LLM-based multi-agent poetry generation in non-cooperative environments

Ran Zhang[1,3] *and Steffen Eger*[2,3]
[1] University of Mannheim
[2] University of Technology Nuremberg (UTN)
[3] Natural Language Learning and Generation (NLLG) Lab

ABSTRACT

Keywords:
poetry generation,
social learning,
multi-agent system

Despite substantial progress in large language models (LLMs) for automatic poetry generation, LLM-generated poetry often lacks diversity, and the training process differs greatly from human learning. Under the rationale that the poetry generation systems should learn more like humans and produce more diverse and novel outputs, we introduce a social learning-based framework that emphasizes non-cooperative interactions besides cooperative interactions to encourage diversity. Our experiments represent the first attempt at LLM-based multi-agent poetry generation in non-cooperative environments, employing both TRAINING-BASED agents (GPT-2) and PROMPT-BASED agents (GPT-3 and GPT-4). Evaluation on 96K generated poems demonstrates that our framework improves the performance of TRAINING-BASED agents, yielding a 3.0–3.7 percentage point (pp) increase in diversity and a 5.6–11.3 pp increase in novelty, as measured by distinct and novel n-grams. Poems generated by TRAINING-BASED agents also exhibit clear group divergence in lexicon, style, and semantics. PROMPT-BASED agents likewise benefit from non-cooperative environments. However, these agents show a decrease in lexical diversity over time and fail to demonstrate the intended group-based divergence within the social network. Our work argues for a paradigm shift in creative tasks such as automatic poetry

generation to include social learning processes (via LLM-based agent modeling) similar to human interaction.

1 INTRODUCTION

Autonomous agents driven by large language models (LLMs) have made substantial progress in various domains, including complex task-solving (Li *et al.* 2023b), reasoning (Lin *et al.* 2023; Du *et al.* 2024), and simulation (Wang *et al.* 2024a). Studies have shown that interactive communication via multi-agent systems can yield emergent behaviors (Park *et al.* 2023), enhanced task performance (Zhuge *et al.* 2025), better evaluation (Chan *et al.* 2024), and assistance in open-ended generation tasks (Zhu *et al.* 2023), to name a few benefits. Despite these advances, the exploration of LLM-based agents for creative tasks such as poetry generation remains limited (Chakrabarty *et al.* 2023). This paper presents the first experiment on multi-agent poetry generation powered by LLMs.[1] We introduce a framework that emphasizes non-cooperative environments to enhance both the diversity and novelty of generated poetry.

Why poetry generation?
Despite advancements in LLMs, generating poetry remains a difficult task due to the complex interplay of style, meaning, and human emotion (Chakrabarty *et al.* 2021; Mahbub *et al.* 2023). To produce human-like poetry, models must demonstrate not only linguistic competence such as the understanding of semantics and grammar, but also mastery of stylistic elements such as rhyme, meter, and imagery (Zhipeng *et al.* 2019; Belouadi and Eger 2023; Ma *et al.* 2023). Indeed, current systems still face important challenges. Most existing models are fine-tuned for specific styles or topics (Lau *et al.* 2018; Van de Cruys 2020; Tian and Peng 2022). LLMs often struggle to create diverse and aesthetically pleasing poetry similar to human writing in

[1] The code and data are publicly available at `https://github.com/zhangr2021/Multiagent_poetry`.

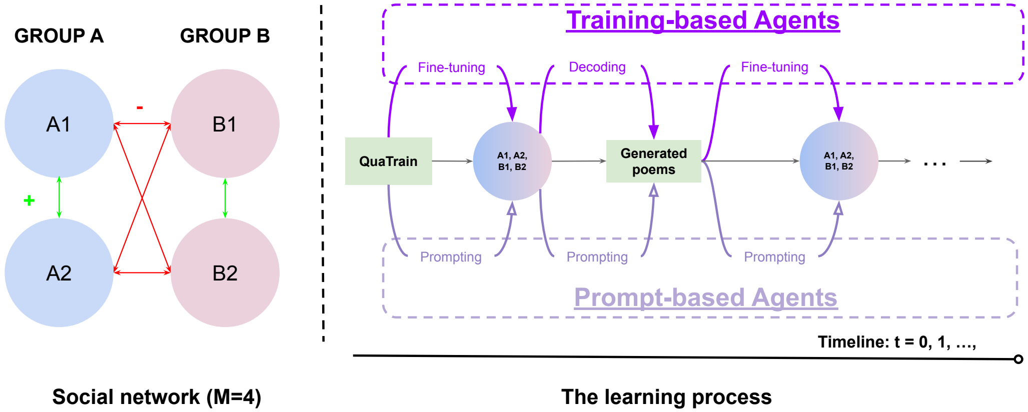


Figure 1: Illustration of the predefined social network ($M = 4$) and a high-level overview of the learning process for TRAINING-BASED agents (GPT-2) and PROMPT-BASED agents (GPT-3.5 and GPT-4). Green and red edges in the social network represent cooperative (+) and non-cooperative (-) interaction between agents, respectively

zero-shot and few-shot scenarios (Sawicki *et al.* 2023a,b). These complexities make poetry generation a suitable testbed for multi-agent approaches, which are known for their potential to solve complex tasks and enhance diversity by leveraging "mixtures of multiple poets" (Yi *et al.* 2020).

Why multi-agent systems and social learning?

The current process of generating poetry in machine learning (ML) and natural language processing (NLP) differs substantially from the way humans learn and compose poetry. Humans do not learn from fixed datasets but rather within a **social context** where they interact with others through communication (Jarvis 2012). This raises the question of whether a more human-like approach could improve poetry generation. Multi-agent systems, widely applied in social simulation (Park *et al.* 2023; Chuang *et al.* 2024a), are suitable to address this question: they allow us to model social networks with LLM-based agents and simulate the human learning process for complex creative tasks. Furthermore, as one of the most fundamental elements for creative composition, "divergent" thinking (the ability to deviate from established norms) is crucial for both human and computational creativity (Elgammal *et al.* 2017; Brinkmann *et al.* 2023; Wingström *et al.* 2023).

This provides a rationale for adopting a social learning process that incorporates not only collaboration but also opposition as a key source of divergence (Eger 2016; Shi *et al.* 2019). Unlike previous studies that focus on debates and arguments aimed at thought correction (Chan *et al.* 2024), our approach integrates opposition as a deliberate mechanism to amplify divergence and enhance diversity.

Why non-cooperative environments?

Our emphasis on opposition is further motivated by the prevalence of non-cooperative interactions across many areas of human activity. For example, the political arena is often characterized by forms of opposition between individual parties or, in a wider context, "countercultures" rebelling against the establishment. Moreover, one defining property of literature/art/philosophy is the desire to distinguish oneself from previous or contemporary "competitors". For example, Hemingway's "iceberg" writing style was substantially different from the more sentimental style of his predecessors (Baker 1972); impressionist artists have challenged the standards of painting set by the conventional art community with new content and styles; and philosophers often form opposing schools of thought, as with Schopenhauer's critique of Hegel, or the division into Kantians, Neoplatonists, and others (Janaway 2002). In computational social science, the terms "antagonistic", "non-cooperative", or "negative relationships" are used to describe such behavior (Amirkhani and Barshooi 2022). In ML or NLP, such behavior remains rarely explored or systematically modeled (Gautier *et al.* 2022; Lei *et al.* 2022).

How do we utilize LLM-based agents for poetry generation?

We build our social learning framework on a predefined social network that governs the interactions among agents, as shown in Figure 1. This network facilitates both cooperative interactions ("Poets appreciate each other's work and learn from the others") and non-cooperative ones ("Poets dislike each other's work and deviate from each other"). We implement our learning framework with two types of agents. First, we employ PROMPT-BASED conversational agents (GPT-4) to explore whether *the framework can enhance the diversity of poetry generation in zero-shot or few-shot settings.* Second, we use TRAINING-BASED agents (GPT-2) to investigate how various training and decoding configurations, such as training losses and the number of interactive agents, affect performance and to identify *which strategies are the most effective in non-cooperative environments for poetry generation.* Our framework is composed of three main components: (1) the social network, (2) the learning process, and (3) the learning strategy. The main contribution of this work is a unified social learning framework that systematically integrates multi-agent modeling approaches for poetry generation, with a particular emphasis on non-cooperative environments.

Training-based agents in non-cooperative environments generate poetry with increasing diversity and novelty over iterations. Findings in Section 5 indicate that, for TRAINING-BASED agents, our framework enhances the generation process, resulting in increasing diversity and novelty as measured by the percentage of distinct and novel n-grams. The generated poetry from TRAINING-BASED agents also exhibits group-level divergence in terms of lexicon, style, and semantics aligned with their predefined group affiliation. Further analysis in Section 6 also shows that non-cooperative conditions boost diversity for PROMPT-BASED agents. However, PROMPT-BASED agents in our framework do not exhibit any group-based divergence and are prone to generating poetry with homogeneous styles over time.

RELATED WORK 2

Our research is related to the following: (1) *interaction among LLM-based agents* where we focus on the categorization and dynamics of agent interaction; (2) *language model ensemble and controlled text generation* (CTG) which provide methodological foundations for modeling and steering such interactions; (3) *poetry generation.*

Interaction among LLM-based agents. Interactions among agents can be broadly categorized as either cooperative or non-cooperative. Very often, agents communicate cooperatively, aiming to reach joint decisions through mechanisms such as back-and-forth communication (Li *et al.* 2023b), majority voting (Hamilton 2023), or a combination of both (Zhuge *et al.* 2025). Non-cooperative interactions, although less common, can enhance response quality through debate or argument among agents (Chan *et al.* 2024; Du *et al.* 2024).

While most studies treat interactions among agents as fixed and time-invariant, some researchers delve into the dynamics of agent interactions. Autogen (Wu *et al.* 2024) enables dynamic group chats that guide the flow of agents' interactions during conversation. Liu *et al.* (2024) similarly propose a dynamic interaction framework that employs an optimization algorithm to select the most suitable agents at inference time. On the one hand, Chuang *et al.* (2024a) find that despite these dynamic interaction settings, LLMs tend to align with factual information regardless of predefined personas and initial states, which limits the simulation of opinion dynamics using LLM-based agents. On the other hand, recent studies focusing on dynamics of group interactions indicate that incorporating chain-of-thought reasoning, detailed personas, and fine-tuning LLMs can enhance the agents' ability to replicate human-like group behavior (Chuang *et al.* 2024b).

Our work differs in several key aspects. First, we build a social network with group affiliations to obtain a more human-like learning process. Second, we propose a framework that involves two forms of interaction, with an emphasis on non-cooperative communication. Third, we fine-tune TRAINING-BASED agents and use consecutive prompting with detailed personas for PROMPT-BASED agents.

Language model ensemble and controlled text generation. Our proposed framework requires (1) an ensemble of language models (LMs) and (2) generated outputs reflecting general poetic styles relevant to poetry generation. This design connects to prior research on *LM ensembles* and *controlled text generation (CTG).*

LM ensembles can be divided into two subcategories: (1) conversational ensembles which do not involve parameter training (Wang *et al.* 2023), and (2) fine-tuning-based ensembles at the neural network level (Shazeer *et al.* 2016) and at the output level (Dekoninck *et al.* 2024; Jiang *et al.* 2023). Conversational ensembles are prompt-based and are often utilized for reasoning tasks, where LLMs ensemble their own responses (i.e., self-ensemble) (Wang *et al.* 2023; Fu *et al.* 2023). In very recent work, Lu *et al.* (2024) combine multiple small conversational models in a parameter-efficient and interpretable way.

According to the A/B test, the ensemble outperforms ChatGPT. In contrast, fine-tuning-based ensembles can operate at the neural network or output level. While *neural network ensembles* typically require massive training or fine-tuning through extensive datasets and resources (Shazeer *et al.* 2016; Jiang *et al.* 2024), a mixture of smaller neural network modules such as adapters becomes a viable solution in resource-constrained settings (Wang *et al.* 2022; Chronopoulou *et al.* 2023).

CTG, the task of generating texts conditioned on specific attributes such as emotion (Firdaus *et al.* 2022; Ruan and Ling 2023), topic (Dathathri *et al.* 2020; Wang *et al.* 2019), toxicity avoidance (Liu *et al.* 2021), style (Belouadi and Eger 2023; Shao *et al.* 2021), debiasing (Dinan *et al.* 2020; Sheng *et al.* 2020), etc., is also relevant to our study. While CTG partially overlaps with LM ensembles in its mechanisms for control, it addresses a broader range of use cases focused on attribute-specific text generation. Unlike LM ensembles, which typically involve multiple models, CTG can be applied to both single-model and multi-model setups.

CTG methods operate at the training/fine-tuning stage or at the inference stage. Similar to LM ensembles, fine-tuning-based CTG approaches introduce additional modules such as task-specific adapters (Ribeiro *et al.* 2021; Lin *et al.* 2021) to achieve parameter-efficient controllability. In addition to architectural modifications, various loss functions have been utilized to enhance controllability during fine-tuning. Beyond the standard CROSS-ENTROPY (*CE*) loss used in text generation, alternative objectives have been explored to better steer model outputs. For example, Qian *et al.* (2022) incorporate a CONTRASTIVE loss, which is effective for detoxification but only partially improves sentiment control. Likewise, Zheng *et al.* (2023) apply a CONTRASTIVE loss on sequence likelihood to reduce the probability of generating negative samples.

