used for independent verification. During training, the judge agent leverages five grounded metrics: perplexity $\mathrm{PPL}(x)$, semantic similarity $\mathrm{Sim}(x, X)$, diversity Distinct-$n(x)$, token entropy $H(x)$, and novelty Nov-$n(x)$, to generate interpretable natural language feedback for the prompt optimizer. These metrics provide structural and semantic supervision without exposing scalar reward values directly, thereby avoiding reinforcement bias or overfitting to numeric targets. However, as a measure for independent evaluation, we report standard text-based quality metrics including BLEU, ROUGE, and METEOR to quantify syntactic and lexical correspondence between generated and reference samples. This ensures that our reported performance reflects true generation quality rather than reinforcement feedback bias. Furthermore, to assess the downstream finetunability and functional reliability of the synthesized data, we also compute the *Task Success Rate* (TSR), which measures the proportion of valid, semantically consistent, and diverse generations meeting all constraints.

*Memorization and Collusion Verification.* The five reward metrics used during training (perplexity $\mathrm{PPL}(x)$, semantic similarity $\mathrm{Sim}(x, X)$, diversity Distinct-$n(x)$, token entropy $H(x)$, and novelty Nov-$n(x)$) jointly regulate memorization and collusion within the multi-agent reinforcement learning loop. Each metric enforces a distinct behavioral constraint: $\mathrm{PPL}(x)$ ensures linguistic coherence and penalizes degenerate text, $\mathrm{Sim}(x, X)$ promotes semantic alignment, Distinct-$n(x)$ and $H(x)$ encourage lexical variability, and Nov-$n(x)$ penalizes verbatim reuse of the reference dataset $X$. The judge uses these quantitative signals to answer a fixed set of rubric-based yes/no questions, ensuring that reinforcement is grounded in objective structure and meaning rather than hidden coordination between agents. Because the prompt optimizer $\pi_\theta$ never directly observes scalar rewards but instead receives natural language feedback derived from them, it cannot exploit the reward function through collusion or memorization. Numeric trends across iterations provide diagnostic signals, such as a simultaneous increase in $\mathrm{Sim}(x, X)$ and decrease in Nov-$n(x)$, which explicitly reveal potential leakage or overfitting.

To further ensure independence and verify that generated data remains distinct from the training corpus, we introduce the *Field Overlap* metric as a post-hoc measure. Field Overlap computes the proportion of key–value pairs or fields in generated JSON samples that exactly match those in the reference set. High overlap values indicate potential copying or memorization, while low overlap combined with low $\mathrm{PPL}(x)$ suggests faithful generalization with coherent generation. Unlike $\mathrm{Sim}(x, X)$, which captures semantic similarity, Field Overlap explicitly measures structural duplication, making it a direct test for memorization or cross-agent information leakage. Together, these signals enable both in-loop and independent verification of data novelty, ensuring that AoD produces diverse, semantically faithful, and unbiased synthetic JSON data.

## 4.2 Discussion

Table 1 highlights that AoD achieves a rare balance between structural precision and generative diversity, outperforming both diffusion-based and autoregressive systems. High Similarity (0.88) combined with strong Diversity (0.72) and Novelty (0.83) demonstrates that AoD generates data that remains semantically faithful while exploring new schema-consistent configurations. The Entropy score (6.10) indicates balanced lexical richness rather than repetitive phrasing, and the low Perplexity (22.1) confirms fluent and coherent language modeling. This pattern is not coincidental; it directly reflects the balance between exploration and regulation within AoD's architecture. The diffusion generator introduces stochastic breadth, while the judge cluster applies linguistic and structural constraints that stabilize the output space. Reinforcement-guided optimization aligns these opposing forces, producing samples that are both creative and compositionally valid, even in datasets with complex nested structures.

Independent metrics reinforce this interpretation. AoD leads on BLEU, ROUGE-L, and METEOR, showing that its diversity does not compromise grammatical or semantic fidelity. The high Task Success Rate (0.79) indicates that generated records satisfy both content and structure requirements, while the lowest Field Overlap (0.29) confirms that AoD avoids memorization by generating distinct key-value combinations unseen in training. These numerical patterns arise from the multi-agent reward structure: the natural language evaluator (NLE) introduces interpretive continuity by converting discrete metric signals into graded linguistic feedback, while the autoregressive judge performs sequential validation across keys and fields. This layered supervision stabilizes reward propagation, making optimization smoother and preventing overfitting to numeric heuristics. The result is visible in the metrics—Similarity and Perplexity improve simultaneously, Diversity and Novelty rise without structural drift, and Entropy remains high yet coherent. AoD's reinforcement signals therefore encode both form and meaning, producing generalization that persists across unseen distributions.

