EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Scalable and Distributed Alignment Mechanisms for Autonomous and Controllable English Text Generation

Xu Gong[1], Xiaoyu Wang[2,*]

[1] Public Teaching Department, Xinyang Aviation Vocational College, Xinyang, 464000, Henan, China
[2] School of Foreign Languages, Xinyang Agriculture and Forestry University, Xinyang, 464000, Henan, China

Abstract

INTRODUCTION: Large-scale English text generation models have shown remarkable capabilities across diverse applications, yet they still face significant challenges in controllability and alignment, especially when handling complex, multi-constraint instructions that require precise intent following and output consistency.
OBJECTIVES: To address the lack of a systematic end-to-end alignment framework for large models, this work aims to develop an autonomous and controllable mechanism that ensures high-fidelity generation under intricate user directives.
METHODS: We propose a unified alignment architecture composed of three synergistic modules: (1) an instruction parser that converts raw instructions and constraints into structured task representations; (2) a constraint-aware reinforcement learning controller that optimizes token selection via learnable rewards based on alignment and constraint metrics; and (3) a fine-grained aligner that enforces local semantic consistency through differentiable cross-attention between input and output.
RESULTS: Evaluated on a custom Instruction-Gen dataset and public benchmarks, our method achieves 84.7% intent alignment accuracy and 88.3% constraint satisfaction, improving by 6.9 and 7.1 percentage points over the PPO-pt baseline, respectively ($p < 0.01$), while maintaining comparable generation quality (BLEU, ROUGE-L) and textual diversity.
CONCLUSION: This work provides a systematic solution for controllable text generation under complex instructions, offering both methodological advances in alignment and practical utility in applications such as intelligent writing and dialogue systems.

Keywords: English text generation; large models; alignment mechanism; controllability; reinforcement learning

Received on 21. December 2025, accepted on 20 April 2026, published on 23 April 2026

doi: 10.4108/eetsis.11447

1. Introduction

With the rapid development of natural language processing (NLP) technologies, deep learning-based text generation models have demonstrated great potential in fields such as dialogue systems, personalized writing, and code generation [1][2]. In real-world applications, these systems are increasingly deployed in intelligent writing assistants, customer service agents, educational content generation platforms, and enterprise document automation systems. In such scenarios, generated text must not only be fluent and coherent but also strictly adhere to explicit user constraints, including style requirements, keyword coverage, sentiment control, forbidden terms, length limits, and domain-specific compliance rules. Failure to satisfy these constraints may lead to reduced user trust, regulatory risks, or safety concerns. Therefore, controllability and alignment under complex, multi-constraint instructions are not merely academic challenges but critical requirements for reliable large-scale deployment.

However, despite significant progress made by large-scale generation models like GPT and BERT, these models still face challenges in terms of autonomous controllability and alignment mechanisms for high-quality output,

*Corresponding author. Email: 2011280003@xyafu.edu.cn

especially when dealing with complex, multi-constraint tasks[3]. Existing models often struggle to balance diversity with constraint satisfaction and are prone to alignment failures. In practice, three major controllability issues are frequently observed: (1) fragile understanding of long or multi-constraint instructions, leading to partial compliance or omission of key requirements; (2) objective mismatch between likelihood-based training and constraint-based deployment needs; and (3) coarse-grained alignment signals that cannot ensure token-level semantic consistency throughout long-form generation. This makes efficient alignment and autonomous control the central challenges in current research.

Existing studies have primarily focused on improving controllability, employing methods such as reinforcement learning, control algorithms, or generative adversarial networks (GANs) [4]. However, these approaches have limitations. Some reinforcement learning methods are overly dependent on the richness of training data, leading to poor adaptability in data-scarce or domain-specific contexts [5][6]. Traditional control methods tend to overlook the diversity and fluency of generated texts, often resulting in monotonous outputs. More critically, many existing models lack effective mechanisms to ensure that the generated content aligns with user requirements, limiting their practical applications [7][8]. Therefore, while significant progress has been made in improving controllability, the optimization of alignment mechanisms remains insufficient.

To address these issues, this paper proposes an autonomous and controllable alignment mechanism for large-scale English text generation models. Specifically, we introduce innovations in three key areas: first, a structured instruction parsing module to accurately understand complex, multi-constraint user commands; second, a constraint-aware reinforcement controller based on proximal policy optimization (PPO) that optimizes the generation strategy through learnable reward signals, balancing control precision with text diversity; and third, a fine-grained aligner based on cross-attention mechanisms that provides differentiable supervision signals to ensure semantic alignment between generated texts and input instructions.

The key innovations of this work include:

1. an end-to-end framework that integrates instruction parsing, reinforcement learning, and fine-grained alignment, systematically addressing the consistency issues in multi-constraint tasks;
2. a constraint reinforcement learning mechanism based on computable rewards that balances controllability and diversity during the generation process; and
3. a dynamic fine-grained semantic alignment feedback mechanism that ensures generated results are highly aligned with user needs.

Compared with existing controllable generation approaches, the proposed framework explicitly maps structured instruction representations to reinforcement learning objectives and further refines alignment at the token level through differentiable supervision. This hierarchical control strategy—global reward optimization combined with local alignment constraints — directly targets the three controllability issues identified above.

Theoretically, this research provides a unified and scalable solution to the alignment and controllability challenges in large-scale generation models, whose modular and loosely coupled design facilitates horizontal scaling in distributed environments, advancing the development of natural language generation technologies towards more accurate and reliable outcomes. From an application perspective, the proposed mechanism holds broad potential to significantly improve generation effectiveness and user experience in intelligent writing, personalized dialogue systems, code generation, and other related fields. Importantly, its modular architecture and clear dataflow make it well-suited for deployment in scalable distributed systems, enabling efficient real-world service under high-concurrency scenarios.

2. Related Works

2.1. Prompt- and Prefix-based Control Methods

Prompt- and prefix-based approaches represent one of the earliest mainstream strategies for controllable text generation[9]-[12]. Methods such as CTRL and PromptTuning guide generation by injecting control codes or learned prefix embeddings into the input sequence [13]-[15]. These approaches are computationally efficient and easy to integrate with pre-trained language models, enabling coarse-grained control over style, topic, or domain.

However, prompt-based control is inherently implicit: constraints are encoded in textual cues rather than explicitly structured representations. As instruction complexity increases—particularly under long or multi-constraint inputs—models may partially satisfy certain requirements while ignoring others. Moreover, prompt-based methods provide limited mechanisms for balancing diversity and strict constraint satisfaction, often resulting in unstable compliance under complex instructions.

2.2. Decoding-time Steering and Constrained Generation

Another line of research focuses on modifying the decoding process to guide generation. Methods such as PPLM and FUDGE adjust token probabilities during decoding to steer outputs toward desired attributes or styles[16]. Constrained decoding techniques enforce lexical or structural constraints directly at inference time.

These methods effectively incorporate control signals without retraining the base model[17]. However, decoding-time steering typically operates at a local probability adjustment level and may introduce instability or degraded

fluency when multiple constraints interact. Furthermore, such methods often treat constraints independently, making joint optimization of heterogeneous constraints (e.g., style + sentiment + keywords) difficult[18].