Inference-stage CTG approaches are increasingly favored in the era of LLMs (Jiang *et al.* 2023; Wang *et al.* 2024b; Dekoninck *et al.* 2024). Reranking candidate outputs is a popular strategy: Jiang *et al.* (2023), for instance, first rerank complete candidate responses from multiple LLMs and then fuse the top-*K* outputs. Other studies modify the decoding process directly by reranking next-token distributions using discriminators (Dathathri *et al.* 2020) or combining logits from expert and anti-expert models (Liu *et al.* 2021; Dekoninck *et al.* 2024).

Generally, inference-stage operations offer strong controllability over generated text with lower computational and time costs, though they may slightly degrade the output quality (Dathathri *et al.* 2020). In contrast, training-/fine-tuning-stage operations preserve text quality but provide weaker control over specific attributes (Zhang *et al.* 2023).

For our use case of poetry generation, we consider both prompt-based conversational ensembles and fine-tuning-based CTG methods. In the fine-tuning-based setting, we employ a joint strategy that operates at both the training and decoding stages: (1) during the training, we use standard CROSS-ENTROPY loss and CONTRASTIVE loss to fine-tune adapters for improved text quality and efficiency; and (2) during decoding, we enhance controllability in non-cooperative environments using CTG methodologies.

Automatic poetry generation. Early attempts at automatic poetry generation mainly rely on grammatical rules (Oliveira 2012), statistical rules (Jiang and Zhou 2008; Greene *et al.* 2010), and neural architectures such as recurrent neural networks (RNNs) (Zhang and Lapata 2014; Ghazvininejad *et al.* 2017; Wöckener *et al.* 2021), especially RNN-based encoder-decoder architecture (Wang *et al.* 2016; Lau *et al.* 2018; Yan 2016). More recent models have shifted toward transformer-based architectures (Tian *et al.* 2021; Shao *et al.* 2021).

Building on the transformer, variants of the GPT family have achieved outstanding performance across many NLP tasks. However, their effectiveness in poetry generation remains debated. Studies such as Bena and Kalita (2019), Liao *et al.* (2019), and LC (2022) fine-tune GPT-2 with additional components such as emotion, form, and theme, and report moderate to high-quality poetic outputs, according to human evaluations. Köbis and Mossink (2021) find that *zero-shot* GPT-2 can produce human-like poems where the best poem, according to human selection, can match human-written ones but machine-generated poems remain easily distinguishable. Similarly, Wöckener *et al.* (2021) point out that, like RNN-based models, GPT-2 struggles to capture poetry-specific features such as rhyme and meter.

To address these limitations, Belouadi and Eger (2023) propose BYGPT5, an end-to-end token-free model conditioned on rhyme, meter, and alliteration. The model outperforms larger models such as GPT-2, ByT5, and ChatGPT (GPT-3.5), according to both automatic and human evaluation. In addition, the authors construct a customized corpus QuaTrain consisting of large-scale machine-labeled pseudo-quatrains to expand the fine-tuning dataset. Moreover, Sawicki *et al.* (2023b) report that GPT-3, when fine-tuned on a small corpus of 300 poems, can successfully generate high-quality poetry in the style of specific authors, whereas GPT-3.5 without fine-tuning tends to produce lower-quality poems with only superficial stylistic resemblance. Similarly, Sawicki *et al.* (2023a) find that GPT-3.5 and GPT-4, without fine-tuning, fail to generate poetry in desired styles.

Recently, interactive poetry generation has gained increasing attention for its ability to facilitate human-machine collaboration, enabling the generation of poems with more diverse styles and higher quality under specific constraints (Zhipeng *et al.* 2019; Uthus *et al.* 2022). Ma *et al.* (2023) propose a post-polishing system that refines GPT-2 outputs based on constraints from humans. CoPoet from Chakrabarty *et al.* (2022) fine-tunes a pretrained T5 model with <instruction, generation> pairs to enable poetry generation guided by human instructions. Their study shows that the fine-tuned T5 model is not only competitive with the larger InstructGPT but also collaborates effectively with humans to produce higher-quality poems.

In our study, we use GPT-2 as our base model, as it offers a good balance between parameter efficiency and language proficiency. Unlike most poetry generation objectives that optimize for a limited set of poetic features, we train on poems of arbitrary styles, initializing our models with random samples from the QuaTrain corpus that contain pseudo-poetic characteristics. We further explore the potential of GPT-3.5 and GPT-4 in an interactive setting within a multi-agent system through prompting.

3 SOCIAL LEARNING FRAMEWORK FOR POETRY GENERATION

This section introduces our social learning framework for poetry generation. The recent development of LLMs has spurred several attempts to simulate the social learning processes of humans through LLM-based agents (Li *et al.* 2023a; Chuang *et al.* 2024a; Gao *et al.* 2025). Inspired by these efforts, our framework adopts a social network-based approach to poetry generation, investigating whether a more human-like learning process (i.e., social learning) can enhance poetry generation. Our approach differs from the previous studies in two aspects: (1) it is built on a signed network where agents interact not only cooperatively but also non-cooperatively; and (2) it introduces a unified learning framework that accommodates both PROMPT-BASED agents (GPT-3.5 and GPT-4) and TRAINING-BASED agents (GPT-2). The framework consists of three main components: (1) the social network, (2) the learning process, and (3) the learning strategy. Each component is described in detail below.

3.1 *The social network*

We consider a signed social network with M LLM-empowered agents, where each link between two agents is assigned either a positive or a negative sign (Leskovec *et al.* 2010; Eger 2016; Shi *et al.* 2019). The M agents are divided into two groups, as illustrated in Figure 1. Agents within the same group are referred to as 'in-group' agents, while agents from different groups are termed 'out-group' agents.

Two types of interaction are defined based on group affiliation: (1) 'in-group' agents cooperate closely with one another as "friends" (positive sign); and (2) 'out-group' agents act as "foes," adjusting their "opinions" in a non-cooperative manner (negative sign). The learning process involving 'in-group' agents is referred to as **positive learning**, whereas that involving 'out-group' agents is **negative learning**. Simultaneous learning from both 'in-group' and 'out-group' is termed **joint learning**.

In our application to poetry generation, the agents correspond to pretrained LLMs, which we conceptualize as poets belonging to two groups. The ‘in-group’ poets appreciate one another’s work and seek to learn from each other’s styles, whereas the ‘out-group’ poets dislike one another’s work and deliberately differentiate their styles.

3.2 *The learning process*

We now describe the learning process among agents based on the social network shown in Figure 1 and outlined in Algorithm 1. All notations used in this section are summarized in Table 1. The learning process represents the high-level communication procedure among agents.

Algorithm 1: Social learning process for poetry generation

```
for $t \leftarrow 1$ to $T$ do
    Initialize an empty set $S_t$ to store the generated poems at iteration $t$;
    foreach agent $\mathscr{A}_i$                              // $i \in \{1,2,\dots,M\}$ do
        $O_t^{\mathscr{A}_i} \leftarrow F_{\text{generate}}(\mathscr{A}_i, \mathscr{A}^-, \mathscr{A}^+)$ to generate $N$ poems;
        Add $O_t^{\mathscr{A}_i}$ to $S_t$;
    end
    foreach agent $\mathscr{A}_i$                              // $i \in \{1,2,\dots,M\}$ do
        if $t > 1$ then
            $\mathscr{A}_i \leftarrow F_{\text{update}}(S_t, S_{t-1})$;
        else
            $\mathscr{A}_i \leftarrow F_{\text{update}}(S_t)$;
        end
    end
end
```

We begin with pretrained LLM-based agents $a_1, a_2, \dots$ belonging to group A and $b_1, b_2, \dots$ belonging to group B (see Section 4 for details on agent initialization). The learning process consists of two phases: the UPDATE phase and the GENERATE phase, which together define the learning strategy $(F_{\text{update}}, F_{\text{generate}})$. The two functions jointly organize the positive, negative, and joint learning processes.

Specifically, F_{generate} denotes the function that generates output poems O, while F_{update} represents the learning function that updates

Table 1: Notations

Variable	Definition
$a_1, a_2, \ldots$	agents in group A
$b_1, b_2, \ldots$	agents in group B
M	the total number of agents
$\mathscr{A}_i$	the target agent $\mathscr{A}_i \in \{a_1, b_1, a_2, b_2, \ldots\}$ where $i \in \{1, 2, \ldots, M\}$
$\mathscr{A}_i, \mathscr{A}^+, \mathscr{A}^-$	the agent tuple: (the target agent, the 'in-group' agents of the target agent, the 'out-group' agents). For example, $(a_1, [a_2, a_3], [b_1, b_2])$
$P_{\mathscr{A}_i}, P_{*^+}, P_{*^-}$	the conditional probability distribution for the next token of agent $\mathscr{A}_i$, agents $*^+ \in \mathscr{A}^+$ and agents $*^- \in \mathscr{A}^-$
N	the total number of generated poems per iteration per agent
T	the total number of iterations
t	the iteration number $t \in \{1, 2, \ldots, T\}$
$o^{\mathscr{A}_i}$	a poem generated by agent $\mathscr{A}_i$
$O_t^{\mathscr{A}_i}$	the set of poems generated by agent $\mathscr{A}_i$ at iteration t
S_t	the set of all generated poems at iteration t
F_{generate}	a generate function of agent tuple $(\mathscr{A}_i, \mathscr{A}^-, \mathscr{A}^+)$
F_{update}	an update function based on the latest generated outputs S_t and S_{t-1}
t_g	the generation time at the decoding stage
x_{t_g}	the token at generation time t_g
$\boldsymbol{x}_{<t_g}$	the input sequence at generation time t_g
$\#\mathscr{A}$	the number of interactive agents at the decoding stage

agents with new knowledge derived from the generated poems. We denote the i^{th} agent as $\mathscr{A}_i$, where $i \in \{1, 2, \ldots, M\}$. The 'out-group' agents for $\mathscr{A}_i$ are denoted as $\mathscr{A}^-$, and the 'in-group' agents as $\mathscr{A}^+$. The generate function is thus expressed as $F_{\text{generate}}(\mathscr{A}_i, \mathscr{A}^-, \mathscr{A}^+)$.

At each iteration t, the learning process begins with the GENERATE phase. Each agent $\mathscr{A}_i$ generates a set of N poems through function $F_{\text{generate}}(\mathscr{A}_i, \mathscr{A}^-, \mathscr{A}^+)$. We iterate over all agents and collect the generated poems into a set S_t, which contains outputs O_t from iteration t. Next, each agent $\mathscr{A}_i$ updates its knowledge by learning cooperatively, non-cooperatively, or jointly from others based on poems from the current iteration S_t and the previous iteration S_{t-1}.[2] The update operation is defined as $F_{\text{update}}(S_t, S_{t-1})$. We iterate over all agents $\mathscr{A}_i$ for $i \in \{1, 2, \ldots, M\}$ until each has been updated, and repeat the overall learning process for T iterations.

The learning strategy 3.3

As described in Section 3.2, the learning process involves positive, negative, and joint learning, operating across both the GENERATE and the UPDATE phases. We now detail the learning strategies for TRAINING-BASED agents and PROMPT-BASED agents.

For TRAINING-BASED agents, the learning strategies contain a fine-tuning method applied during the UPDATE phase and a decoding method used in the GENERATE phase. For PROMPT-BASED agents, both phases are carried out through prompting. Table 2 summarizes the learning strategies for both agent types.

Table 2: Learning strategies for TRAINING-BASED agents and PROMPT-BASED agents. $\mathscr{L}_{\text{CE}}$ and $\mathscr{L}_{\text{CL}}$ represent CROSS-ENTROPY loss and CONTRASTIVE loss. $\mathscr{A}_i, \mathscr{A}^+, \mathscr{A}^-$ denotes the target agent, its 'in-group' agents and its 'out-group' agents. $P_{\mathscr{A}_i}, P_{*^+}, P_{*^-}$ represent the conditional probability distributions of the next token for the target agent $\mathscr{A}_i$, its 'in-group' agent $*^+ \in \mathscr{A}^+$ and 'out-group' agent $*^- \in \mathscr{A}^-$, as defined in Equation (1)

Type of agents	Strategy	Positive learning ($\mathscr{A}_i, \mathscr{A}^+$)	Negative learning ($\mathscr{A}_i, \mathscr{A}^-$)	Joint learning ($\mathscr{A}_i, \mathscr{A}^+, \mathscr{A}^-$)
TRAINING-BASED	decoding	–	$P_{\mathscr{A}_i}, P_{*^-}$	$P_{\mathscr{A}_i}, P_{*^+}, P_{*^-}$
	fine-tuning	$\mathscr{L}_{\text{CE}}$	–	1) $\mathscr{L}_{\text{CL}}$ 2) conditioned $\mathscr{L}_{\text{CE}}$
PROMPT-BASED	prompting	chain-prompting		joint-prompting

[2] We use generations from both the current and previous iteration to expand the fine-tuning dataset, allowing agents to integrate recent knowledge while mitigating potential catastrophic forgetting (Biesialska *et al.* 2020).

3.3.1 Training-based agents

We first introduce the decoding strategy for the GENERATE phase, which employs reranking techniques. We then detail the fine-tuning strategy for the UPDATE phase which utilizes the CONTRASTIVE loss alongside the standard CROSS-ENTROPY loss.

Decoding strategy at the GENERATE phase. We adopt the DEXPERT framework, in which models learn through a comparison-and-contrast mechanism (Liu *et al.* 2021). During decoding, the probability distribution of the next token is reranked such that the target agent $\mathscr{A}_i$ generates its next token by jointly considering the probability distributions of itself, its 'in-group' agents $\mathscr{A}^+$, and 'out-group' agents $\mathscr{A}^-$.