These dynamics also clarify why AoD achieves an uncommon combination of low Perplexity and high Entropy. Traditional diffusion systems increase diversity but often generate syntactically unstable text, while autoregressive systems enforce structure at the cost of variability. AoD bridges this divide by coupling diffusion-driven exploration with sequential constraint verification. The NLE's linguistic grounding allows lexical and semantic expansion to occur in a controlled way, and the judge's autoregressive reasoning enforces causal dependencies between fields. Together, they produce the observed equilibrium: high-entropy text that remains syntactically fluent and semantically consistent. This mechanism explains why AoD avoids the typical diffusion drift toward incoherence and the autoregressive bias toward repetition. The differences among baseline models further contextualize AoD's superiority. Static autoregressive systems such as LLaMA and Qwen prioritize conditional likelihood maximization, maintaining high Similarity

**Table 1: Comparison of AoD with baselines, The first five metrics correspond to those used in training, while the last five serve as independent evaluation metrics. Higher is better for all metrics except Perplexity and Field Overlap. Values averaged across all datasets and Prompt Optimizer + Judge LLM pairs for AoD. Each experiment was repeated 15 times.**

| Model | Similarity | Diversity | Novelty | Entropy | Perplexity | BLEU | ROUGE-L | METEOR | TSR | Field Overlap |
|---|---|---|---|---|---|---|---|---|---|---|
| *Static Autoregressive Baselines (single-pass prompting)* | | | | | | | | | | |
| LLaMA-3.1 8B | 0.86 | 0.42 | 0.48 | 5.18 | 21.6 | 33.9 | 38.1 | 27.9 | 0.71 | 0.38 |
| Qwen-3 8B | 0.85 | 0.44 | 0.50 | 5.22 | 22.3 | 34.1 | 37.6 | 27.8 | 0.70 | 0.36 |
| DeepSeek-R1 8B | 0.87 | 0.41 | 0.47 | 5.14 | 20.9 | 35.2 | 38.8 | 28.0 | 0.73 | 0.35 |
| Gemma-2 9B | 0.84 | 0.43 | 0.49 | 5.19 | 22.0 | 34.5 | 37.8 | 28.1 | 0.72 | 0.37 |
| Mistral 7B | 0.83 | 0.44 | 0.48 | 5.25 | 23.2 | 33.7 | 37.2 | 27.5 | 0.69 | 0.39 |
| GPT-4.1 Nano | 0.84 | 0.46 | 0.55 | 5.16 | 21.9 | 30.5 | 36.0 | 26.2 | 0.66 | 0.38 |
| GPT-4.1 Mini | 0.84 | 0.47 | 0.56 | 5.12 | 21.5 | 30.9 | 36.5 | 26.4 | 0.67 | 0.37 |
| GPT-4.1 | 0.85 | 0.48 | 0.56 | 5.10 | 21.2 | 31.0 | 36.8 | 26.6 | 0.68 | 0.37 |
| *Diffusion and Prompt-Based Baselines* | | | | | | | | | | |
| Diffusion-LM [23] | 0.72 | 0.60 | 0.72 | 5.82 | 29.4 | 28.1 | 33.5 | 25.1 | 0.61 | 0.42 |
| DiffLM [50] | 0.74 | 0.63 | 0.70 | 5.90 | 28.6 | 27.5 | 32.9 | 24.6 | 0.63 | 0.41 |
| UniGen [45] | 0.78 | 0.52 | 0.63 | 5.64 | 27.5 | 30.8 | 35.0 | 26.0 | 0.67 | 0.40 |
| PromptBreeder [12] | 0.80 | 0.51 | 0.59 | 5.51 | 25.7 | 31.2 | 36.7 | 26.5 | 0.68 | 0.38 |
| EvoPrompt [14] | 0.81 | 0.49 | 0.57 | 5.48 | 25.1 | 32.4 | 37.0 | 27.0 | 0.70 | 0.37 |
| CodecLM [42] | 0.82 | 0.47 | 0.56 | 5.42 | 24.8 | 33.0 | 37.5 | 27.3 | 0.71 | 0.36 |
| LLaDA [27] | 0.79 | 0.69 | 0.81 | 6.03 | 27.0 | 29.5 | 34.2 | 25.8 | 0.69 | 0.35 |
| **AoD (ours)** | **0.88** | **0.82** | **0.83** | **6.10** | **22.1** | **35.6** | **40.1** | **29.3** | **0.79** | **0.29** |