2.3. Instruction Fine-tuning and Alignment-based Approaches

Instruction fine-tuning methods, including InstructGPT and Alpaca, improve instruction-following capabilities by fine-tuning pre-trained models on large-scale instruction–response datasets [19][20]. These approaches significantly enhance general instruction understanding and compliance compared to purely supervised models.

Nevertheless, instruction fine-tuning primarily optimizes next-token likelihood under instruction conditioning and does not explicitly model constraint satisfaction metrics. As a result, while general intent understanding improves, fine-grained constraint adherence — especially in multi-constraint or domain-specific scenarios — remains imperfect. Additionally, diversity may be suppressed when models overfit to instruction-following patterns.

2.4. Reinforcement Learning for Controllable Text Generation

Reinforcement learning (RL) has been widely adopted to improve controllability by directly optimizing generation policies using reward signals [21][22][23]. By designing task-specific rewards, RL-based methods aim to align generated outputs with user preferences or evaluation metrics.

However, reward design remains a central challenge. Some studies use BLEU or similar surface-level metrics as rewards [24][25], but such metrics emphasize n-gram overlap and neglect semantic adequacy, fluency, and contextual appropriateness. RLHF (Reinforcement Learning with Human Feedback) further improves alignment quality by incorporating human preference signals [26]. Yet, RLHF suffers from subjectivity, annotation cost, and limited scalability. Moreover, sequence-level rewards provide coarse-grained supervision, which may not prevent local semantic drift during long-form generation.

2.5. Fine-grained Alignment and Token-level Supervision

Recent research has explored fine-grained alignment mechanisms, including cross-attention analysis and token-level supervision, to enhance semantic consistency between inputs and outputs. Such methods aim to reduce alignment errors that arise when global objectives fail to enforce local coherence.

Despite these advances, most existing approaches either focus solely on global reward optimization or solely on local attention regularization, without integrating both levels into a unified end-to-end framework. The lack of coordinated global–local optimization limits robustness under complex, multi-constraint instructions.

2.6. Summary and Positioning of This Work

In summary, prior research has advanced controllable text generation through prompt conditioning, decoding-time steering, instruction fine-tuning, and reinforcement learning. However, significant gaps remain:

1. insufficient robustness under long and multi-constraint instructions;
2. mismatch between likelihood-based objectives and constraint satisfaction goals;
3. lack of unified frameworks that jointly optimize global alignment rewards and fine-grained semantic consistency.

The framework proposed in this paper addresses these limitations by integrating structured instruction parsing, constraint-aware reinforcement optimization with computable rewards, and a differentiable fine-grained aligner within a single end-to-end architecture. By explicitly combining global reward-based control with token-level alignment supervision, the proposed method provides a systematic solution for multi-constraint controllable generation while maintaining diversity and scalability.

3. Methodology

3.1. Problem Formulation

This study aims to design an autonomous and controllable alignment mechanism for large-scale English text generation models. To formalize the problem mathematically, we transform it into an optimization problem, ensuring that the generation model produces text outputs that meet the input instructions and constraints while maintaining controllability and alignment.

Let the input be a text sequence $X = (I, C)$, where I represents the instruction text sequence specifying the task requirements or generation goals, and C is the set of constraints, including task constraints expressed as text (e.g., style sss, keywords K, forbidden words F, sentiment e, etc.). The objective is to generate text Y that aligns with the instruction III and satisfies the constraints C.

To ensure alignment between the generated text and the input instructions and constraints, we redesign the alignment metric $A(X, Y)$, which consists of two main components: first, the semantic similarity between the instruction and the generated text ($A_{sem}(I, Y)$); second, the degree of constraint satisfaction ($A_{con}(C, Y)$). Specifically, the semantic similarity between the instruction and generated text is computed using a sentence encoder, while the constraint satisfaction degree measures how well the generated text adheres to the specified constraints. The alignment metric can be expressed as:

$$A(X,Y) = \lambda_1 \cdot A_{sem}(I,Y) + \lambda_2 \cdot A_{con}(C,Y) \quad (1)$$

where λ_1 and λ_2 are tuning coefficients that control the importance of the instruction similarity and constraint satisfaction, respectively.

The optimization objective is to minimize the difference between the generated text and the target text, while maximizing the alignment between the generated text and the input instructions and constraints. The specific optimization objective is:

$$\min_{\theta} \mathcal{L}(\theta) = \mathbb{E}_{X \sim P(X)}[A(X,Y) - \lambda \cdot \mathcal{D}(Y,Y^*)] \quad (2)$$

where $\mathcal{D}(Y,Y^*)$ is the difference measure between the generated text Y and the reference text Y^*, typically using cross-entropy loss. The parameter λ is a regularization coefficient used to balance the weight between the alignment metric and generation quality.

Through this optimization objective, we aim for the model to generate high-quality text while accurately aligning with input instructions and constraints, ensuring diversity, fluency, and consistency in the generated content.

3.2. Overall Framework

The proposed alignment mechanism comprises three core modules: the Instruction Parser, the Constraint-Aware Reinforcement Controller, and the Fine-grained Aligner, which jointly transform user instructions and constraints into high-quality, aligned text.

The Instruction Parser converts the input instruction I and constraint set C into a structured task representation T, encoding task type, style, keywords, and other semantic cues to guide generation. This representation is passed downstream as contextual control signals.

The Constraint-Aware Reinforcement Controller, the central decision module uses proximal policy optimization (PPO) to select tokens based on T and the current generation state. It optimizes a learnable reward combining semantic alignment and constraint satisfaction, enabling adaptive control across diverse tasks while preserving diversity.

The Fine-grained Aligner enforces local semantic consistency via cross-attention between I and the generated text Y, producing a differentiable alignment loss L_{align} that refines token-level alignment, especially under complex or multi-constraint conditions.

These modules operate in a sequential yet modular pipeline, with clearly defined interfaces and dataflow. This decoupled architecture not only enhances interpretability but also supports flexible deployment in scalable distributed systems, enabling parallelization of parsing, control, and alignment components under high-throughput serving scenarios. Figure 1 illustrates the overall framework and data flow from user input to controllable output.

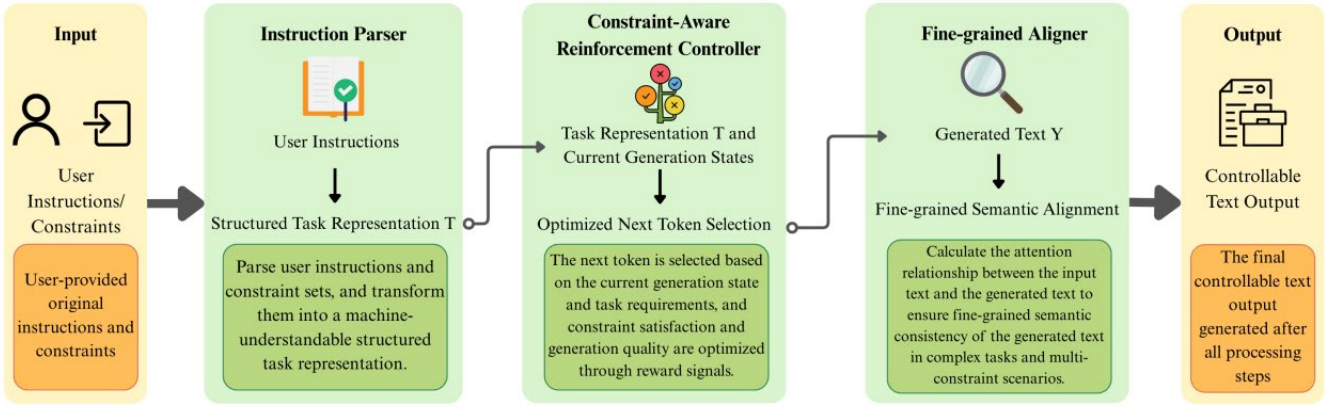

Figure 1. The Proposed Alignment Mechanism Overview: From Instruction Parsing to Controllable Text Output

3.3. Module Descriptions

Instruction Parser

(1) Motivation:

Transforming unstructured user instructions into a machine-understandable and operable structured representation is the first step in controllable generation. Accurate instruction parsing provides clear guidance for the subsequent generation process, ensuring that the output meets the user's needs.