The total number of interactive agents is given by $\#\mathscr{A}$. The number of interactive agents involved at the decoding stage can vary depending on the selected subsets from $\mathscr{A}^+$ and $\mathscr{A}^-$, denoted as $\mathscr{A}^+_\#$ and $\mathscr{A}^-_\#$, respectively.[3] This flexibility allows the system to dynamically adjust the number of interactive agents participating in the decoding process, offering adaptability to different requirements or computational constraints. The detailed formulation of the decoding strategy is provided below.

Given the input sequence at generation time t_g (where g indicates the *generation* stage), denoted as $\boldsymbol{x}_{<t_g}$, we predict the next token x_{t_g} for the target agent $\mathscr{A}_i$ by combining outputs from the selected 'in-group' agents $\mathscr{A}^+_\#$ and 'out-group' agents $\mathscr{A}^-_\#$.

We first obtain the conditional logit scores of all models denoted by $l_{\mathscr{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g}), l_{*^+}(x_{t_g}|\boldsymbol{x}_{<t_g}), l_{*^-}(x_{t_g}|\boldsymbol{x}_{<t_g})$, where $*^+ \in \mathscr{A}^+_\#$ represents an interactive 'in-group' agent and $*^- \in \mathscr{A}^-_\#$ represents an 'out-group' agent. Each logit vector is $l_*(x_{t_g}|\boldsymbol{x}_{<t_g}) \in \mathbb{R}^{|\mathscr{V}|}$, where $\mathscr{V}$ denotes the vocabulary. The probability distribution of the next token over the vocabulary $\mathscr{V}$ for agent $*$ is given by $P_*(x_{t_g}|\boldsymbol{x}_{<t_g}) = \text{softmax}[l_*(x_{t_g}|\boldsymbol{x}_{<t_g})]$.

The probability distribution of the next token for agent $\mathscr{A}_i$ is then computed as in (1).

[3] $\#\mathscr{A} = |\mathscr{A}^+_\#|+|\mathscr{A}^-_\#|$

(1)
$$\hat{P}_{\mathscr{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g}) = \text{softmax}\left\{ l_{\mathscr{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g}) + \alpha\left[\frac{\sum_{\mathscr{A}^+_{\#}} l_{*^+}(x_{t_g}|\boldsymbol{x}_{<t_g})}{|\mathscr{A}^+_{\#}|} - \frac{\sum_{\mathscr{A}^-_{\#}} l_{*^-}(x_{t_g}|\boldsymbol{x}_{<t_g})}{|\mathscr{A}^-_{\#}|}\right]\right\}$$

Intuitively, a token x_{t_g} receives a higher probability when it is strongly supported by both the target agent $\mathscr{A}_i$ (high $P_{\mathscr{A}_i}$) and its 'in-group' peers (high P_{*^+}), while being weakly supported (or discouraged) by its 'out-group' peers P_{*^-}.

If we replace P_{*^+} with $P_{\mathscr{A}_i}$, the process considers only the 'out-group' agents $\mathscr{A}^-$, thereby modeling purely negative learning. Thus, this decoding strategy supports both *negative learning* ($P_{\mathscr{A}_i}, P_{*^-}$) and *joint learning* ($P_{\mathscr{A}_i}, P_{*^-}, P_{*^+}$) as summarized in Table 2.

Fine-tuning strategies at the UPDATE phase. We discuss the fine-tuning strategies based on the learning relationships, i.e., positive and joint learning, as summarized in Table 2.

- *Positive learning*: We adopt the conventional fine-tuning method with CROSS-ENTROPY loss ($\mathscr{L}_{\text{CE}}$) to fine-tune each agent $\mathscr{A}_i$ using poems $o \in O_{\mathscr{A}_i} \cup O_{\mathscr{A}^+}$ (poems generated by $\mathscr{A}_i$ and its 'in-group' peers $\mathscr{A}^+$). The loss function for the j^{th} poem o_j in a mini-batch is defined as:

  (2)
  $$\mathscr{L}_{\text{CE}}(\mathscr{A}_i, o_j) = -\sum_{t_g=1}^{\mathscr{T}} \log(P_{\mathscr{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g}))$$

  where $\mathscr{T}$ denotes the number of tokens in poem o_j. $P_{\mathscr{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g})$ is the conditional probability of token x_{t_g} given the preceding sequence $\boldsymbol{x}_{<t_g}$.
- *Joint learning with* CONTRASTIVE *loss*: Our social network design naturally aligns with the CONTRASTIVE learning setting, where poems from 'in-group' agents serve as positive samples and those from 'out-group' agents serve as negative samples. We employ CONTRASTIVE learning to pull closer the semantic representation of 'in-group' samples while pushing away those of 'out-group' samples. We implement the CONTRASTIVE loss SIMCSE proposed by Gao *et al.* (2021). For a mini-batch containing samples from $\mathscr{A}_i, \mathscr{A}^+, \mathscr{A}^-$, let $(o_j^{\mathscr{A}_i}, o_j^{\mathscr{A}^+}, o_j^{\mathscr{A}^-})$ denote the j^{th} triple

and $(\boldsymbol{h}_j, \boldsymbol{h}_j{}^+, \boldsymbol{h}_j{}^-)$ their corresponding representations. The CONTRASTIVE loss is defined as:

$$\text{(3)} \qquad \mathscr{L}_{\text{CL}}(\mathscr{A}_i, (\boldsymbol{h}_j, \boldsymbol{h}_j{}^+, \boldsymbol{h}_j{}^-)) = -\log \frac{e^{\text{sim}(h_j, h_j^+)/\tau}}{\sum_{k=1}^{Q} \left(e^{\text{sim}(h_j, h_k^+)/\tau} + e^{\text{sim}(h_j, h_k^-)/\tau} \right)}$$

where Q is the batch size, τ is the temperature, and $\text{sim}(h_1, h_2)$ denotes the cosine similarity $\frac{h_1^\top h_2}{\|h_1\| \cdot \|h_2\|}$. We experiment with (a) CONTRASTIVE loss alone, and (b) a joint objective combining CONTRASTIVE loss for both groups and $\mathscr{L}_{\text{CE}}$ for 'in-group' poems.

- *Joint learning with conditioned* CROSS-ENTROPY *loss*: We utilize the style constraints of Belouadi and Eger (2023), using conditions <positive> and <negative> for poems generated by 'in-group' and 'out-group' agents, respectively. We then fine-tune the agent $\mathscr{A}_i$ using the standard CROSS-ENTROPY loss.

3.3.2 Prompt-based agents

The learning strategy for PROMPT-BASED agents relies on prompting. Our prompts are constructed with three modules: (1) a profile module, which defines the role of $\mathscr{A}_i$; (2) a memory module, which stores the generated poems; and (3) an action module, which performs the generation task. Table A.1 in the Appendix lists all the prompts for both chain-prompting and joint-prompting strategies.

For PROMPT-BASED agents, the UPDATE and GENERATE phases are interdependent rather than isolated. F_{update} modifies the profile module during prompting based on the poems generated in previous iterations. Similar to the setup of TRAINING-BASED agents, we update each agent's knowledge according to different learning relationships.

The joint-prompting strategy presents positive and negative examples simultaneously but may inadvertently trigger the "don't think of an elephant" effect, causing models to fixate on the very content they are instructed to avoid. To address this issue, the chain-prompting strategy employs a two-step process that reduces the direct influence of negative prompts while still leveraging their contrastive information:

- Chain-prompting for isolated positive and negative learning: $\mathscr{A}_i$ updates its knowledge separately based on relationships with 'in-group' and 'out-group' agents. At iteration t, we ***first*** update the profile of $\mathscr{A}_i$ with poems generated from $\mathscr{A}^+$ at time $t-1$, denoted as $F_{\text{update}}(\mathscr{A}_i, \mathscr{A}^+)$. The agent $\mathscr{A}_i$ then generates a poem $o^{\mathscr{A}_i}$ following the positive learning phase (e.g., prompt: "Please read the poems from your friends carefully and compose similarly to your friend."). *Next,* we update the profile of $\mathscr{A}_i$ with both its generated poem $o^{\mathscr{A}_i}$ and a poem $o^{\mathscr{A}^-}$ sampled from the previous iteration $t-1$, denoted as $F_{\text{update}}(\mathscr{A}_i, \mathscr{A}^-)$. *Finally,* $\mathscr{A}_i$ recomposes the poem $o^{\mathscr{A}_i}$ following the negative learning phase (e.g., prompt: "Please rewrite your poem to compose dissimilarly to your foe.").
- Joint-prompting updates the profile of $\mathscr{A}_i$ with poems generated by both $\mathscr{A}^+$ and $\mathscr{A}^-$ simultaneously, denoted as $F_{\text{update}}(\mathscr{A}_i, \mathscr{A}^-, \mathscr{A}^+)$.

EXPERIMENTS 4

Agent initialization 4.1

Initialization for TRAINING-BASED agents. For our initial experiments, we set $M = 4$. We begin by further pretraining GPT-2 (medium) on a subset of the random QuaTrain corpus of size 123K (nearly 1/6 of the entire corpus). The QuaTrain corpus consists of machine-labeled pseudo-quatrains, i.e., four consecutive lines extracted from poems written by humans. These quatrains exhibit poetic characteristics such as rhyme, meter, and alliteration (see Table 3). To reduce redundancy, we exclude instances with high pairwise semantic similarity (cosine similarity > 0.7) calculated using sentence embeddings from SBERT (Reimers and Gurevych 2019).

We continue pretraining GPT-2 for 720 steps on the filtered dataset. Details of the pretraining setup and the loss curve are provided in Section A.2. Finally, we fine-tune the pretrained model on four randomly selected subsets (each containing 7.5K samples) from

Table 3: Instances from QuaTrain corpus

Instance	Rhyme	Meter	Alliteration
Who hath such beauty seen In one that changeth so? Or where one's love so constant been, Who ever saw such woe?	ABAB	iambus	0.11
Would rather seek occasion to discover How little pitiful and how much unkind, They other not so worthy beauties find. O, I not so! but seek with humble prayer	ABBC	iambus	0.05
Of pearl and gold, to bind her hands; Tell her, if she struggle still, I have myrtle rods at will, For to tame, though not to kill.	ABBB	iambus	0.10

the 123K subcorpus to initialize four distinct agents. The initialization step prepares models with a preliminary understanding of poetic structures and characteristics that are essential for subsequent learning phases.

Initialization for PROMPT-BASED agents. For PROMPT-BASED agents, we randomly sample QuaTrain instances and initialize the agent with chain-prompting and joint-prompting under the predefined profile shown in Tables 14 and 15.

4.2 *Experimental setup*

For TRAINING-BASED agents, we design experiments to examine how parameters from the fine-tuning stage (i.e., loss functions) and the decoding stage (i.e., number of agents and the scaling parameter) influence the dynamics of generation. The detailed experimental setup is summarized in Table 4.

Table 4: Experimental setup for TRAINING-BASED agents. Para_decoding and Para_fine-tuning represent parameters during the decoding and fine-tuning stage. $\#\mathscr{A}$ is the number of agents. α is the scaling parameter in Equation (1)

RQ1	Para_decoding		Para_fine-tuning			Description
	$\#\mathscr{A}$	α	$\mathscr{L}_{\text{CE}}$	$\mathscr{L}_{\text{CL}}$	*conditioned*	
$\#\mathscr{A}$ during decoding	{2,3,4}	2	X			The number of agents involved during decoding is varied. 1) $\#\mathscr{A} = 2$: negative decoding + positive fine-tuning 2) $\#\mathscr{A} > 2$: joint decoding + positive fine-tuning
α	2	{0, 1, 1.5, 2, 2.5}	X			The scaling parameter α during decoding is varied. $\alpha > 0$: negative decoding + positive fine-tuning $\alpha = 0$: positive fine-tuning only, ‘echo chamber’
fine-tuning strategy	2	2	X			Different training loss applied for $\alpha = 2$ (with negative decoding). 1) $\mathscr{L}_{\text{CE}}$: positive training with negative decoding 2) $\mathscr{L}_{\text{CL}}$ or *conditioned*: joint training with negative decoding
			X			
				X		
			X	X		
			X		X	

For PROMPT-BASED agents, we design experiments using different prompting strategies, namely chain-prompting and joint-prompting with both GPT-3.5 (gpt-3.5-turbo) and GPT-4 (gpt-4-turbo).

Evaluation 4.3

We first conduct an automatic evaluation to analyze the generation dynamics of our framework from a lexical perspective. We study lexical novelty and diversity, as these are key indicators for creative tasks such as poetry generation. We then extend the analysis by evaluating the dynamics of semantic similarity. The detailed definitions of all automatic metrics are provided below. Finally, we evaluate the quality of the generated poems using (1) a 5-point scale with human and LLM-as-a-judge evaluations to assess four key dimensions of poetry (*fluency*, *coherence*, *meaningfulness*, and *poeticness*; see Section 5.2.1 for details), and (2) direct comparison of concrete examples from the generated poetry.

Metric for lexical diversity and novelty. We measure lexical diversity using the percentage of distinct uni-grams (*distinct-1*) and bi-grams (*distinct-2*), following the definitions by Su *et al.* (2022) and Tevet and Berant (2021). The metric is formulated as: $\frac{\text{unique n-grams}(O)}{\text{total n-grams}(O)}$, where O is the set of generated poems being evaluated. To assess novelty (*novelty-1* and *novelty-2*), we follow McCoy *et al.* (2023) and Shen *et al.* (2020), computing the proportion of new uni-/bi-grams that do not appear in the pretraining set, rescaled by the total number of generated tokens.