but collapsing on Diversity and Novelty due to deterministic decoding. Diffusion baselines like DiffLM or Diffusion-LM invert this pattern, producing diverse but structurally fragile data because their denoising trajectories lack schema-aware conditioning. Prompt-evolution frameworks like EvoPrompt and PromptBreeder improve variability through heuristic mutation but fail to sustain progress because they lack credit assignment across sequential edits. In contrast, AoD closes this optimization loop through dynamic prompt refinement driven by multi-dimensional feedback from the judge cluster. The prompt optimizer learns a policy that adjusts prompts not only based on reward magnitude but also on the linguistic context of errors, enabling continual improvement across iterations.

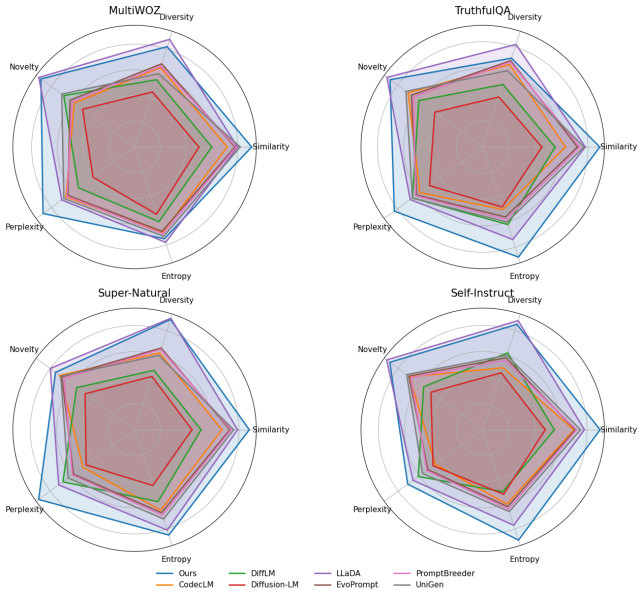


**Figure 2: Comparison of normalized metrics across datasets.**

Figure 2 illustrates AoD's consistency across datasets, where its polygons expand uniformly along all metric axes. In structurally complex domains like *MultiWOZ* and *Super-Natural*, AoD sustains high Similarity and low Perplexity while widening Diversity and Entropy, demonstrating its ability to preserve schema integrity while encouraging lexical variation. On reasoning-heavy datasets such as *TruthfulQA* and *Self-Instruct*, AoD balances Novelty and Similarity, showing that the judge cluster and diffusion generator collaboratively regulate generalization without overfitting. Competing methods either favor diversity at the expense of structure (Diffusion-LM, DiffLM) or maintain structure but exhibit reduced novelty (PromptBreeder, EvoPrompt). The near-regular shape of AoD's region across all four datasets confirms stable performance and adaptability to different data modalities, reinforcing its role as a schema-faithful yet diverse structured data generator.

## 4.3 Ablation Study

*Agentic and Reward Ablation.* Figure 3 illustrates the impact of each agentic component and feedback mechanism within AoD. Introducing the autoregressive prompt optimizer (PO) leads to improvements across all metrics. This indicates that prompt refinement, even without reinforcement, enhances fluency and schema alignment by allowing structured edits over iterations. However, Diversity and Entropy remain constrained since optimization is deterministic and lacks stochastic exploration. Adding reinforcement learning with scalar rewards (*FRL–AR–S*) further enhances Similarity and lowers Perplexity, as scalar feedback enables structured reward shaping for grammatical and syntactic correctness. Yet, gains in Diversity and Entropy are limited by the autoregressive generator's decoding bias, which narrows the output space to high-likelihood continuations. Replacing scalar rewards with natural language (*FRL–AR–NL*) yields smoother reward propagation and improved fluency, as the judge's interpretable feedback offers token-level supervision. Nevertheless, the sequential nature of the autoregressive generator continues to limit the attainable variety.