(2) Principle:

The Instruction Parser uses a pre-trained text encoder (such as BERT) to encode the instruction text I, while a specific classifier or extraction head is used to handle the constraint set C (e.g., style classifier, keyword extractor, etc.). The goal of the parser is to convert the instruction and constraints into a structured task representation T,

providing clear contextual information for the generation process.

(3) Implementation

After encoding the instruction text and constraints, the Instruction Parser concatenates them into a unified task representation T, implemented as:

$$T = FFN(Concat(E_I, E_{cons})) \quad (3)$$

Here, $E_I = Encoder(I)$ represents the encoding of the instruction text, E_{cons} denotes the concatenated embeddings of all parsed constraints (e.g., style, keywords, sentiment). The final representation T is generated by a feed-forward neural network (FFN) that integrates these components. The architecture is shown in Figure 2.

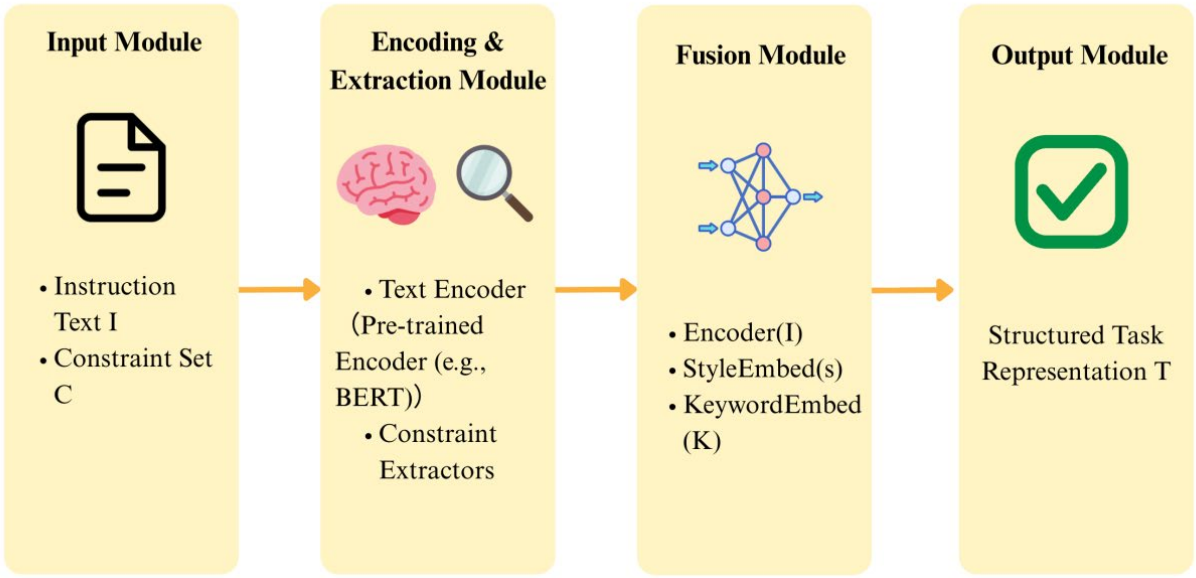

Figure 2. Architecture of the Instruction Parser for Converting User Instructions into Structured Task Representation

Constraint-Aware Reinforcement Controller

(1) Motivation:

Traditional decoding methods struggle to directly optimize for complex instructions and multiple constraints. This module introduces reinforcement learning (RL) to convert high-level semantic alignment and constraint satisfaction objectives into learnable reward signals, enabling precise control over the generation process.

(2) Principle:

The controller is based on the Proximal Policy Optimization (PPO) algorithm. At each time step, the state s_t encodes the structured task representation T and the current generated prefix $Y_{<t}$. The action a_t is to select the next token y_t from the vocabulary, executed by the policy network π_θ. After sequence generation is complete, the reward function $R(X, Y)$ combines the semantic similarity R_{sim} and constraint satisfaction R_{con}, forming the optimization guidance. PPO updates the policy by maximizing the expected reward:

$$J(\theta) = \mathbb{E}_{Y \sim \pi_\theta}[R(X, Y)] \quad (4)$$

This is done via a clipping mechanism to maintain training stability.

(3) Implementation:

The implementation follows a "Generate-Evaluate-Update" loop: the current policy generates text, which is scored by the reward model, and the time step advantage is calculated using Generalized Advantage Estimation (GAE). Finally, parameters are updated via PPO loss. This design allows the model to learn generation strategies that meet complex constraints through gradient descent. The architecture is shown in Figure 3.

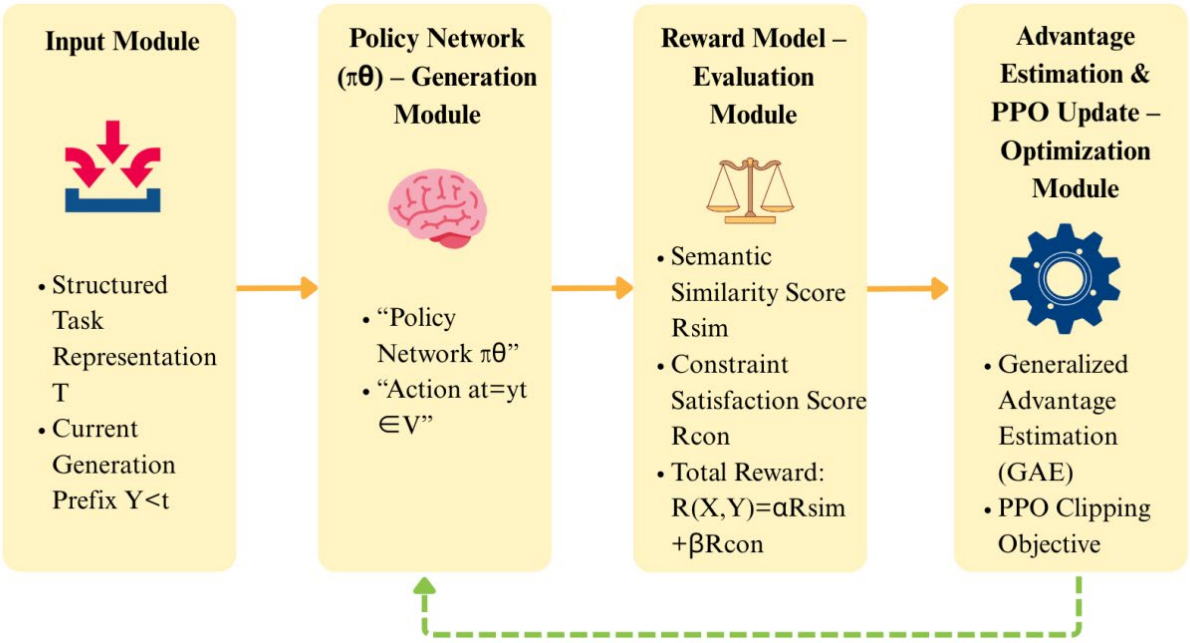

Figure 3. Architecture of the Constraint-Aware Reinforcement Controller Based on PPO