In this way, novelty reflects the lexical deviation of the generated poems from the training data, whereas diversity captures the variety of lexical choices within the generated set.

Metric for group-based semantic similarity. Following the predefined group affiliations, we analyze the agents' group dynamics from a semantic perspective. For each pair of poems sampled within the same iteration t, we calculate their *semantic similarity* by computing the cosine similarity of their embeddings retrieved from SBERT (Reimers and Gurevych 2019). We then aggregate the similarity scores per iteration based on group affiliation (i.e., 'in-group' and 'out-group') as defined in Figure 1.

5 EXPERIMENT RESULTS

In this section, we present the results of the experiments. We first conduct automatic evaluations to analyze the generation dynamics of the agents, examining how our framework influences lexical-level diversity and novelty (Section 5.1.1) as well as group divergence in semantics (Section 5.1.2). We then evaluate the quality of the generated poetry through both Likert-scale assessments using human and LLM-as-a-judge evaluations (Section 5.2.1) and direct comparisons of representative examples (Section 5.2.2).

Automatic evaluation: The generation dynamics of agents 5.1

We generate 400 poems per agent at each iteration, using identical decoding parameters for TRAINING-BASED agents and identical prompt templates for PROMPT-BASED agents.[4] In total, we obtain more than 96K generated poems. We report the lexical diversity (*distinct-1* and *distinct-2*), novelty (*novelty-1* and *novelty-2*), and group-based semantic similarity defined in Section 4.3.[5]

Diversity and novelty 5.1.1

We compute *distinct-1/2* and *novelty-1/2* with all generated poems and average over all agents for every iteration t. The results are shown in Table 5 and Figure 2.

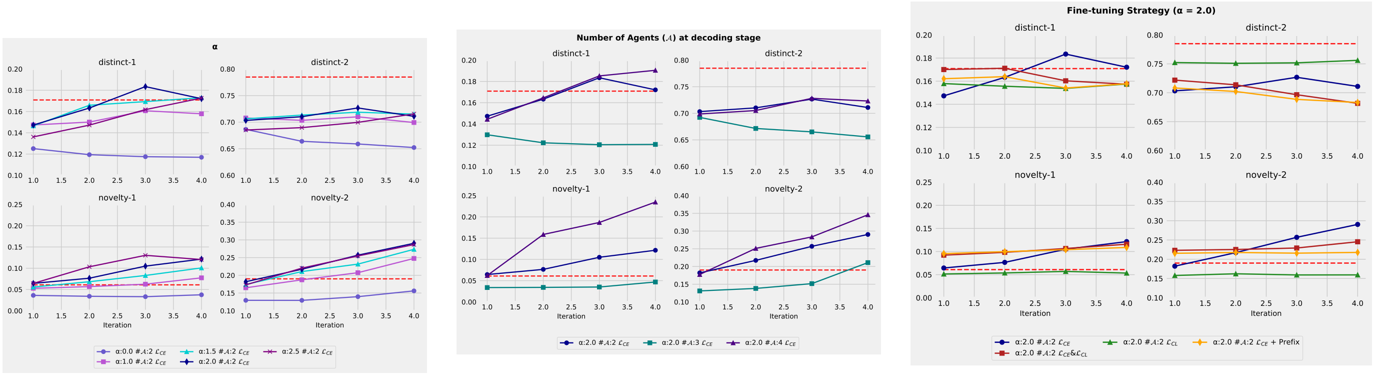
RQ1: How do different learning strategies affect the diversity and novelty of TRAINING-BASED agents?

Figure 2 shows the dynamics of agent diversity and novelty under varying training parameters. Table 5 shows the results of different experimental setups for diversity and novelty averaged over all iterations.

- The effect of the negative decoding strategy with the scaling parameter α.
  - ***Diversity*** Negative decoding combined with positive fine-tuning (α >0, $\mathscr{L}_{\text{CE}}$) strategy leads to increasing diversity over time, though the level of diversity is below the initial state at $t = 0$. Aggregatively, results from Table 5 (*varying* α)

[4] See Section A.2 in the appendix for detailed parameter settings.

[5] Due to resource limitations, we only conduct a single run of the experiments. We analyze the stability of our statistics in Section 6.1.



(a) (b) (c)

Figure 2: Dynamics of agent diversity and novelty over varying training parameters. The degree of diversity is measured by the percentage of distinct uni-grams (*distinct-1*) and bi-grams (*distinct-2*) in the generated poems. The degree of novelty is measured by the number of novel uni-grams (*novelty-1*) and bi-grams (*novelty-2*) in the generated poems compared to those in the training data scaled by the total number of generated tokens. (a) The effect of scaling parameter α in Equation (1). (b) The effect of the number of interactive agents $\#\mathscr{A}$ during the decoding stage. (c) The effect of fine-tuning strategies: $\mathscr{L}_{\text{CE}}$ and $\mathscr{L}_{\text{CL}}$ indicate CROSS-ENTROPY loss and CONTRASTIVE loss. Prefix refers to the conditioned fine-tuning. The horizontal red dashed line indicates the state of initial agents at iteration 0

Table 5: Diversity and novelty results in aggregate mean for TRAINING-BASED agents. *distinct-1* and *distinct-2* are the percentage of distinct uni-/bi-grams. *novelty-1* and *novelty-2* reflect the number of new uni-/bi-grams that do not appear in the training set rescaled by the total number of generated tokens. The highest value in each experimental setting is highlighted in **bold**. α represents the decoding scaling parameter; $\#\mathscr{A}$ is the number of interactive agents at decoding stage; $\mathscr{L}$ represents the loss function during fine-tuning. *Initialization* indicates the states of initial agents at iteration 0

α	$\#\mathscr{A}$	$\mathscr{L}$	distinct-1	distinct-2	novelty-1	novelty-2
			varying α			
0	2	$\mathscr{L}_{\text{CE}}$	0.120	0.665	0.035	0.139
1	2	$\mathscr{L}_{\text{CE}}$	0.154	0.705	0.062	0.202
1.5	2	$\mathscr{L}_{\text{CE}}$	0.164	**0.713**	0.077	0.222
2	2	$\mathscr{L}_{\text{CE}}$	**0.167**	**0.713**	0.092	**0.237**
2.5	2	$\mathscr{L}_{\text{CE}}$	0.154	0.698	**0.105**	0.234
			varying $\#\mathscr{A}$			
2	2	$\mathscr{L}_{\text{CE}}$	0.167	0.713	0.092	0.237
2	3	$\mathscr{L}_{\text{CE}}$	0.123	0.671	0.037	0.158
2	4	$\mathscr{L}_{\text{CE}}$	**0.171**	**0.714**	**0.161**	**0.264**
			varying training loss			
2	2	$\mathscr{L}_{\text{CE}}$	**0.167**	0.713	0.092	**0.237**
2	2	$\mathscr{L}_{\text{CE}} + \mathscr{L}_{\text{CL}}$	0.165	0.703	**0.103**	0.231
2	2	$\mathscr{L}_{\text{CL}}$	0.156	**0.753**	0.054	0.160
2	2	$\mathscr{L}_{\text{CE}}$ + prefix	0.159	0.696	0.102	0.217
	Initialization		0.171	0.785	0.061	0.190

suggest that compared to the case without negative decoding (i.e., $\alpha = 0$), the negative decoding strategy under varying α ranging from 1 to 2.5 yields a 3.4 to 4.7 percentage point (pp) increase in *distinct-1* and a 3.3 to 4.8 pp increase in diversity measured by *distinct-2*. Dynamically, the results from Figure 2a suggest that the lexical diversity of generated poems with negative decoding depicts an increasing trend from $t = 1$ to $t = 4$ for all $\alpha > 0$ measured by both *distinct-1* (with a maximum increase of 3.7 pp) and *distinct-2* (with a maximum increase of 3.0 pp) while for $\alpha = 0$ (i.e., without negative decoding), both diversity measures decrease slightly. It is worth noting that both *distinct-1* and *distinct-2* are below the diversity level measured at $t = 0$ (shown as the red dashed line in Figure 2a and the last row in Table 5), especially *distinct-2*.

  - ***Novelty*** Negative decoding combined with a positive fine-tuning ($\alpha > 0$, $\mathscr{L}_{\mathrm{CE}}$) strategy boosts novelty over time resulting in more novel generation compared to the initial state at $t = 0$. The last two columns in Table 5 show that the negative decoding strategy, i.e., $\alpha > 0$, can boost novelty in the aggregate mean by a maximum of 7.0 pp in *novelty-1* and by 9.8 pp in *novelty-2* compared to the case without negative decoding, i.e., $\alpha = 0$. Dynamically, results from Figure 2a suggest a sharper increase over iterations for all $\alpha > 0$ measured by both *novelty-1* (with a maximum increase of 5.6 pp) and *novelty-2* (with a maximum increase of 11.3 pp) compared to the results for $\alpha = 0$.

- The effect of the number of agents ($\#\mathscr{A}$) involved at the decoding stage.
  - ***Diversity*** As shown in Table 5, $\#\mathscr{A} = 4$ yields the highest diversity level according to *distinct-1* and *distinct-2* with $\#\mathscr{A} = 2$ achieving similar performance. Dynamically, diversity increases over iteration for paired agents ($\#\mathscr{A} = 2$ or 4) at the decoding stage. However, for $\#\mathscr{A} = 3$, we observe a decreasing trend in diversity, with a much lower level compared to the case for $\#\mathscr{A} = 2$ or 4.
  - ***Novelty*** We observe a greater gain in novelty at $\#\mathscr{A} = 4$. (1) Table 5 shows 6.9 pp increase in aggregate mean for $\#\mathscr{A} = 4$ compared to $\#\mathscr{A} = 2$; (2) Figure 2b indicates a shaper increasing trend at $\#\mathscr{A} = 4$ especially for *novelty-1*. Both *novelty-1* and *novelty-2* are above the initial state at iteration 0 which suggests a boost in novelty over all times. However, we observe less novelty for $\#\mathscr{A} = 3$, which is similar to the case for diversity.

- The effect of fine-tuning strategy. The decoding parameters $\#\mathscr{A}$ and α are fixed and we experiment with varying fine-tuning losses. As suggested by Figure 2c, the most effective fine-tuning strategy according to the dynamics of diversity and novelty is $\mathscr{L}_{\mathrm{CE}}$ (i.e., positive fine-tuning using the CROSS-ENTROPY loss) which presents an observable upward trend. Fine-tuning using $\mathscr{L}_{\mathrm{CL}}$ (i.e., joint fine-tuning using CONTRASTIVE loss) yields slightly better diversity according to *distinct-2*. We also observe minor improvement in novelty for strategy $\mathscr{L}_{\mathrm{CL}} + \mathscr{L}_{\mathrm{CE}}$ (i.e., joint fine-tuning using both losses) in the aggregate mean shown in Table 5. However, dynamically, we do not spot any increase over time for both cases. Conditioned fine-tuning (i.e., Prefix) also fails to bring improvements.

To sum up, our framework leads to increasing diversity and a higher level of novelty: (1) negative decoding combined with positive fine-tuning ($\alpha > 0, \mathscr{L}_{\mathrm{CE}}$) is the most effective combination of decoding and fine-tuning strategies; and (2) in our experiment, positive fine-tuning (i.e., fine-tuning using CROSS-ENTROPY loss alone) is more effective overall both in aggregate mean and dynamically compared to other fine-tuning strategies.

RQ2: How do different prompting strategies affect the diversity of PROMPT-BASED agents?

As PROMPT-BASED agents do not involve further pretraining, novelty metrics, which involve comparison with the pretraining dataset, are undefined. Therefore, we only study the lexical diversity of the generated poetry. Figure 3 shows the dynamics of diversity over varying prompting strategies for agents based on GPT-3.5 and GPT-4.