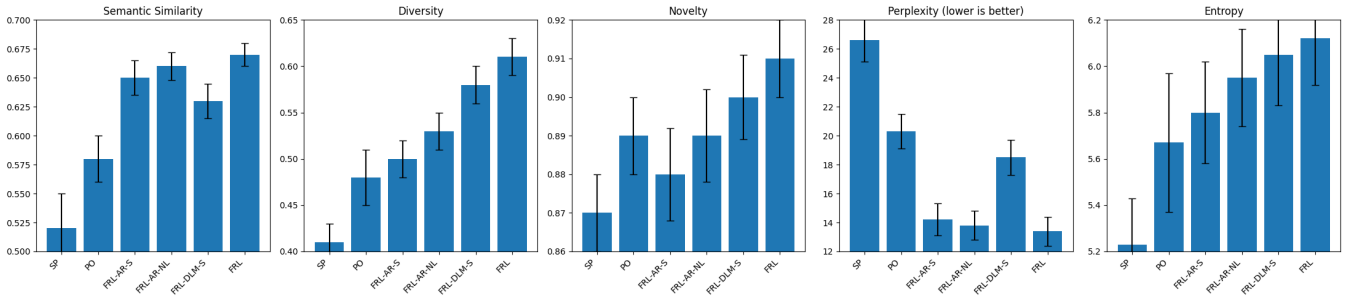


**Figure 3: Ablation of agents, averaged across datasets and Optimizer–Judge pairs. SP = Static Prompt, PO = Prompt Optimizer, FRL–AR–S = Autoregressive generator with scalar rewards, FRL–AR–NL = Autoregressive generator with natural language feedback, FRL–DLM–S = Diffusion generator with scalar rewards, FRL = Diffusion generator with natural language feedback.**

**Table 2: Transferability for prompt optimizer (rows) and judge (columns). Reports Diversity (↑) and GPU runtime seconds (↓) averaged for all datasets. LLaDA is fixed generator. L= LLaMA, Q = Qwen, D = DeepSeek, G = Gemma, M = Mistral.**

| | L | | Q | | D | | G | | M | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Opt.** | Div. | Rt. | Div. | Rt. | Div. | Rt. | Div. | Rt. | Div. | Rt. |
| L | 0.83 | 112 | 0.81 | 110 | 0.80 | 115 | 0.78 | 117 | 0.82 | 113 |
| Q | 0.82 | 118 | 0.85 | 119 | 0.84 | 116 | 0.79 | 118 | 0.80 | 121 |
| D | 0.80 | 115 | 0.83 | 117 | 0.86 | 114 | 0.82 | 118 | 0.81 | 120 |
| G | 0.79 | 120 | 0.81 | 121 | 0.82 | 122 | 0.84 | 118 | 0.83 | 119 |
| M | 0.82 | 111 | 0.83 | 113 | 0.81 | 114 | 0.79 | 115 | 0.84 | 112 |

Transitioning to a diffusion generator while maintaining scalar rewards (*FRL–DLM–S*) produces an increase in Diversity, Novelty, and Entropy. The diffusion-based generator $g_\phi$ enables bidirectional context propagation, allowing multiple semantic trajectories to emerge under the same prompt while maintaining structural coherence. However, in the absence of natural language feedback, Similarity improvements are modest and Perplexity variance rises, reflecting coarse reward alignment. The complete AoD configuration (*FRL*) combines the strengths of both: diffusion-driven diversity with linguistically grounded feedback. The LLM Judge and NLE stabilize the learning dynamics, reducing Perplexity and reinforcing high Similarity and Entropy while maintaining elevated Novelty.