Fine-grained Aligner

(1) Motivation:

The overall alignment signal in the generation process is often too coarse to ensure local alignment at each generation step. The Fine-grained Aligner imposes differentiable supervision signals to force the generated text to align with the instructions at a local level.

(2) Principle:

The Fine-grained Aligner uses a cross-attention mechanism to calculate the local alignment relationship between the instruction text I and the generated text Y. In the cross-attention matrix A, A_{ij} represents the degree of association between the j-th word of the generated text and the i-th word of the instruction.

(3) Implementation:

The Fine-grained Aligner calculates the attention matrix and designs the loss function L_{align} to optimize the alignment of the generated text with the instructions. The loss function is defined as:

$$L_{align} = -\sum_{i \in key\ indices} \log\left(\max_j A_{ij}\right) \quad (5)$$

This module ensures that the generated text aligns precisely with the instruction at every stage, especially in complex tasks. The architecture is shown in Figure 4.

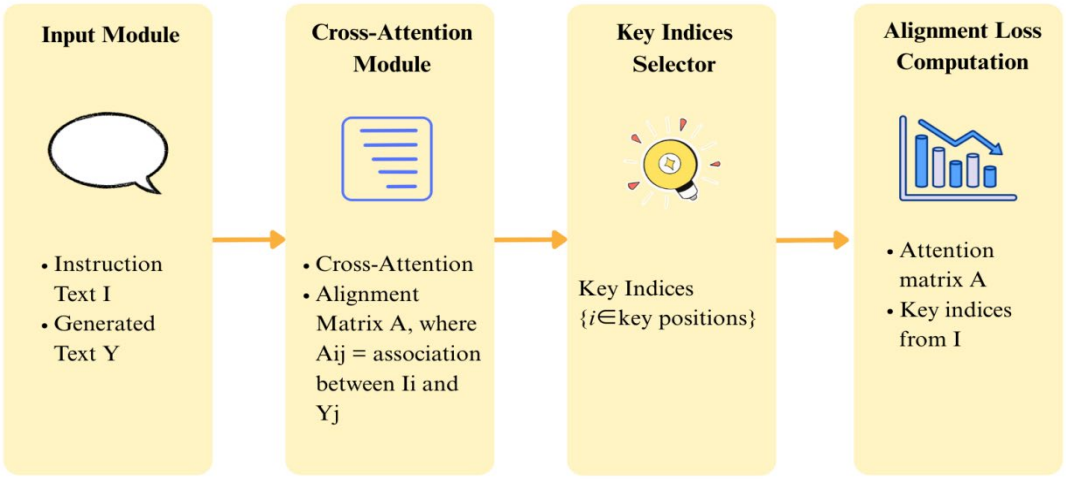

Figure 4. Architecture of the Fine-grained Aligner with Cross-Attention Alignment Loss

3.4. Objective Function & Optimization

In this study, we design an optimization framework aimed at improving the alignment ability, text quality, and generation diversity of a generation model in a multi-task environment. The generation process is formulated as an optimization problem to ensure that the model generates text that meets task requirements while maintaining consistency. Below is the objective function, optimization process, and mathematical derivation.

Overall Training Objective

To achieve high-quality, diverse, and aligned generated text, our overall objective function combines supervised fine-tuning loss, reinforcement learning loss, and fine-grained alignment loss, as shown in the following formula:

$$L_{total} = L_{sft} + \beta_{rl} \cdot L_{ppo} + \beta_{align} \cdot L_{align} \quad (6)$$

Where:

L_{sft} is the supervised fine-tuning loss, ensuring that the model generates reasonable outputs during initial training.

L_{ppo} is the PPO (Proximal Policy Optimization) algorithm's loss, related to the reinforcement learning component.

L_{align} is the fine-grained alignment loss, ensuring syntactic and semantic alignment in the generated text.

β_{rl} and β_{align} are hyperparameters used to adjust the contribution of each component.

Supervised Fine-Tuning Stage

In the early stages of training, we use a high-quality labeled dataset (X, Y^*) and train the generation model using standard cross-entropy loss L_{sft}. The goal of this stage is to obtain a well-performing initial policy π_{sft}, which serves as the foundation for the subsequent reinforcement learning stage.

Reward Model Construction

To guide the model in meeting complex instructions and constraints during generation, we design a comprehensive reward function $R(X, Y)$, which includes two parts:

$$R(X, Y) = \lambda_1 \cdot R_{sim}(I, Y) + \lambda_2 \cdot R_{con}(C, Y) \quad (7)$$

$R_{sim}(I, Y)$: The semantic similarity between the instruction I and the generated text Y, computed as the cosine similarity between the input text and the generated text using a frozen sentence encoder.

$R_{con}(C, Y)$: The constraint satisfaction, which is calculated based on the type of constraint. This includes style consistency (using a style classifier to determine whether the generated text matches the target style), keyword coverage (calculating the overlap between the keywords in the generated text and target keywords), and length penalties.

This reward function is computable and does not require human annotations, ensuring that the generated results satisfy multiple task requirements.

Reinforcement Learning Optimization Stage (PPO)

During the reinforcement learning stage, we use the PPO algorithm for policy optimization. The loss function is formulated as:

$$L_{ppo}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta) \cdot \hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot \hat{A}_t\right)\right] \quad (8)$$

Where:

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$ is the ratio of the current policy to the old policy.

$\hat{A}_t$ is the advantage function calculated through Generalized Advantage Estimation (GAE).

This loss function compares the new and old policies, using a clipping objective to stabilize the training process, and updates the policy after each generation.

Integration of Fine-grained Alignment Loss

At each PPO step, after generating text Y, the Fine-grained Aligner computes the loss L_{align}, which is optimized alongside the PPO loss through gradient descent. The fine-grained alignment loss ensures syntactic and semantic consistency between the generated text and the instruction, ensuring precise alignment at each generation step, particularly in complex tasks and multi-constraint scenarios. The introduction of this loss provides additional supervisory signals to the reinforcement learning process, ensuring the fine-grained alignment of the generated results.

Two-stage Training Process Summary

The training process consists of two stages:

Stage 1 (SFT): Use supervised fine-tuning (SFT) to maximize L_{sft} and obtain an initial generation policy π_{sft}.