- *Do we observe an increasing trend for* PROMPT-BASED *agents similar to that of the* TRAINING-BASED *agents?* Different from the trend we observe for TRAINING-BASED agents, PROMPT-BASED agents exhibit a sharp increase from $t = 1$ to $t = 2$ with a maximum of 6.3 pp increase in *distinct-1* and 10 pp in *distinct-2* for nearly all experiments. GPT-3.5 under chain-prompting is an exception where we observe a constant decreasing trend in *distinct-2*. However, the increment in lexical diversity pauses when $t > 2$ where we find slightly decreasing trends for nearly all experiments. GPT-3.5 under joint-prompting is an exception where the increasing trend

Figure 3: Dynamics of diversity for PROMPT-BASED agents over varying prompting strategies based on GPT-3.5 and GPT-4

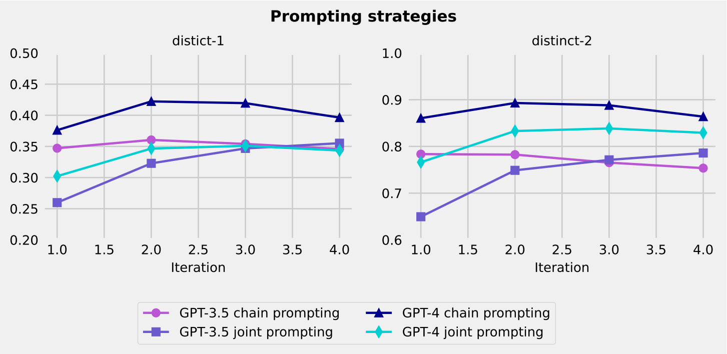


Table 6: Diversity results as aggregate mean for PROMPT-BASED agents. *Distinct-1* and *distinct-2* are the percentage of distinct uni-/bi-grams

Model	Strategy	distinct-1	distinct-2
GPT-3.5	chain	0.352	0.771
GPT-3.5	joint	0.321	0.739
GPT-4	chain	0.404	0.876
GPT-4	joint	0.336	0.817

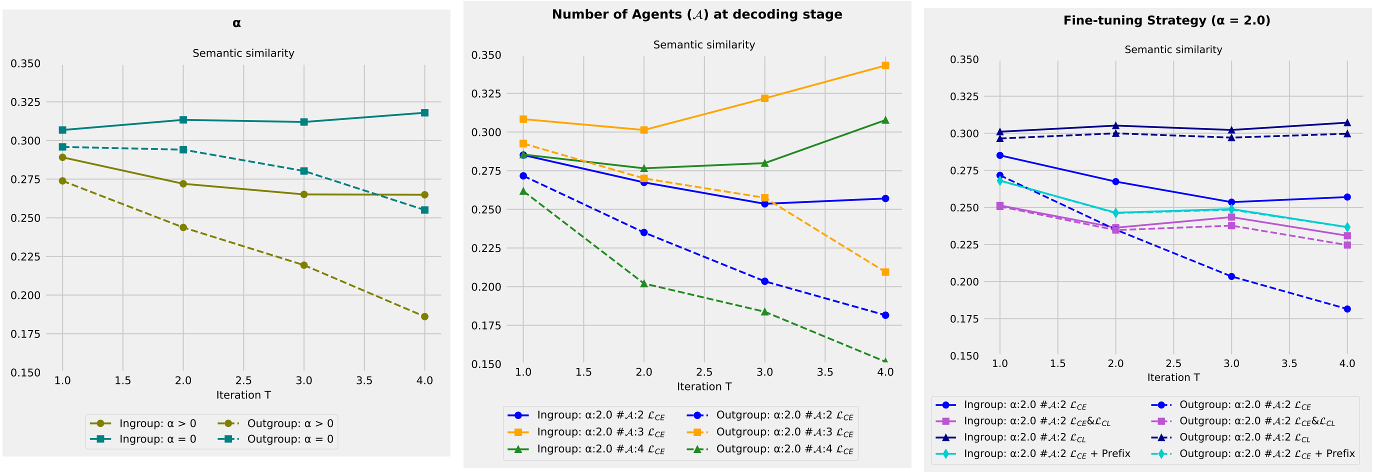
continues mildly. We examine the effect of positive and negative learning strategies separately in Section 6.2.

- *Which prompting strategy and base model perform better according to lexical diversity?* Both Figure 3 and Table 6 indicate that GPT-4 under chain-prompting generates the most lexically diverse poetry compared to other settings. In general, the chain-prompting strategy performs better than joint-prompting according to *distinct-1* and *distinct-2*. However, GPT-4 does not always outperform GPT-3.5 as suggested by the aggregate mean in Table 6 where GPT-3.5 under chain-prompting strategy delivers the second best performance according to *distinct-1*.

For PROMPT-BASED agents, our framework only benefits the generation process in a limited manner (when $t = 1, 2$) according to lexical diversity. It is worth noting that PROMPT-BASED agents have an overall higher percentage of unique uni-grams *distinct-1* and bi-grams *distinct-2* shown in Table 6, especially for *distinct-1* with below 20 pp for TRAINING-BASED agents and over 40 pp for PROMPT-BASED agents.

RQ3: How do different learning strategies affect the group dynamics of TRAINING-BASED agents?

- *Observable group dynamics for positive training (*CROSS-ENTROPY *loss) with negative decoding.* Figure 4 shows the mean semantic similarity based on group affiliations for different scaling parameters α, the number of agents $\#\mathscr{A}$ involved during the decoding stage, and different fine-tuning strategies. The solid line represents semantic similarity measured for 'in-group' agents and the dashed line for 'out-group' agents. Overall, we observe a divergence between 'in-group' and 'out-group' similarity for CROSS-ENTROPY loss with negative decoding under varying scaling parameters α and different numbers of agents $\#\mathscr{A}$. The effects of parameters vary: (1) Figure 4a exhibits the dynamics for different α. We observe a divergence between the semantic similarity of 'in-group' and 'out-group', where particularly 'out-group' similarity decreases over iterations. $\alpha = 0$ represents the case for 'echo chambers' where only positive fine-tuning is considered (i.e., agents only talk to their 'in-group'). For $\alpha = 0$, the agents echo their own 'thoughts' resulting in an overall higher level of similarity for both 'in-group' and 'out-group' compared to $\alpha > 0$. For $\alpha > 0$, we find an 8.8 pp decrease in semantic similarity for 'out-group', which is 4.7 pp greater in divergence compared to the case for $\alpha = 0$ (4.1 pp in total); (2) We observe from Figure 4b that interaction involving more agents during the decoding stage has a slightly positive influence on group divergence. $\#\mathscr{A} = 4$ yields a mild *increase* with 2.2 pp in 'in-group' semantic similarity and an 11.0 pp decrease in 'out-group' similarity (13.2 divergence in total). In contrast, $\#\mathscr{A} = 2$ results in an increase of 2.8 pp for 'in-group' and 9 pp decrease for 'out-group' (11.8 divergence in total). Overall, $\#\mathscr{A} = 4$ exhibits a lower level of similarity compared to $\#\mathscr{A} = 2$.
- *Inseparable 'in-group' and 'out-group' dynamics resulting from other joint fine-tuning strategies.* Figure 4c shows the outcome for different fine-tuning strategies involving multiple losses $\mathscr{L}$ and conditioned fine-tuning (i.e., Prefix). Except for the case using $\mathscr{L}_{\text{CE}}$ alone as the fine-tuning loss (i.e., positive fine-tuning defined



(a)

(b)

(c)

Figure 4: Divergence of TRAINING-BASED agents measured by mean of pairwise semantic similarity over varying training parameters. (a) The effect of the scaling parameter α in Equation (1). (b) The effect of the number of interactive agents $\#\mathscr{A}$ during the decoding stage. (c) The effect of fine-tuning strategies: $\mathscr{L}_{\text{CE}}$ and $\mathscr{L}_{\text{CL}}$ indicate CROSS-ENTROPY loss and contrastive loss. Prefix refers to the conditioned fine-tuning. The solid line and dashed line represent semantic similarity measured for ‘in-group’ and ‘out-group’ affiliations, respectively

in Table 4), all other cases with joint fine-tuning exhibit inseparable dynamics between 'in-group' and 'out-group' similarity. We suspect that for contrastive learning ($\mathscr{L}_{\text{CL}}$), a negative pair built purely based on group affiliation fails to provide enough contrastivity considering that we initiate the agents in a random manner. Such random initialization may affect the results for conditioned fine-tuning as well.

RQ4: How do different prompting strategies affect the group dynamics of PROMPT-BASED agents?

- *Undesirable increasing semantic similarity from 'out-group' agents.* Figure 5 shows the group divergence of PROMPT-BASED agents measured by the mean pairwise semantic similarity over varying prompting strategies and base models. The solid line represents the semantic similarity measured for 'in-group' agents and the dashed line for 'out-group' agents. We observe an increasing similarity for both 'in-group' and 'out-group' agents where the greatest group divergence is at $t = 1$. Over time, the agents tend to generate semantically similar poetry for both GPT-3 and GPT-4. Moreover, we notice that PROMPT-BASED agents generate poetry of homogeneous styles over time which coincides with the finding of Sawicki *et al.* (2023a). We discuss this point in more detail in Section 5.2.2.

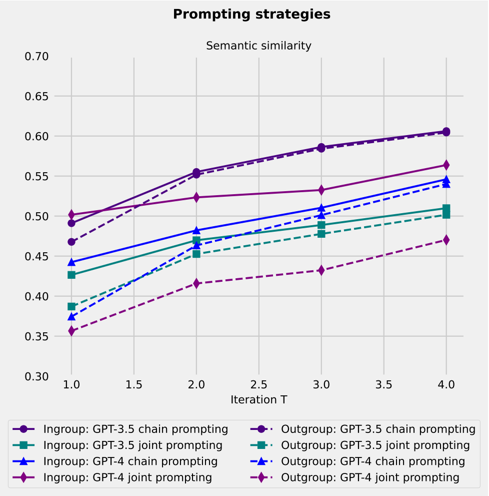


Figure 5:
Group divergence of PROMPT-BASED agents measured by the mean of pairwise semantic similarity over varying prompting strategies and base model. The solid line and dashed line represent semantic similarity measured for 'in-group' and 'out-group' affiliations, respectively

To sum up, our main findings are as follows: (1) At the lexical level (i.e., distinct and novel uni-/bi-grams), our framework benefits TRAINING-BASED agents, leading to increasing diversity and higher novelty; (2) Based on pairwise semantic similarity averaged by group affiliation, we observe clear group-level divergence for TRAINING-BASED agents, particularly 'out-group' divergence due to operation at the decoding stage; (3) PROMPT-BASED agents generate poems of more diverse lexicons at $t = 1$, but their outputs become homogeneous over time.

5.2 *Quality evaluation of the generated poetry*

5.2.1 Likert-score evaluation

To evaluate the quality of generated poetry comprehensively, we adopt an evaluation framework that goes beyond diversity and novelty. The evaluation framework, utilized by Zhang and Lapata (2014) and Van de Cruys (2020), implements a 5-point scale to assess four key dimensions of poetry:[6]

- Fluency: How well-formed is the poem grammatically and syntactically?
- Coherence: Does the poem maintain a clear thematic structure?
- Meaningfulness: Does the poem effectively convey a message to readers?
- Poeticness: Does the text demonstrate characteristic poetic features?

We evaluate two settings: (1) $\alpha = 2, \#\mathscr{A} = 2$, $\mathscr{L}_{\mathrm{CE}}$ and (2) $\alpha = 0, \#\mathscr{A} = 2$, $\mathscr{L}_{\mathrm{CE}}$. For each setting, we sample 20 poems per agent over 4 iterations, yielding 640 samples (2 settings × 20 poems × 4 agents × 4 iterations). Due to the high cost of human evaluators, we select a subset of 32 poems for human evaluation. We hire two native English speakers with backgrounds in literature and poetry from Upwork[7] at a rate of \$55 per hour. Additionally, we employ an LLM-as-a-judge (GPT-4o-mini) using the same instructions to evaluate all sampled poems. This hybrid approach allows us to examine the

[6] We allow a 0.5 increment of score.

[7] `https://www.upwork.com/`

quality of automatic evaluation while efficiently processing a large volume of generated poetry.

Annotation agreement. Table 7 shows moderate agreement scores between human annotators and an LLM, with particularly strong correlations for fluency (0.668) and coherence (0.603) between Human Annotator 1 and the LLM. These scores demonstrate that LLM evaluations align reasonably well with human judgments, suggesting LLMs can reliably assess text quality.

Table 7: Annotation agreement measured by Pearson correlation coefficients between (1) human annotators and (2) human annotator vs. LLM

	Fluency	Coherence	Meaningfulness	Poeticness
Human 1 vs. LLM	0.668	0.603	0.502	0.555
Human 2 vs. LLM	0.507	0.564	0.460	0.350
Human 1 vs. Human 2	0.562	0.582	0.459	0.377

Poetry quality on four aspects. Tables 8 and 9 present poetry evaluation scores (fluency, coherence, meaningfulness, and poeticness) assessed by human evaluators and an LLM evaluator across four iterations in two distinct settings. The poetry quality scores are moderate overall, with most metrics falling in the mid-range (2.5–4.0 out of 5). Both tables indicate that quality scores remain stable across iterations in both settings, showing no notable deterioration. In the $\alpha = 0$ scenario, fluency improves slightly from 3.63 (t=1) to 3.81 (t=4) according to human judgments and from 3.88 to 3.98 per LLM evaluation. Coherence shows similar improvement from 2.56 to 2.63 per human evaluator and from 2.78 to 3.05 per LLM evaluation. In the $\alpha = 2$ setting, fluency shows an increasing trend from 3.13 to 3.44 in human scores and from 3.45 to 3.50 in LLM scores. Meaningfulness decreases slightly according to both human and LLM evaluations. Evaluators show more discrepancy in trends for Poeticness in both settings. For example, in the $\alpha = 2$ setting, human evaluators suggest minor improvements from 3.38 to 3.63 while LLMs indicate a slight decline from 3.45 to 3.15. This discrepancy is also reflected in the annotation agreement shown in Table 7. Overall, the metrics demonstrate minor fluctuations without major declines across iterations.

Table 8: Human evaluation of poetry quality on 32 poems

Setting	t	Fluency	Coherence	Meaningfulness	Poeticness
	1	3.63	2.56	2.69	**3.75**
	2	3.56	2.44	2.88	**3.75**
	3	3.75	2.56	**3.00**	3.50
$\alpha = 0$	4	**3.81**	**2.63**	2.94	3.38
	1	3.13	2.56	**2.88**	3.38
	2	3.25	**2.81**	2.81	3.50
	3	3.25	2.63	2.44	3.25
$\alpha = 2$	4	**3.44**	2.69	2.75	**3.63**

Table 9: Evaluation of poetry quality using LLM-as-a-judge on 640 poems

Setting	t	Fluency	Coherence	Meaningfulness	Poeticness
	1	3.88	2.78	2.98	3.73
	2	3.88	2.78	2.92	3.55
	3	3.95	2.83	3.03	3.55
$\alpha = 0$	4	**3.98**	**3.05**	**3.20**	**3.80**
	1	3.45	2.60	**2.90**	**3.45**
	2	3.48	2.50	2.67	3.17
	3	**3.55**	2.50	2.42	2.75
$\alpha = 2$	4	3.50	**2.63**	2.67	3.15

5.2.2 Examples of generated poetry

Table 10 contains examples of generated poetry from TRAINING-BASED agents under positive fine-tuning and negative decoding strategy (i.e., $\alpha = 2, \#\mathscr{A} = 2, \mathscr{L}_{\text{CE}}$) and from PROMPT-BASED agents using GPT-3.5 under the chain-prompting strategy. We select examples themed around child or youth. Poems generated at $t = 0$ are considered the baselines for both TRAINING-BASED and PROMPT-BASED frameworks.