*Model Transferability.* Table 2 highlights AoD's model-agnostic behavior across various combinations of autoregressive LLMs serving as the Prompt Optimizer and Judge agents, with LLaDA fixed as the generator. Across all pairings, Diversity scores remain consistently high (0.79–0.86), indicating that reinforcement-driven coordination generalizes regardless of the underlying model architecture. This demonstrates that AoD's policy learning operates on the shared language space of feedback and prompts, rather than relying on any specific model's internal representations. The prompt–feedback exchange mechanism $\pi_\theta(\Delta P \mid h)$ is thus invariant to the optimizer and judge configurations, enabling interchangeable agents without performance collapse. Runtime results further support AoD's reproducibility on consumer-grade hardware. Average GPU runtimes per feedback–generation cycle range from 110–122 seconds, even for 8–9B parameter models, confirming that multi-agent rollouts remain tractable under mid-range configurations. This efficiency stems from the frozen generator $g_\phi$ and the lightweight communication loop between autoregressive agents, which limits backpropagation overhead. Together, these results establish that AoD can be replicated using open-weight or API-based LLMs while preserving diversity and stability, making it accessible without dependence on high-end compute resources.

*Case Study: Structured JSON Synthesis in AoD..* We demonstrate AoD's functionality using the *MultiWOZ 2.1* booking domain as a task. The Prompt Optimizer first drafts a schema-conditioned instruction, e.g., "Generate a JSON object with fields {`name`, `address`, `phone`, `price_range`, `postcode`}." The DLM $g_\phi$ then produces diverse samples such as `{"name": "Parkview Inn", "address": "12 Milton Rd", "phone": "01223 443890", "price_range": "moderate", "postcode": "CB4 1LG"}`. The NLE computes metrics, while the LLM Judge transforms them into feedback, e.g., "*The JSON is valid and fluent but duplicates price patterns; introduce more unique names and locations.*" The optimizer uses this feedback through $\pi_\theta(\Delta P \mid h)$, iteratively improving prompt specificity and sampling balance. Over five successive iterations, Similarity rises from 0.64 to 0.88, Diversity and Novelty exceed 0.80, and Perplexity drops from 31.2 to 22.5. The Judge Agent confirms that the generated JSON records remain syntactically correct yet distinct. Field Overlap falls to 0.29, and TSR reaches 0.79, indicating low memorization and strong generalization.

## 5 CONCLUSION

AoD is the first framework to study how DLMs operate in a multi-agent reinforcement learning environment, demonstrating that natural language feedback can drive controllable, high-quality structured data generation. Futhermore, AoD achieves schema-compliant JSON outputs with higher diversity, novelty, and perplexity than standard LLM counterparts, while remaining reproducible on consumer hardware. Although limited to JSON synthesis, this work establishes DLMs as a powerful alternative for structured data generation and opens the door to future extensions for tabular datasets, code, and other structured domains.

# REFERENCES


[1] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2019. Not Enough Data? Deep Learning to the Rescue! https://arxiv.org/abs/1911.03118

[2] André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. 2024. Comprehensive Exploration of Synthetic Data Generation: A Survey. https://arxiv.org/abs/2401.02524

[3] Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding LLMS THE RIGHT WAY: Fast, non-invasive constrained generation. https://arxiv.org/abs/2403.06988

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. *arxiv.org* 4 (05 2020). https://arxiv.org/abs/2005.14165

[5] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. https://arxiv.org/abs/1810.00278

[6] Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on Large Language Model-Enhanced Reinforcement Learning: Concept, Taxonomy, and Methods. *IEEE Transactions on Neural Networks and Learning Systems* (01 2024), 1–21. https://doi.org/10.1109/tnnls.2024.3497992

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311* (April 2022). https://arxiv.org/abs/2204.02311

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. https://arxiv.org/abs/2501.12948 arXiv preprint arXiv:2501.12948.

[9] Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. 2023. Controlled Text Generation via Language Model Arithmetic. https://arxiv.org/abs/2311.14479

[10] Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. 2025. XGRAMMAR: Flexible and efficient structured generation engine for large language models. https://arxiv.org/abs/2411.15100

[11] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. https://doi.org/10.48550/arXiv.2305.14325

[12] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-Referential Self-Improvement Via Prompt Evolution. https://doi.org/10.48550/arXiv.2309.16797

[13] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. https://arxiv.org/abs/2210.08933

[14] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers. *arXiv (Cornell University)* (01 2023). https://doi.org/10.48550/arxiv.2309.08532

[15] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. https://doi.org/10.48550/arXiv.2402.01680

[16] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, van , and Tim Salimans. 2021. Autoregressive Diffusion Models. https://arxiv.org/abs/2110.02037

[17] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large Language Models Can Self-Improve. *ACL Anthology* Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (01 2023). https://doi.org/10.18653/v1/2023.emnlp-main.67