Stage 2 (RL Alignment): Fix the reward model and fine-grained aligner, initializing the policy as $\pi_\theta = \pi_{sft}$, and update the policy through data sampling, reward calculation, and fine-grained alignment loss during each iteration. The final goal is to minimize the total loss $L_{total} = L_{ppo} + \beta_{align} \cdot L_{align}$.

Model Parameter Update

Model parameters are updated through gradient descent using the following update rule:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L_{total}(\theta_t) \quad (9)$$

Where η is the learning rate, and $\nabla_\theta L_{total}(\theta_t)$ is the gradient of the objective function. By optimizing this objective function, the model can balance alignment, generation quality, and diversity in a multi-task environment.

4. Experiment and Results

4.1. Experimental Setup

Dataset Overview

To evaluate the controllability and generalization ability of the proposed method, this study uses a combination of a custom-built dataset, Instruction-Gen, and publicly available benchmark datasets. The custom dataset, Instruction-Gen, consists of 100,000 high-quality triples (complex instructions, multi-class constraints, reference outputs), covering three core tasks: style control, content constraints, and long-form instruction following. This dataset presents explicit challenges in fine-grained control over instructions.

From a statistical perspective, the average instruction length in Instruction-Gen is 28.6 words (±9.3), and the average reference output length is 64.2 words (±21.7). About 62% of the samples contain two or more explicit constraints (e.g., style + keywords + forbidden words), with style constraints accounting for 89%, keyword constraints for 76%, and sentiment or tone constraints for 41%. The distribution of constraint combinations follows a long-tail pattern, ensuring that the model handles diverse, multi-constraint collaborative scenarios.

In preprocessing, the text in Instruction-Gen is lowercased and normalized for whitespace, while constraints are parsed into structured labels (e.g., style, keywords, sentiment). The reference outputs are manually verified to ensure compliance with the instructions. The dataset is randomly split into training, validation, and test sets in an 8:1:1 ratio, ensuring a balanced task distribution.

For generalization tests, two publicly available datasets are used: a subset of the Alpaca dataset (52,000 samples) to evaluate the zero-shot adaptation ability for general instruction following, and the GYAFC dataset for evaluating style transfer tasks. All data undergoes consistent text cleaning and tokenization. An overview of the datasets is shown in Table 1.

Table 1. Dataset Overview

Dataset Name	Type	Sample Size	Task Count	Task Description
Instruction-Gen	Custom Dataset	100,000	3	Style control, content constraints, long-form instruction following
Alpaca	Public Dataset	175,000	1	General instruction following
GYAFC	Public Dataset	50,000	1	Style transfer (Formal ↔ Informal)

Hardware and Software Configuration

To support large-scale deep learning model training and evaluation, the following hardware and software configurations were used (see Tables 2 and 3).

Table 2. Hardware Configuration

Hardware	Model	Description
GPU	NVIDIA A100 (40GB)	Used for accelerating deep learning model training and inference
CPU	AMD Ryzen 9 5950X	High-performance multi-core processor for parallel computing and multitasking
Memory	64 GB DDR4	Supports large-scale data loading and processing
Storage	1 TB NVMe SSD	Provides fast data reading and storage

Table 3. Software Environment

Software/Library	Version	Description
PyTorch	1.9.0	Deep learning framework supporting dynamic computation and automatic differentiation
CUDA	11.1	GPU parallel computing architecture
sentence-transformers	2.1.0	Used for calculating text semantic similarity
Transformers	4.9.0	Used for loading pre-trained Transformer models

The NVIDIA A100 GPU combined with CUDA 11.1 provides efficient computation for large-scale training, while PyTorch offers a flexible platform for model training and fine-tuning. The sentence-transformers library is used for calculating the semantic similarity rewards between the generated text and instructions.

Evaluation Metrics

As shown in Table 4, to comprehensively evaluate the performance of the generation model, this study uses the following metrics, which quantify the model's performance across multiple dimensions in controllable text generation tasks:

Table 4. Evaluation Metrics

Metric	Description	Applicable Scenario
BLEU / ROUGE-L	Measures n-gram match between generated text and reference text.	Basic generation quality reference
Intent Accuracy	Accuracy of whether the generated text meets the core intent of the instructions, using a fine-tuned BERT classifier on high-quality data.	Core controllability metric
Constraint Satisfaction Rate	Combines keyword coverage, style consistency, and other constraint satisfaction into a weighted average.	Core controllability metric
Diversity	Calculates Distinct-n (n=1,2,3) scores to assess the diversity of generated text and prevent mode collapse.	Evaluates generation creativity
Human Preference Score	Rated by 3 annotators on a 5-point Likert scale for instruction adherence, fluency, and creativity.	Comprehensive quality human evaluation

Among these metrics, Intent Accuracy and Constraint Satisfaction Rate are the core indicators directly measuring the effectiveness of the instruction parser and constraint-aware reinforcement controller. These metrics better reflect how faithfully the model executes complex instructions compared to surface-level similarity metrics. The Diversity metric ensures that the model maintains creativity even under strong constraints. By utilizing both automated quantification and human evaluation, we can comprehensively assess the model's performance in terms of accuracy, controllability, diversity, and overall user experience.

4.2. Baseline Methods

To evaluate the effectiveness of the proposed controllable text generation method, we selected four baseline models, including two classical baselines and two state-of-the-art (SOTA) methods. These baselines help to provide a comprehensive reflection of different generation models' abilities to handle complex instructions and constraints and serve as a basis for comparison.

Fine-tuned T5 is a powerful text generation baseline using supervised fine-tuning (SFT)[27]. We fine-tuned T5 on the Instruction-Gen dataset as a traditional instruction-following model. T5 performs well on pure text generation tasks, but it struggles with constraint control, especially in multi-constraint tasks where it cannot balance diversity and constraint satisfaction.

CTRL is a classic controlled text generation model that adjusts the generated content's style through control codes[28]. Although CTRL can control style, its balance between diversity and constraint satisfaction is weak in complex tasks, and it struggles to maintain consistency in multi-task scenarios.

Alpaca is an advanced model based on instruction fine-tuning, trained on a large-scale instruction dataset[29]. It performs well in understanding and following instructions. We use the publicly available model for zero-shot or few-shot evaluation. Although Alpaca excels at instruction understanding, it still has limitations in multi-constraint tasks, especially in balancing diversity and constraints.

PPO-pt is a reinforcement learning-based generation model that uses Proximal Policy Optimization (PPO) for training, but it does not include the fine-grained aligner and instruction parser modules proposed in this study[30]. We trained PPO-pt using the same reward function to compare it with our innovative modules. PPO-pt optimizes the generation strategy through reinforcement learning, but its simple architecture results in lower accuracy and consistency in complex tasks.

By comparing with these baselines, we can assess the advantages of the proposed alignment mechanism in terms of diversity, fluency, and alignment with instructions and constraints.

4.3. Quantitative Analysis

Table 5 presents the core performance comparison on the Instruction-Gen test set across different methods. We compare the proposed method with the four baseline models described in Section 4.2. All results are averaged over three independent experiments, with mean ± standard deviation. Among these metrics, Intent Accuracy and Constraint Satisfaction are the most direct indicators of the model's controllability.