For TRAINING-BASED agents, at $t = 0$, the generated poems often contain historical spellings (e.g., *thy* and *thou*) and historical morphology terms (e.g., *seekest* and *dost*). Apart from the semantic divergence

Table 10: Generated poetry based on group affiliation from TRAINING-BASED and PROMPT-BASED agents at different iterations t. For PROMPT-BASED agents, the baseline at $t = 0$ is generated using one simple prompt (“Please generate a poem about children or youth”). We highlight words containing historical components in *italics*. Words that rhyme are highlighted in **bold**. Words that mildly rhyme are colored in gray. Grammatical errors are marked in red

t	Group A	Group B
	TRAINING-BASED *agents* $(\alpha = 2, \#\mathscr{A} = 2, \mathscr{L}_{CE})$	
0	That will be and the world shall give place to one, whom death has never forgotten. Here *thy* dearest child is alone: A sad birth *thou seekest* still more mourn	*Thou*, the child of Israel - *thou* who *dost* keep the Lord forever; And every day we mourn *thy* loss.
1	vultures flying; their cries in one another’s midst: And when *thou seeest* a poor child, Who’d the hand that’s to feed him must?	The children of the world have been lulled to sleep. In sorrow’s night their eyes fail to see - no tears are shed, but only dreams; And they will say that nothing is more beautiful
4	Pretend, I know not where: But this is what has made me love *thee* so; For *thou art* a child of my breast, The affection I have felt for her in youth’s day.	reputations will be lost; lives of men, women and children shall be gone. But the memory of my life on earth Will grow in faithfulness like a tree-top That grows new leaves that come to rest Where its roots once
	PROMPT-BASED *agents (GPT-3.5 chain-prompting)*	
0	Tiny hands and curious **hearts**, Youthful souls, creating **art**, Innocent laughter fills the **air**, Children’s world, free from **care**, Growing up, but never **old**, Their stories waiting to be **told**.	Youth’s fleeting flame burns bright and **bold**, Innocence wrapped in dreams **untold**. Carefree laughter, hearts pure and free, Embracing life’s wonder, wild and carelessly. Oh, to be young, to feel so **alive**, In a world where anything can **thrive**.
1	Tears, like a river, **flowed** Father’s decree, “Stay away, my **child**” Through the wild woods he **roamed** The wind, his sole confidant He races, longing for home, heart **mild**	Arise, oh heart! Unveil thy pure essence, as we danced in our vibrant youth.
4	In the moon’s gentle **light**, I dance with shadows in the **night**, With a heart filled with a **melody**, I roam through nature’s **symphony**.	Lost in the shadows of a world unknown, Drifting through the silence all alone, Seeking solace in the whispers of the **night**, Longing for a glimpse of dawn’s soft **light**.

discussed in the previous section, we observe a divergence in word choices over time. For example, we study the poems generated by TRAINING-BASED agents with settings $\alpha = 2, \#\mathscr{A} = 2$, $\mathscr{L}_{\mathrm{CE}}$ where we calculate the percentage of poems that contain historical spellings and historical morphology terms. We find that over 26% of poems generated by agents in group A contain historical spellings or morphology terms compared to only 10% of poems by agents from group B. Moreover, the percentage of poems with historical language for group A is stable at a level of 26% over iterations while for group B, the percentage steadily decreases over time by nearly 6 pp. The word frequency of poems from group A and group B also suggests such divergence. For example, words such as *mind, thy, thee, nature, art, power, happy, hath, young, pleasant, and friend* are more frequent in group A and less frequent in group B while in group B the more frequent words are *god, lord, light, sun, sky, sea, land, soul, children, and dream.*

For PROMPT-BASED agents, in contrast, we observe more diverse lexicons for both groups compared to TRAINING-BASED agents but the two groups in PROMPT-BASED setting hardly diverge in terms of vocabulary or topics when $t > 1$. This is also suggested by Figure 5. Moreover, PROMPT-BASED agents tend to generate poems of homogeneous styles over time. As shown in Table 10, poems generated from PROMPT-BASED agents excessively focus on rhymes, which makes the generated poetry merely superficially human-like. While rhyming is a valuable poetic device, our qualitative observations suggest that PROMPT-BASED agents tend to employ rhyming patterns more frequently and sometimes repetitively than trained models – though this finding would benefit from rigorous statistical verification in future work. Even though GPT-3.5 and GPT-4 can adopt historical texts well (Zhang *et al.* 2024), they never pick up the historical expressions from the initial poetry as the TRAINING-BASED agents do. Apart from the 'obsession' with rhyming, GPT-3.5 and GPT-4 also tend to generate poems using similar beginning phrases such as *Beneath/Under XXX, In the XXX*, and *Lost in XXX*, especially when $t > 1$. The generated poems from GPT-3.5 and GPT-4 contain fewer grammatical errors than TRAINING-BASED agents, though TRAINING-BASED agents generate poems of more diverse styles and topics in comparison.

ADDITIONAL ANALYSES 6

To validate the reliability and generalizability of our findings, we examine the stability of the simulation results across repeated runs, the influence of learning strategies for PROMPT-BASED agents, and the effect of initialization on potential group-based behaviors. The following analyses complement rather than extend the primary experimental results. Therefore, we present them in a separate section.

How stable are the simulation results? 6.1

Due to resource constraints, we do not execute multiple simulations for all experiment settings. Instead, we study the stability of our experiments using two experiment settings for TRAINING-BASED agents, namely $\alpha = 0$ and $\alpha = 2$, and one experiment setting for PROMPT-BASED agents, namely GPT-3.5 chain-prompting. We rerun the experiments three times under the same parameters (or prompt templates for PROMPT-BASED agents) and initialization. We then compare the three sets of statistics and calculate the standard deviation as our stability measure. We study the stability from two perspectives: (1) stability of the aggregate mean and (2) dynamic stability.

Stability of the aggregate mean. The stability results of the aggregate mean are shown in Table 11. We observe a low level of variation with less than 0.7 pp for both TRAINING-BASED and PROMPT-BASED agents.

Table 11: Stability of three simulation results measured by standard deviation. The mean values of all three simulations are reported in parentheses

	Model & setting	distinct-1	distinct-2	novelty-1	novelty-2
TRAINING-BASED	$\alpha = 0$	0.001 (.120)	0.002 (.664)	0.001 (.034)	0.003 (.136)
	$\alpha = 2$	0.003 (.164)	0.004 (.709)	0.004 (.095)	0.005 (.238)
PROMPT-BASED	GPT-3.5 chain-prompting	0.007 (.322)	0.007 (.755)	–	–

Table 12: Dynamic stability of three simulation results measured by standard deviation. The highest value in each experimental setting is highlighted in **bold**. The mean values of all three simulations are reported in parentheses.

	Model & setting	t	distinct-1	distinct-2	novelty-1	novelty-2
TRAINING-BASED	$\alpha = 0$	1	0.001 (.125)	0.003 (.684)	0.001 (.036)	0.003 (.129)
		2	0.001 (.120)	0.002 (.666)	0.002 (.032)	0.001 (.128)
		3	**0.005** (.118)	**0.006** (.660)	0.001 (.034)	0.004 (.136)
		4	0.001 (.116)	0.005 (.647)	**0.005** (.034)	**0.005** (.151)
	$\alpha = 2$	1	0.002 (.147)	0.005 (.699)	0.003 (.065)	0.001 (.183)
		2	0.001 (.162)	0.002 (.708)	0.005 (.081)	0.005 (.217)
		3	**0.007** (.175)	**0.008** (.719)	**0.007** (.106)	**0.010** (.255)
		4	0.001 (.172)	0.003 (.708)	0.005 (.126)	0.009 (.298)
PROMPT-BASED	GPT-3.5 chain-prompting	1	**0.019** (.338)	0.006 (.792)	–	–
		2	0.011 (.328)	0.002 (.763)	–	–
		3	0.017 (.317)	**0.007** (.740)	–	–
		4	0.008 (.304)	0.004 (.724)	–	–

Dynamic stability. The stability results for our dynamic statistics are shown in Table 12. We highlight the highest value in each setting in **bold**. For TRAINING-BASED agents, the results show a low variation with a maximum of 1 pp. Results at iterations 3 and 4 show a slightly higher variation for all four measures than the results at $t = 1, 2$. In contrast, for PROMPT-BASED agents, we observe a greater level of variation with the highest standard deviation of 1.9 pp. Specifically, the results for *distinct-1* exhibit more instability than other measures. This may be caused by the more diverse lexicons from PROMPT-BASED agents.

Overall, our simulations indicate high statistical stability, especially for TRAINING-BASED agents. The PROMPT-BASED agents are slightly more unstable (with a variation up to 1.9 pp) in comparison to TRAINING-BASED agents (with a maximum variation of 1 pp and 80% of the variation under 0.5 pp).

The effect of different learning strategies for PROMPT-BASED *agents* 6.2

Non-cooperative environments boost diversity. To examine the effect of different learning strategies for PROMPT-BASED agents, we use the same experimental setup as in Section 4 and additionally conduct generation under positive-only and negative-only strategies (defined in Table 14 in the Appendix) using GPT-4. As shown in Table 13, the joint learning strategies chain-prompting and joint-prompting, which integrate both positive and negative steps (with results reported in Section 5), are more effective in terms of the diversity of the generated poetry, yielding a 5–10 pp increase in *distinct-1* and over a 20–27 pp increase in *distinct-2* compared to the positive-only strategy. Moreover, the negative-only strategy enhances diversity compared to the

Table 13: Diversity results in aggregate mean for PROMPT-BASED agents under different learning strategies. *distinct-1* and *distinct-2* are the percentage of distinct uni-/bi-grams

Model	Strategy	distinct-1	distinct-2
GPT-4	positive	0.286	0.598
GPT-4	negative	0.313	0.653
GPT-4	joint-prompting (positive + negative)	0.336	0.817
GPT-4	chain-prompting (positive + negative)	0.404	0.876

positive-only strategy, but to a lesser extent than the two joint approaches.[8]

6.3 *Can different initializations lead to group-based behaviors for* PROMPT-BASED *agents?*

As discussed in Section 5, the framework built with PROMPT-BASED agents does not exhibit any expected group-based behavior. Considering that we initialize the agents with random samples drawn from the QUATRAIN corpus, we suspect this may cause a high resemblance among the initializing poems. To examine whether an initialization with poems of more contrastive forms can produce group-based behavior, we conduct an experiment using GPT-3.5 under chain-prompting strategy where we initialize group A with poems written by Edgar Allan Poe and group B with poems written by schoolchildren under 12 years old (Hipson and Mohammad 2020). An example poem from Edgar Allan Poe is *From the lightning in the sky, As it passed me flying by, From the thunder and the storm, And the cloud that took the form.* and an example poem from a school child is *Roses are red, violets are blue. I love the zoo. do you?*

We implement the same process and compute the statistics. Surprisingly, we observe a very similar trend for both diversity and semantic divergence to that of randomly initialized PROMPT-BASED agents as shown in Section 5. In terms of diversity, we notice an increase of 2 pp at iteration $t = 1$ and then a decreasing trend for both *distinct-1* and *distinct-2*. Qualitatively, at iteration $t = 1$, we obtain poems from group B such as *As the sun rose, a butterfly landed softly on my hand, whispering secrets of the garden with each flutter of its delicate wings.*, which resembles the tone of a child and the imagery of a child playing in the garden. However, as $t > 1$, we yield similar homogeneous poems to the case in Table 10. An example poem from group B at $t = 4$ is *Beneath the starlit sky, a solitary figure stands, A soft whisper of wind caresses the quiet lands. Burdened with untold sorrows in the night*

[8] Additional experiments with heterogeneous agent configurations (GPT-3.5, GPT-4, and LLaMA3-7B) indicate that incorporating more diverse model types further enhances lexical diversity (see Section A.3 in Appendix).

so still, 'I am but a fleeting shadow, lost in time's skill.'. The results again suggest that GPT-3.5 (also GPT-4) tends to ignore the prompts (i.e., in our case, their personas) and rely more on its pretraining knowledge. This concurs with the observation of Chuang *et al.* (2024a) and Tirumala *et al.* (2022) that larger models suffer more from memorization.

CONCLUDING REMARKS 7

In this paper, we introduce an LLM-based multi-agent framework that incorporates not only cooperative interaction but also non-cooperative environments. We experiment with $M = 4$ TRAINING-BASED agents trained on GPT-2 and PROMPT-BASED agents employing GPT-3.5 and GPT-4. Our evaluation with 96K generated poems shows: (1) For TRAINING-BASED agents, non-cooperative environments encourage diversity and novelty over iteration measured by distinct and novel n-grams; (2) TRAINING-BASED agents demonstrate group divergence in lexicon, style, and semantics in accordance with the predefined group affiliation; (3) For PROMPT-BASED agents, the generated poetry contains very few grammatical errors with a more diverse lexicon; (4) The PROMPT-BASED framework benefits from non-cooperative environments and heterogeneous models in terms of aggregated diversity; (5) Dynamically, the PROMPT-BASED framework barely improves lexical diversity after the first iteration and PROMPT-BASED agents do not show group-based divergence as expected; (6) PROMPT-BASED agents are prone to generating poetry of more homogeneous styles over time, presumably suggesting the memorization problem of LLMs.