[18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55 (11 2022). https://doi.org/10.1145/3571730

[19] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. https://doi.org/10.48550/arXiv.2310.06825

[20] Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction. https://arxiv.org/abs/2303.04132

[21] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv:1909.05858 [cs]* (09 2019). https://arxiv.org/abs/1909.05858

[22] Chungpa Lee, Jongho Im, and Kim Joseph. 2025. A Generalized Theory of Mixup for Structure-Preserving Synthetic Data. https://arxiv.org/abs/2503.02645

[23] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. https://arxiv.org/abs/2205.14217

[24] Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. TruthfulQA: Measuring How Models Mimic Human Falsehoods. *arXiv:2109.07958 [cs]* (09 2021). https://arxiv.org/abs/2109.07958

[25] Shengcai Liu, Caishun Chen, Xinghua Qu, Ke Tang, and Yew-Soon Ong. 2023. Large Language Models as Evolutionary Optimizers. https://arxiv.org/abs/2310.19046

[26] Calvin Luo. 2022. Understanding Diffusion Models: A Unified Perspective. *arXiv:2208.11970 [cs]* (08 2022). https://arxiv.org/abs/2208.11970

[27] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large Language Diffusion Models. https://arxiv.org/abs/2502.09992

[28] OpenAI. 2023. GPT-4 Technical Report. *arXiv:2303.08774 [cs]* (03 2023). https://doi.org/10.48550/arXiv.2303.08774

[29] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv:2203.02155 [cs]* (03 2022). https://arxiv.org/abs/2203.02155

[30] Bo Pan, Jiaying Lu, Ke Wang, Li Zheng, Zhen Wen, Yingchaojie Feng, Minfeng Zhu, and Wei Chen. 2024. AgentCoord: Visually Exploring Coordination Strategy for LLM-based Multi-Agent Collaboration. https://arxiv.org/abs/2404.11943

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://arxiv.org/abs/1910.10683

[32] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092 [cs]* (02 2021). https://arxiv.org/abs/2102.12092

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv:2112.10752 [cs]* (04 2022). https://arxiv.org/abs/2112.10752

[34] Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv:2303.11366 [cs]* (03 2023). https://arxiv.org/abs/2303.11366

[35] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and et al. 2024. Gemma: Open Models Based on Gemini Research and Technology. https://doi.org/10.48550/arXiv.2403.08295 arXiv preprint arXiv:2403.08295.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971 [cs]* (02 2023).

[37] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. https://arxiv.org/abs/2501.06322

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. https://arxiv.org/abs/1706.03762

[39] Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. 2023. Generating Faithful Synthetic Data with Large Language Models: A Case Study in Computational Social Science. https://doi.org/10.48550/arXiv.2305.15041

[40] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning Language Model with Self Generated Instructions. *arXiv (Cornell University)* (12 2022). https://doi.org/10.48550/arxiv.2212.10560

[41] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, and et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. https://arxiv.org/abs/2204.07705 arXiv preprint arXiv:2204.07705.

[42] Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. CodecLM: Aligning Language Models with Tailored Synthetic Data. https://arxiv.org/abs/2404.05875

[43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting

Elicits Reasoning in Large Language Models. *arXiv:2201.11903 [cs]* (10 2022). https://arxiv.org/abs/2201.11903

[44] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. https://doi.org/10.48550/arXiv.2308.08155

[45] Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, and Lichao Sun. 2024. UniGen: A Unified Framework for Textual Dataset Generation Using Large Language Models. https://arxiv.org/abs/2406.18966

[46] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. https://arxiv.org/abs/2202.07922

[47] Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias. https://doi.org/10.48550/arXiv.2306.15895

[48] Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2023. Mixed-Type Tabular Data Synthesis with Score-based Diffusion in Latent Space. https://arxiv.org/abs/2310.09656

[49] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *arXiv:2303.18223 [cs]* (03 2023). https://arxiv.org/abs/2303.18223

[50] Ying Zhou, Xinyao Wang, Yulei Niu, Yaojie Shen, Lexin Tang, Fan Chen, Ben He, Le Sun, and Longyin Wen. 2024. DiffLM: Controllable Synthetic Data Generation via Diffusion Language Models. https://arxiv.org/abs/2411.03250