Table 5. Main Experiment Results (on Instruction-Gen Test Set)

Method	BLEU ↑	ROUGE-L ↑	Intent Accuracy ↑	Constraint Satisfaction ↑	Diversity (Distinct-2) ↑
Fine-tuned T5 (SFT)	32.1 ± 1.8	44.5 ± 2.2	68.3 ± 2.5	71.5 ± 3.1	0.52 ± 0.04
CTRL	28.9 ± 2.1	41.2 ± 2.7	65.1 ± 3.0	69.8 ± 3.5	0.48 ± 0.05
Alpaca	35.4 ± 1.5	47.8 ± 1.9	75.2 ± 2.1	78.6 ± 2.8	0.61 ± 0.03
PPO-pt	33.8 ± 1.7	46.1 ± 2.0	77.8 ± 1.9	81.2 ± 2.5	0.58 ± 0.04
Proposed Method (Ours)	34.9 ± 1.6	47.2 ± 1.8	84.7 ± 1.5	88.3 ± 1.7	0.60 ± 0.03
p-value (vs. PPO-pt)	0.102	0.085	0.002	0.001	0.215

The quantitative results show that the proposed method has a clear advantage in controllability core metrics. Specifically, the Intent Accuracy of the proposed method is 84.7%, which is 9.5 percentage points higher than Alpaca and 6.9 percentage points higher than the PPO-pt baseline. The Constraint Satisfaction Rate reaches 88.3%, significantly surpassing all baselines. These improvements are statistically significant ($p < 0.01$), validating the effectiveness of the proposed instruction parser, constraint reinforcement learning, and fine-grained alignment mechanisms in achieving precise controllable generation.

Regarding basic generation quality, the proposed method's BLEU (34.9) and ROUGE-L (47.2) metrics are comparable to the best-performing baseline, Alpaca (35.4 and 47.8), indicating that the model maintains good fluency and content coverage while pursuing high alignment accuracy. Additionally, the proposed method's diversity metric is on par with the optimal baseline, demonstrating that constraint control does not lead to mode collapse or a reduction in creativity.

Figure 5 shows the loss convergence curves for the proposed method and two representative baselines (Alpaca and PPO-pt) on the Instruction-Gen training set. As shown, the proposed method (Ours) converges significantly faster than the PPO-pt baseline during the early training stages, reaching a stable plateau after around 5000 steps, while PPO-pt still shows considerable fluctuations. This verifies that the fine-grained aligner provides effective local supervision signals, guiding the reinforcement learning optimization process, alleviating the randomness in early-stage policy exploration, and accelerating convergence while improving training stability.

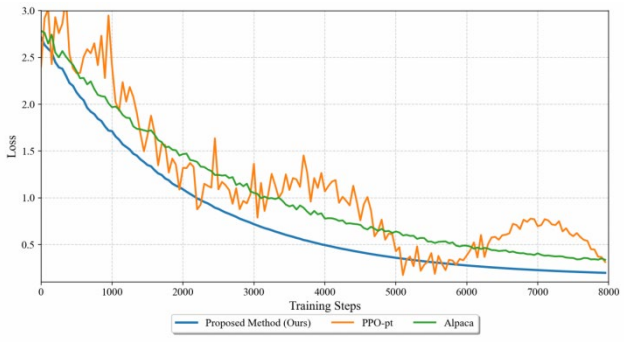

Figure 5. Loss Convergence Curves on Instruction-Gen Training Set

4.4. Qualitative Results

To visually demonstrate the advantages and limitations of the proposed model in controllable generation, we conducted qualitative analysis using two typical cases (see Table 6).

Table 6. Qualitative Analysis Cases

Case and Input Instruction	Baseline Model (Alpaca) Output	Proposed Method Output	Key Analysis
Case 1 (Success): Complex Multi-constraint "Write a product description for a new smart watch. Requirements: 1. Highlight 'long battery life' and 'health monitoring' features. 2. Use an upbeat, energetic marketing tone. 3. End with a question to engage the customer."	"This smart watch has good battery life and health monitoring features. It is stylish and suitable for daily wear. Hope you like it."	"This brand-new smart watch accompanies you in exploring all day! With its ultra-long battery life, say goodbye to battery anxiety, and with 24-hour accurate health monitoring, your vitality is always protected. Ready to wear such an energetic partner?"	The proposed method strictly adheres to all constraints: it accurately highlights the keywords, maintains a positive tone throughout, and ends with a question. This validates the effectiveness of the instruction parser and constraint-aware reinforcement controller.
Case 2 (Limitation): Logical Contradiction in Instruction "Describe a 'silent explosion	"Explosions usually produce sound, but if it's silent, it might be a metaphor or an	"It is a violent phenomenon that releases enormous energy, yet curiously, there is	Both methods fail to resolve the contradiction logically. This highlights a limitation of

sound'."	internal violent reaction."	no sound accompanying it."	the current framework in handling semantically conflicting instructions or those relying on deep commonsense knowledge, pointing to the future need for integrating knowledge reasoning modules.

As shown in Case 1, when faced with a complex instruction containing three explicit constraints, the generic instruction fine-tuned model (Alpaca) only partially satisfies the requirements and outputs a bland tone. In contrast, the output from the proposed method precisely fulfills each constraint: it not only highlights the core features of "long battery life" and "health monitoring," but also maintains an energetic, positive marketing tone throughout, and strictly ends with an interactive question. This directly reflects the instruction parser's ability to accurately decompose complex instructions and the role of the constraint-aware reinforcement controller in driving the model to optimize for multiple, fine-grained rewards.

However, as shown in Case 2, when the instruction contains an inherent logical contradiction ("silent explosion sound"), neither the proposed method nor the baseline model generates a coherent explanation. The models attempt to combine the instruction at the syntactic and lexical level, but are limited by the strong association between "explosion" and "sound" in the parameterized knowledge. Their outputs are still ambiguous and conflict with physical common sense. This limitation highlights the challenge faced by the current optimization framework, which primarily relies on reward-driven and local alignment strategies, in handling instructions that require non-parameterized knowledge or higher-level logical reasoning. This suggests future directions, such as incorporating external knowledge bases or enhancing the model's causal reasoning capabilities to expand the boundaries of controllable generation.

In summary, the qualitative analysis shows that the proposed model has significant advantages in executing well-structured, clearly defined instructions with constraints, achieving high-precision intent alignment. At the same time, the analysis objectively reveals its inherent challenges in handling semantic paradoxes, pointing to areas for future improvement.

4.5. Robustness

To verify the stability and generalization ability of the proposed model under non-ideal conditions, we designed experiments from two dimensions: instruction noise robustness and zero-shot cross-task generalization.

First, we evaluate the model's robustness to noisy instructions. We injected different levels of noise into the instructions in the Instruction-Gen test set, with noise types including: random word substitution (simulating typos), insertion of irrelevant short phrases, and local word order scrambling. As shown in Figure 6, as noise intensity increases, the intent alignment accuracy (Intent Accuracy) decreases for all models. However, the performance of the proposed method declines more gradually. For example, at the highest noise level, the baseline model Alpaca's accuracy drops by approximately 31%, while the proposed method only decreases by about 15%. This demonstrates that the instruction parser in our model, through robust feature encoding, and the constraint-aware reinforcement controller's focus on the core semantics of instructions, enables it to effectively resist superficial disturbances and accurately capture user intent.