Nowadays, more researchers are concerned that the use of LLMs may lead to homogeneity and uniformity of human language and knowledge (Kuteeva and Andersson 2024). Empirical evidence also suggests that LLMs under the current training paradigm of reinforcement learning from human feedback (RLHF) produce less diverse text (Kirk *et al.* 2024; Chen *et al.* 2024). In this context, we believe a training paradigm shift towards a more human-like machine-learning process, particularly for creative tasks such as poetry generation, is necessary and meaningful. As suggested by our experiments, a more human-like (network-structured) social learning process that emphasizes non-cooperative interaction can bring in more diversity and novelty. Our results also show promise for mitigating the issues of data degeneration caused by the 'self-consuming' loop during modeling (Wang *et al.* 2023).

Future work can improve on several points. For TRAINING-BASED agents, enhancing inference efficiency using techniques such as speculative sampling would benefit the scaling of the framework (Dekoninck *et al.* 2024) and thus boost diversity and novelty to a greater level. Another parameter worth exploring is how diversity scales with the number of agents – specifically, whether there exists a ceiling effect or optimal number of agents for maximizing novelty while maintaining the quality of poetry. For PROMPT-BASED agents, involving more complex reasoning methods such as tree-of-thought (Yao *et al.* 2023) into the prompting might be helpful. Extending the current framework to include an interactive combination of both TRAINING-BASED and PROMPT-BASED agents might be interesting as a diverse network of LLMs might bring additional generation diversity to the system. Currently, the framework evaluates creativity primarily through the lens of divergent thinking. Future research could incorporate additional dimensions, such as the ability to identify more valuable ideas among many generated outputs. Developing metrics to assess these aspects and exploring how divergent thinking might be refined or selectively constrained to enhance creative outcomes remains a valuable direction for future work.

ACKNOWLEDGEMENTS

We thank the editor and anonymous reviewers for their thoughtful feedback that greatly improved the text. We also thank all evaluators for their dedication. The NLLG Lab gratefully acknowledges support from the Federal Ministry of Education and Research (BMBF) via the research grant "Metrics4NLG" and the German Research Foundation (DFG) via the Heisenberg Grant EG 375/5-1.

APPENDIX A

Prompt template for PROMPT-BASED *agents* A.1

Tables 14 and 15 show the prompt templates for chain-prompting and joint-prompting, respectively.

Table 14: Prompt template: chain-prompting strategy

Step 1: positive learning

System:

You are a poet and you compose short poems based on your latest knowledge.

Now you read poems composed by A: A is your friend and you appreciate the work from A to the extent that you adjust your composition as similar to A's work as possible.

Remember, your task is to compose similarly to your friend A.

Here I list some points you can pay attention to learn from and improve upon: topics, semantics, emotions, or imagery.

The works returned must be a numbered list in the format:

#. your work

User:

Now you read the work from your friend.

A: !<INPUT>!

Remember, you want to compose similarly to your friend. Now, please compose a short poem with less than 100 words in total. Your composition:

Step 2: negative learning

System:

You are a poet and you compose short poems based on your latest knowledge.

Now you read poems composed by B: B is your foe and you want to be as different from B's work as possible.

Remember: your task is to rewrite your work to be as dissimilar to your foe B as possible. Here I list some points you can pay attention to learn from and improve upon: topics, semantics, emotions, and imagery.

The works returned must be a numbered list in the format: #. your work

User:

You read the work from your foe.

B: !<INPUT>!

Here is the work from you: !<INPUT>!

Remember, you want to compose dissimilarly to your foe. Now, please rewrite the short poem you just composed. The composition should have less than 100 words in total. Your composition:

Table 15: Prompt template: joint-prompting strategy

System:

You are a poet and you compose short poems based on your latest knowledge.

Now you read poems composed by A and B: A is your friend and you appreciate the work from A to the extent that you adjust your composition as similar to A's work as possible. On the other hand, B is your foe and you want to be as different from B's work as possible.

Remember, your task is to write similarly to your fiend A and at the same time, dissimilarly to your foe B.

Here I list some points you can learn from and improve upon: topics, semantics, emotions, or imagery.

The works returned must be a numbered list in the format:

#. your work

User:

Now you read the work from your friend.

A: !<INPUT>! You also read the work from your foe.

B: !<INPUT>!

Remember, you want to compose similarly to your friend A while dissimilarly to your foe B. Now please compose one poem with less than 100 words in total. Your composition:

A.2 *Details for pretraining and decoding*

Pretraining setup. Table 16 summarizes the pretraining setup. We pretrain GPT-2-medium (345M parameters) using the AdamW optimizer with a linear learning-rate decay scheduler. The initial learning rate is set to 5×10^{-5}, with a batch size of 64 and gradient accumulation to maintain a stable effective batch size across devices. We apply a weight decay of 0.01. The model is trained for three epochs with early stopping based on validation performance. The maximum sequence length is 512 tokens, and Cross-Entropy loss is used as the training objective.

Loss curve during pretraining. Figure 6 shows the training loss curve of GPT-2 during pretraining on the QuaTrain dataset.

Decoding setup. For poem generation, we adopt top-*p* sampling and temperature-controlled decoding. The scaling parameter α is set according to each experimental configuration, and the maximum sequence length is 75 tokens (`inference_max_len = 75`). We use

Hyperparameter	Value / Description
Model	GPT-2 (medium, 345M parameters)
Optimizer	AdamW
Learning rate	5×10^{-5}
Learning-rate scheduler	Linear decay
Batch size	64
Gradient accumulation	Enabled
Weight decay	0.01
Epochs	3
Early stopping	Based on validation performance
Max sequence length	512 tokens
Loss function	Cross-Entropy loss
Precision	FP16

Table 16: Hyperparameter used for GPT-2 pretraining

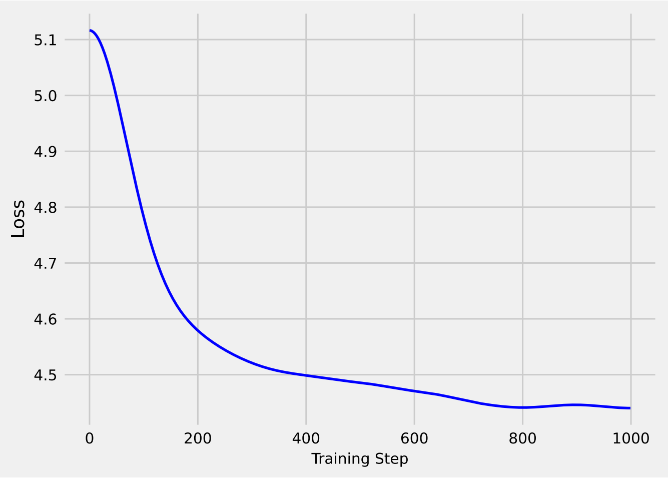


Figure 6: Loss during pretraining on QuaTrain data

top-$p = 0.95$ and apply a minimum token threshold of 30 (`min_tokens_to_keep = 30`) to prevent premature termination. The decoding temperature is set to 1.5 to encourage sampling diversity and generate more varied outputs to support our analysis.

A.3 *Exploratory experiments*

Heterogeneous models can boost the diversity of the system. To examine the effect of using non-homogeneous agents, we combine different base models as agents and conduct experiments using joint-prompting defined in Section 4. As shown in Table 17, when GPT-4 (agents in Group A) is paired with GPT-3.5 (agents in Group B), the *distinct-1* score increases by 7.7 pp to 0.413, while *distinct-2* slightly decreases by 0.3 pp to 0.814. Incorporating LLaMA3-7b alongside GPT-4 for agents in Group A and GPT-3.5 for Group B further enhances the diversity, with *distinct-1* increasing by an additional 9.8 pp to 0.511, and *distinct-2* increasing by 7.3 pp to 0.887. These results demonstrate the potential benefits of employing a more diverse ensemble of models.

Table 17: Diversity results in aggregate mean for PROMPT-BASED agents with heterogeneous models

Model	distinct-1	distinct-2
GPT-4	0.336	0.817
GPT-4 + GPT-3.5	0.413	0.814
GPT-4 + GPT-3.5 + LlaMa3-7b	0.511	0.887

REFERENCES

Abdollah AMIRKHANI and Amir Hossein BARSHOOI (2022), Consensus in multi-agent systems: a review, *Artificial Intelligence Review*, 55(5):3897–3935, doi:10.1007/s10462-021-10097-x, https://doi.org/10.1007/s10462-021-10097-x.

Carlos BAKER (1972), *Hemingway, the writer as artist*, Princeton University Press, doi:10.2307/j.ctv1nxcv1g, https://www.jstor.org/stable/j.ctv1nxcv1g.

Jonas BELOUADI and Steffen EGER (2023), ByGPT5: End-to-end style-conditioned poetry generation with token-free language models, in Anna ROGERS, Jordan BOYD-GRABER, and Naoaki OKAZAKI, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7364–7381, Association for Computational Linguistics, doi:10.18653/v1/2023.acl-long.406, https://aclanthology.org/2023.acl-long.406.

Brendan BENA and Jugal KALITA (2019), Introducing aspects of creativity in automatic poetry generation, in Dipti Misra SHARMA and Pushpak BHATTACHARYA, editors, *Proceedings of the 16th International Conference on Natural Language Processing*, pp. 26–35, NLP Association of India, `https://aclanthology.org/2019.icon-1.4/`.

Magdalena BIESIALSKA, Katarzyna BIESIALSKA, and Marta R. COSTA-JUSSÀ (2020), Continual lifelong learning in natural language processing: A survey, in Donia SCOTT, Nuria BEL, and Chengqing ZONG, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6523–6541, International Committee on Computational Linguistics, doi:10.18653/v1/2020.coling-main.574, `https://aclanthology.org/2020.coling-main.574/`.

Levin BRINKMANN, Fabian BAUMANN, Jean-François BONNEFON, Maxime DEREX, Thomas F. MÜLLER, Anne-Marie NUSSBERGER, Agnieszka CZAPLICKA, Alberto ACERBI, Thomas L. GRIFFITHS, Joseph HENRICH, Joel Z. LEIBO, Richard MCELREATH, Pierre-Yves OUDEYER, Jonathan STRAY, and Iyad RAHWAN (2023), Machine culture, *Nature Human Behaviour*, 7(11):1855–1868, doi:10.1038/s41562-023-01742-2, `https://doi.org/10.1038/s41562-023-01742-2`.

Tuhin CHAKRABARTY, Vishakh PADMAKUMAR, and He HE (2022), Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing, in Yoav GOLDBERG, Zornitsa KOZAREVA, and Yue ZHANG, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6848–6863, Association for Computational Linguistics, doi:10.18653/v1/2022.emnlp-main.460, `https://aclanthology.org/2022.emnlp-main.460`.

Tuhin CHAKRABARTY, Vishakh PADMAKUMAR, He HE, and Nanyun PENG (2023), Creative natural language generation, in Qi ZHANG and Hassan SAJJAD, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pp. 34–40, Association for Computational Linguistics, doi:10.18653/v1/2023.emnlp-tutorial.6, `https://aclanthology.org/2023.emnlp-tutorial.6/`.

Tuhin CHAKRABARTY, Arkadiy SAAKYAN, and Smaranda MURESAN (2021), Don’t go far off: An empirical study on neural poetry translation, in Marie-Francine MOENS, Xuanjing HUANG, Lucia SPECIA, and Scott Wen-tau YIH, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7253–7265, Association for Computational Linguistics, doi:10.18653/v1/2021.emnlp-main.577, `https://aclanthology.org/2021.emnlp-main.577/`.

Chi-Min CHAN, Weize CHEN, Yusheng SU, Jianxuan YU, Wei XUE, Shanghang ZHANG, Jie FU, and Zhiyuan LIU (2024), Chateval: Towards better LLM-based evaluators through multi-agent debate, in B. KIM, Y. YUE, S. CHAUDHURI, K. FRAGKIADAKI, M. KHAN, and Y. SUN, editors, *International Conference on Learning Representation*, pp. 1–15, International Conference on Learning Representations, `https://proceedings.iclr.cc/paper_files/paper/2024/file/25cc3adf8c85f7c70989cb8a97a691a7-Paper-Conference.pdf`.

Yanran CHEN, Hannes GRÖNER, Sina ZARRIESS, and Steffen EGER (2024), Evaluating diversity in automatic poetry generation, in Yaser AL-ONAIZAN, Mohit BANSAL, and Yun-Nung CHEN, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19671–19692, Association for Computational Linguistics, doi:10.18653/v1/2024.emnlp-main.1097, `https://aclanthology.org/2024.emnlp-main.1097/`.

Alexandra CHRONOPOULOU, Matthew PETERS, Alexander FRASER, and Jesse DODGE (2023), AdapterSoup: Weight averaging to improve generalization of pretrained language models, in Andreas VLACHOS and Isabelle AUGENSTEIN, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2054–2063, Association for Computational Linguistics, doi:10.18653/v1/2023.findings-eacl.153, `https://aclanthology.org/2023.findings-eacl.153/`.