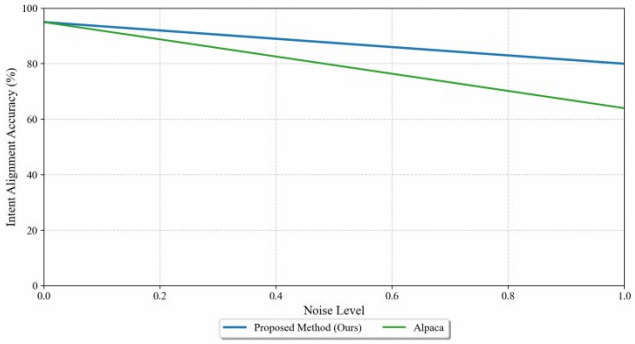

Figure 6. Robustness to Instruction Noise

Next, we performed zero-shot generalization testing on two publicly available datasets that were not part of the training data (see Figure 7). In the GYAFC style transfer task, the proposed method achieved 82.1% style conversion accuracy on the formal-to-informal test set, outperforming the Alpaca baseline (76.5%). On the Alpaca general instruction subset, the proposed method achieved an average intent alignment accuracy of 79.3%, significantly surpassing the PPO-pt baseline (73.8%). These results suggest that after training with our framework, the model learns not just surface-level patterns for specific tasks, but more generalizable skills for instruction understanding and constraint satisfaction. This is primarily attributed to the reward-based optimization paradigm in the framework, which drives the model to master the universal skill of "following instructions," rather than memorizing training set distributions, allowing it to quickly adapt to new, unseen task types.

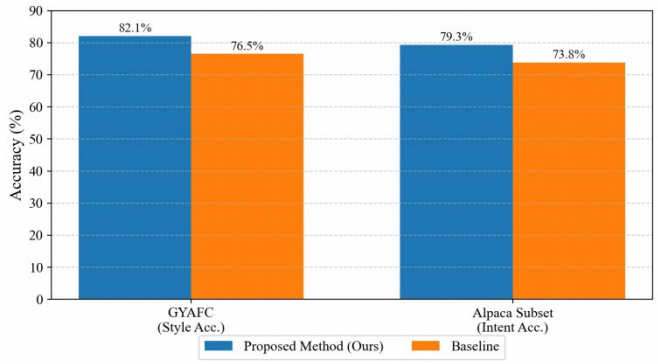

Figure 7. Zero-Shot Cross-Task Generalization Performance

Overall, the robustness experiments show that the proposed model demonstrates superior stability and adaptability compared to the baseline models when facing imperfectly expressed instructions or new task requirements. This confirms that the proposed framework not only achieves high-precision control under ideal conditions but also has the potential for reliable application in complex real-world scenarios.

4.6. Ablation Study

To validate the necessity of the three core components in the proposed framework, we conducted a systematic ablation study, with results shown in Table 7. We used the supervised fine-tuning model (Base SFT Model) as the baseline and progressively added or removed modules to analyze their contributions.

Table 7. Ablation Study Results (on Instruction-Gen Test Set)

Model Configuration	Intent Accuracy ↑	Constraint Satisfaction ↑	BLEU ↑	ROUGE-L ↑
Base SFT Model	68.3 ± 2.5	71.5 ± 3.1	32.1 ± 1.8	44.5 ± 2.2
Base SFT + Fine-grained Aligner	72.1 ± 2.2 (+3.8)	75.0 ± 2.8 (+3.5)	32.5 ± 1.7 (+0.4)	44.9 ± 2.1 (+0.4)
Base SFT + Constraint-Aware Reinforcement Controller (PPO)	77.8 ± 1.9 (+9.5)	81.2 ± 2.5 (+9.7)	33.8 ± 1.7 (+1.7)	46.1 ± 2.0 (+1.6)
w/o Instruction Parser (Simple Concatenation)	79.5 ± 2.0 (-5.2)	83.1 ± 2.3 (-5.2)	33.1 ± 1.9 (-1.8)	45.5 ± 2.2 (-1.7)
Full Model (Ours)	84.7 ± 1.5	88.3 ± 1.7	34.9 ± 1.6	47.2 ± 1.8

From Table 7, it is evident that each module contributes clearly and irreplaceably to the final performance. First, comparing the Base SFT Model with the model incorporating the Fine-grained Aligner, the latter achieves approximately a 3.8 and 3.5 percentage point improvement in Intent Accuracy and Constraint Satisfaction, respectively, while the BLEU/ROUGE improvements are minor. This indicates that the Fine-grained Aligner primarily enhances the local semantic alignment accuracy of the generated text with the instruction, especially for subtle requirements in long instructions, with a limited impact on fluency.

Next, comparing the Base SFT Model with the model including the Constraint-Aware Reinforcement Controller (PPO), we observe a significant increase in controllability metrics (Intent Accuracy and Constraint Satisfaction), with improvements of 9.5 and 9.7 percentage points. This confirms that the Constraint-Aware Reinforcement Controller (PPO) is key to improving global controllability by transforming abstract constraints into learnable reward signals, directly driving the model to learn how to satisfy user instructions.

Third, when the instruction parser is removed and simple concatenation of instructions and constraints is used as input (i.e., w/o Instruction Parser), the model performance, especially Intent Accuracy, significantly decreases by 5.2 percentage points. This highlights that the instruction parser is crucial for accurately understanding and structuring complex instructions, providing clear task representations for effective optimization.

Finally, the full model, integrating all modules, achieves the best performance across all metrics. The performance improvement is not just the sum of the contributions of each module, particularly in controllability metrics, where it outperforms the "Base SFT + PPO" baseline. This reflects the synergistic effect between the three modules: the instruction parser provides clear structured representations, offering explicit optimization targets for reinforcement learning; the fine-grained aligner's local supervision guides the reinforcement learning process to converge more precisely, together achieving superior global alignment and local consistency.

5. Discussion and Conclusion

This study proposes a unified framework for scalable and controllable English text generation by integrating structured instruction parsing, constraint-aware reinforcement learning, and fine-grained semantic alignment. The experimental results demonstrate consistent improvements in intent accuracy and constraint satisfaction while maintaining competitive generation quality and diversity. These findings indicate that effective controllability requires not only global objective

optimization but also structured representation and local alignment supervision.

From a methodological perspective, the performance gains can be attributed to the coordinated interaction of three components. The instruction parser transforms unstructured instructions into explicit task representations, reducing ambiguity in multi-constraint scenarios. The constraint-aware reinforcement controller directly optimizes alignment-related objectives through computable rewards, addressing the mismatch between likelihood-based training and deployment requirements. Meanwhile, the fine-grained aligner provides token-level supervision via cross-attention mechanisms, mitigating local semantic drift during long-form generation. The ablation results confirm that these modules contribute complementary benefits rather than isolated improvements.

In practical terms, the proposed framework is particularly suitable for real-world applications where strict adherence to user constraints is required, such as intelligent writing assistants, customer-service dialogue systems, and enterprise document automation. The modular and loosely coupled architecture also facilitates scalable deployment in distributed environments, allowing instruction parsing, policy optimization, and alignment supervision to be parallelized or independently optimized.

Nevertheless, several limitations remain. First, the framework still struggles with logically contradictory or commonsense-intensive instructions, indicating limited reasoning capability beyond reward-driven optimization. Second, the integration of multiple modules increases training complexity and computational overhead. Although the design supports distributed scalability, efficiency optimization remains an important direction for future work. Third, while experiments cover both custom and public datasets, broader domain diversity and ultra-long instruction scenarios warrant further evaluation.

Future research will focus on three aspects: (1) integrating external knowledge or reasoning-enhanced modules to better handle semantic contradictions and complex logical requirements; (2) improving computational efficiency through model compression, distillation, or system-level co-design; and (3) extending the framework to multilingual and multimodal generation tasks.

Overall, this work provides a systematic and scalable solution for multi-constraint controllable text generation, advancing both methodological alignment strategies and practical deployment feasibility in large-scale language generation systems.

References

[1] Khan, S., Serajuddin, M., Hasan, Z., Alvi, S. A. M., Ayub, R., & Sharma, A. (2023, December). Natural Language Generation (NLG) with Reinforcement Learning (RL). In International Conference on Artificial Intelligence and Speech Technology (pp. 303-318). Cham: Springer Nature Switzerland.

[2] Wu, Y. (2024). Large language model and text generation. In Natural language processing in biomedicine: A practical guide (pp. 265-297). Cham: Springer International Publishing.

[3] Yao, Q., Fang, F., Chen, Y., Liu, J., Mo, H., & Ao, Y. (2025). AI Large Models for Power System: A Survey and Outlook. IET Smart Energy Systems, 1(1), 3-21.

[4] Lin, H., Liu, Y., Li, S., & Qu, X. (2023). How generative adversarial networks promote the development of intelligent transportation systems: A survey. IEEE/CAA journal of automatica sinica, 10(9), 1781-1796.

[5] Tang, K. H., Ghanem, M. C., Gasiorowski, P., Vassilev, V., & Ouazzane, K. (2025). Synchronisation, Optimisation and Adaptation of Machine Learning Techniques for Computer Vision in Cyber-Physical Systems: A Comprehensive Analysis. IET Cyber-Physical Systems: Theory & Applications, 10(1), e70031.

[6] Uc-Cetina, V., Navarro-Guerrero, N., Martin-Gonzalez, A., Weber, C., & Wermter, S. (2023). Survey on reinforcement learning for language processing. Artificial Intelligence Review, 56(2), 1543-1575.

[7] Zhou, W., Jiang, Y. E., Wilcox, E., Cotterell, R., & Sachan, M. (2023, July). Controlled text generation with natural language instructions. In International Conference on Machine Learning (pp. 42602-42613). PMLR.

[8] Yang, Y., Gui, D., Yuan, Y., Liang, W., Ding, H., Hu, H., & Chen, K. (2023). Glyphcontrol: Glyph conditional control for visual text generation. Advances in Neural Information Processing Systems, 36, 44050-44066.

[9] Goyal, R., Kumar, P., & Singh, V. P. (2023). A Systematic survey on automated text generation tools and techniques: application, evaluation, and challenges. Multimedia Tools and Applications, 82(28), 43089-43144.

[10] Chilamkurthi, V., Agarwalla, B., & Kumar, K. S. (2024, December). Empowering Virtual Assistant Capabilities by Leveraging Generative Adversarial Networks (GANs) for Advancements in Deep Learning with NLP (Natural Language Processing). In International Conference on Biologically Inspired Techniques in Many-Criteria Decision-Making Technologies (pp. 243-253). Cham: Springer Nature Switzerland.

[11] Scotti, V., Sbattella, L., & Tedesco, R. (2023). A primer on seq2seq models for generative chatbots. ACM Computing Surveys, 56(3), 1-58.

[12] Chen, J., Liu, Z., Huang, X., Wu, C., Liu, Q., Jiang, G., ... & Chen, E. (2024). When large language models meet personalization: Perspectives of challenges and opportunities. World Wide Web, 27(4), 42.

[13] Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., ... & Gadekallu, T. R. (2024). Gpt (generative pre-trained transformer)—A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. IEEE access, 12, 54608-54649.

[14] Lu, L., Liu, Y., Xu, W., Li, H., & Sun, G. (2023). From task to evaluation: an automatic text summarization review. Artificial Intelligence Review, 56(Suppl 2), 2477-2507.

[15] Zeng, B., Lyu, C., Liu, S., Zeng, M., Wu, M., Ni, X., ... & Zhang, K. (2025, July). Marco-Bench-MIF: On Multilingual Instruction-Following Capability of Large Language. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 24058-24072).

[16] Gehrmann, S., Clark, E., & Sellam, T. (2023). Repairing the cracked foundation: A survey of obstacles in evaluation

practices for generated text. Journal of Artificial Intelligence Research, 77, 103-166.
[17] Falaki, A. A., & Gras, R. (2025). A novel unsupervised fine-tuning method for text summarization, and highlighting the limitations of ROUGE score. Machine Learning with Applications, 100666.
[18] Troiano, E., Velutharambath, A., & Klinger, R. (2023). From theories on styles to their transfer in text: Bridging the gap with a hierarchical survey. Natural Language Engineering, 29(4), 849-908.
[19] Qiu, J., Fang, Q., & Kang, W. (2025). Towards controllable and explainable text generation via causal intervention in LLMs. Electronics, 14(16), 3279.
[20] Jeong, H., Lee, H., Kim, C., & Shin, S. (2024). A survey of robot intelligence with large language models. Applied Sciences, 14(19), 8868.
[21] Yang, C., & Fang, Q. (2025). Edge-AI Enabled Resource Allocation for Federated Learning in Cell-Free Massive MIMO-Based 6G Wireless Networks: A Joint Optimization Perspective. Electronics, 14(19), 3938.
[22] Zhou, J., Gao, L., Lu, C., & Yao, X. (2025). Collaborative optimization of manufacturing service allocation via multi-task transfer learning evolutionary approach. Journal of Intelligent Manufacturing, 36(3), 1761-1779.
[23] Li, C., Zhang, M., Mei, Q., Kong, W., & Bendersky, M. (2024, May). Learning to rewrite prompts for personalized text generation. In Proceedings of the ACM Web Conference 2024 (pp. 3367-3378).
[24] Rame, A., Couairon, G., Dancette, C., Gaya, J. B., Shukor, M., Soulier, L., & Cord, M. (2023). Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. Advances in Neural Information Processing Systems, 36, 71095-71134.
[25] Gao, X., & Fang, Q. (2025). Multi-granularity sentiment analysis and learning outcome prediction for Chinese educational texts based on transformer architecture. Discover Artificial Intelligence, 5(1), 212.
[26] Xie, Y., & Fang, Q. (2025). An energy-aware generative AI edge inference framework for low-power IoT devices. Electronics, 14(20), 4086.
[27] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140), 1-67.
[28] Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.
[29] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., ... & Hashimoto, T. B. (2023, June). Stanford alpaca: An instruction-following llama model.
[30] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., ... & Christiano, P. F. (2020). Learning to summarize with human feedback. Advances in neural information processing systems, 33, 3008-3021.