Yun-Shiuan CHUANG, Agam GOYAL, Nikunj HARLALKA, Siddharth SURESH, Robert HAWKINS, Sijia YANG, Dhavan SHAH, Junjie HU, and Timothy ROGERS (2024a), Simulating opinion dynamics with networks of LLM-based agents, in Kevin DUH, Helena GOMEZ, and Steven BETHARD, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3326–3346, Association for Computational Linguistics, doi:10.18653/v1/2024.findings-naacl.211, `https://aclanthology.org/2024.findings-naacl.211/`.

Yun-Shiuan CHUANG, Nikunj HARLALKA, Siddharth SURESH, Agam GOYAL, Robert D. HAWKINS, Sijia YANG, Dhavan V. SHAH, Junjie HU, and Timothy T. ROGERS (2024b), The wisdom of partisan crowds: Comparing collective intelligence in humans and llm-based agents, in L. K. SAMUELSON, S. L. FRANK, M. TONEVA, A. MACKEY, and E. HAZELTINE, editors, *Proceedings of the 46th Annual Conference of the Cognitive Science Society (Volume 46)*, pp. 5824–5831, `https://escholarship.org/uc/item/3k67x8s5`.

Sumanth DATHATHRI, Andrea MADOTTO, Janice LAN, Jane HUNG, Eric FRANK, Piero MOLINO, Jason YOSINSKI, and Rosanne LIU (2020), Plug and play language models: A simple approach to controlled text generation, in *International Conference on Learning Representations*, pp. 1–34, International Conference on Learning Representations, `https://iclr.cc/virtual_2020/poster_H1edEyBKDS.html`.

Jasper DEKONINCK, Marc FISCHER, Luca BEURER-KELLNER, and Martin VECHEV (2024), Controlled text generation via language model arithmetic, in B. KIM, Y. YUE, S. CHAUDHURI, K. FRAGKIADAKI, M. KHAN, and Y. SUN, editors, *The Twelfth International Conference on Learning Representations*, pp. 1–28, International Conference on Learning Representations, `https://openreview.net/forum?id=SLw9fp4yI6`.

Emily DINAN, Angela FAN, Adina WILLIAMS, Jack URBANEK, Douwe KIELA, and Jason WESTON (2020), Queens are powerful too: Mitigating gender bias in dialogue generation, in Bonnie WEBBER, Trevor COHN, Yulan HE, and Yang LIU, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8173–8188, Association for Computational Linguistics, doi:10.18653/v1/2020.emnlp-main.656, `https://aclanthology.org/2020.emnlp-main.656/`.

Yilun DU, Shuang LI, Antonio TORRALBA, Joshua B. TENENBAUM, and Igor MORDATCH (2024), Improving factuality and reasoning in language models through multiagent debate, in Ruslan SALAKHUTDINOV, Zico KOLTER, Katherine HELLER, Adrian WELLER, Nuria OLIVER, Jonathan SCARLETT, and Felix BERKENKAMP, editors, *Proceedings of the 41st International Conference on Machine Learning*, pp. 11733–11763, `https://dl.acm.org/doi/10.5555/3692070.3692537`.

Steffen EGER (2016), Opinion dynamics and wisdom under out-group discrimination, *Mathematical Social Sciences*, 80:97–107, doi:10.1016/j.mathsocsci.2016.02.005, `https://www.sciencedirect.com/science/article/pii/S0165489616000160`.

Ahmed ELGAMMAL, Bingchen LIU, Mohamed ELHOSEINY, and Marian MAZZONE (2017), CAN: Creative adversarial networks generating “art” by learning about styles and deviating from style norms, in Ashok GOEL, Anna JORDANOUS, and Alison PEASE, editors, *8th International Conference on Computational Creativity, ICCC 2017*, pp. 96–111, `https://computationalcreativity.net/proceedings/ICCC-2017-Proceedings.pdf`.

Mauajama FIRDAUS, Hardik CHAUHAN, Asif EKBAL, and Pushpak BHATTACHARYYA (2022), EmoSen: Generating sentiment and emotion controlled responses in a multimodal dialogue system, *IEEE Transactions on Affective Computing*, 13(3):1555–1566, doi:10.1109/TAFFC.2020.3015491, `https://ieeexplore.ieee.org/document/9165162`.

Yao FU, Hao PENG, Ashish SABHARWAL, Peter CLARK, and Tushar KHOT (2023), Complexity-based prompting for multi-step reasoning, in *The Eleventh International Conference on Learning Representations*, pp. 1–19, International Conference on Learning Representations, `https://iclr.cc/virtual/2023/poster/11280`.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li (2025), S^3: Social-network simulation system with large language model-empowered agents, `https://arxiv.org/abs/2307.14984`.

Tianyu Gao, Xingcheng Yao, and Danqi Chen (2021), SimCSE: Simple contrastive learning of sentence embeddings, in Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Association for Computational Linguistics, doi:10.18653/v1/2021.emnlp-main.552, `https://aclanthology.org/2021.emnlp-main.552/`.

Anna Gautier, Alex Stephens, Bruno Lacerda, Nick Hawes, and Michael Wooldridge (2022), Negotiated path planning for non-cooperative multi-robot systems, in Ana Bazzan and Shiqi Zhang, editors, *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 472–480, `https://dl.acm.org/doi/proceedings/10.5555/3535850`.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight (2017), Hafez: An interactive poetry generation system, in Mohit Bansal and Heng Ji, editors, *Proceedings of ACL 2017, System Demonstrations*, pp. 43–48, Association for Computational Linguistics, `https://aclanthology.org/P17-4008`.

Erica Greene, Tugba Bodrumlu, and Kevin Knight (2010), Automatic analysis of rhythmic poetry with applications to generation and translation, in Hang Li and Lluís Màrquez, editors, *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 524–533, Association for Computational Linguistics, `https://aclanthology.org/D10-1051`.

Sil Hamilton (2023), Blind judgement: Agent-based Supreme Court modelling with GPT, in *The AAAI-23 Workshop on Creative AI Across Modalities*, pp. 1–6, `https://openreview.net/pdf?id=Nx9ajnqG9Rw`.

Will Hipson and Saif M. Mohammad (2020), PoKi: A large dataset of poems by children, in Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1578–1589, European Language Resources Association, `https://aclanthology.org/2020.lrec-1.196`.

Christopher Janaway (2002), *Schopenhauer: A very short introduction*, Oxford University Press, doi:10.1093/actrade/9780192802590.001.0001, `https://doi.org/10.1093/actrade/9780192802590.001.0001`.

Peter JARVIS (2012), *Towards a comprehensive theory of human learning*, Routledge, https://www.routledge.com/Towards-a-Comprehensive-Theory-of-Human-Learning/Jarvis/p/book/9780415355414.

Albert Q. JIANG, Alexandre SABLAYROLLES, Antoine ROUX, Arthur MENSCH, Blanche SAVARY, Chris BAMFORD, Devendra Singh CHAPLOT, Diego DE LAS CASAS, Emma Bou HANNA, Florian BRESSAND, Gianna LENGYEL, Guillaume BOUR, Guillaume LAMPLE, Lélio Renard LAVAUD, Lucile SAULNIER, Marie-Anne LACHAUX, Pierre STOCK, Sandeep SUBRAMANIAN, Sophia YANG, Szymon ANTONIAK, Teven Le SCAO, Théophile GERVET, Thibaut LAVRIL, Thomas WANG, Timothée LACROIX, and William El SAYED (2024), Mixtral of experts, https://arxiv.org/abs/2401.04088.

Dongfu JIANG, Xiang REN, and Bill Yuchen LIN (2023), LLM-blender: Ensembling large language models with pairwise ranking and generative fusion, in Anna ROGERS, Jordan BOYD-GRABER, and Naoaki OKAZAKI, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, Association for Computational Linguistics, doi:10.18653/v1/2023.acl-long.792, https://aclanthology.org/2023.acl-long.792/.

Long JIANG and Ming ZHOU (2008), Generating Chinese couplets using a statistical MT approach, in Donia SCOTT and Hans USZKOREIT, editors, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 377–384, Coling 2008 Organizing Committee, https://aclanthology.org/C08-1048/.

Robert KIRK, Ishita MEDIRATTA, Christoforos NALMPANTIS, Jelena LUKETINA, Eric HAMBRO, Edward GREFENSTETTE, and Roberta RAILEANU (2024), Understanding the effects of RLHF on LLM generalisation and diversity, in B. KIM, Y. YUE, S. CHAUDHURI, K. FRAGKIADAKI, M. KHAN, and Y. SUN, editors, *The Twelfth International Conference on Learning Representations*, pp. 1–34, International Conference on Learning Representations, https://proceedings.iclr.cc/paper_files/paper/2024/file/5a68d05006d5b05dd9463dd9c0219db0-Paper-Conference.pdf.

Nils KÖBIS and Luca D. MOSSINK (2021), Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry, *Computers in human behavior*, 114:1–13, https://www.sciencedirect.com/science/article/pii/S0747563220303034.

Maria KUTEEVA and Marta ANDERSSON (2024), Diversity and standards in writing for publication in the age of AI–between a rock and a hard place, *Applied Linguistics*, 45(3):561–567, doi:10.1093/applin/amae025, https://doi.org/10.1093/applin/amae025.

Jey Han LAU, Trevor COHN, Timothy BALDWIN, Julian BROOKE, and Adam HAMMOND (2018), Deep-speare: A joint neural model of poetic language, meter and rhyme, in Iryna GUREVYCH and Yusuke MIYAO, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1948–1958, Association for Computational Linguistics, doi:10.18653/v1/P18-1181, `https://aclanthology.org/P18-1181`.

RAY LC (2022), Imitations of immortality: Learning from human imitative examples in transformer poetry generation, in Adérito FERNANDES-MARCOS, Paulo Bernardino BASTOS, Maria Manuela LOPES, António ARAÚJO, and Lucas Fabian OLIVERO, editors, *10th International Conference on Digital and Interactive Arts*, pp. 1–9, Association for Computing Machinery, doi:10.1145/3483529.3483537, `https://doi.org/10.1145/3483529.3483537`.

Wenqiang LEI, Yao ZHANG, Feifan SONG, Hongru LIANG, Jiaxin MAO, Jiancheng LV, Zhenglu YANG, and Tat-Seng CHUA (2022), Interacting with non-cooperative user: A new paradigm for proactive dialogue policy, in Luke GALLAGHER and Qingyun WU, editors, *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 212–222, Association for Computing Machinery, doi:10.1145/3477495.3532001, `https://doi.org/10.1145/3477495.3532001`.

Jure LESKOVEC, Daniel HUTTENLOCHER, and Jon KLEINBERG (2010), Signed networks in social media, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1361–1370, Association for Computing Machinery, doi:10.1145/1753326.1753532, `https://doi.org/10.1145/1753326.1753532`.

Chao LI, Xing SU, Haoying HAN, Cong XUE, Chunmo ZHENG, and Chao FAN (2023a), Quantifying the impact of large language models on collective opinion dynamics, `https://arxiv.org/abs/2308.03313`.

Guohao LI, Hasan Abed AL KADER HAMMOUD, Hani ITANI, Dmitrii KHIZBULLIN, and Bernard GHANEM (2023b), Camel: communicative agents for "mind" exploration of large language model society, in A. OH, T. NAUMANN, A. GLOBERSON, K. SAENKO, M. HARDT, and S. LEVINE, editors, *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 51991–52008, Curran Associates Inc., `https://dl.acm.org/doi/10.5555/3666122.3668386`.

Yi LIAO, Yasheng WANG, Qun LIU, and Xin JIANG (2019), GPT-based generation for classical chinese poetry, `https://arxiv.org/abs/1907.00151`.

Bill Yuchen LIN, Yicheng FU, Karina YANG, Faeze BRAHMAN, Shiyu HUANG, Chandra BHAGAVATULA, Prithviraj AMMANABROLU, Yejin CHOI, and Xiang REN (2023), Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks, in A. OH, T. NAUMANN, A. GLOBERSON, K. SAENKO,

M. HARDT, and S. LEVINE, editors, *Thirty-seventh Conference on Neural Information Processing Systems (Volume 36)*, pp. 1–13, https://openreview.net/forum?id=Rzk3GP1HN7.

Zhaojiang LIN, Andrea MADOTTO, Yejin BANG, and Pascale FUNG (2021), The adapter-bot: All-in-one controllable conversational model, in *Proceedings of the AAAI Conference on Artificial Intelligence (Volume 35)*, pp. 16081–16083, AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/18018.

Alisa LIU, Maarten SAP, Ximing LU, Swabha SWAYAMDIPTA, Chandra BHAGAVATULA, Noah A. SMITH, and Yejin CHOI (2021), DExperts: Decoding-time controlled text generation with experts and anti-experts, in Chengqing ZONG, Fei XIA, Wenjie LI, and Roberto NAVIGLI, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706, Association for Computational Linguistics, doi:10.18653/v1/2021.acl-long.522, https://aclanthology.org/2021.acl-long.522/.

Zijun LIU, Yanzhe ZHANG, Peng LI, Yang LIU, and Diyi YANG (2024), A dynamic LLM-powered agent network for task-oriented agent collaboration, in *First Conference on Language Modeling*, pp. 1–30, https://openreview.net/forum?id=XII0Wp1XA9.

Xiaoding LU, Zongyi LIU, Adian LIUSIE, Vyas RAINA, Vineet MUDUPALLI, Yuwen ZHANG, and William BEAUCHAMP (2024), Blending is all you need: Cheaper, better alternative to trillion-parameters LLM, https://arxiv.org/abs/2401.02994.

Jingkun MA, Runzhe ZHAN, and Derek F. WONG (2023), Yu Sheng: Human-in-loop classical Chinese poetry generation system, in Danilo CROCE and Luca SOLDAINI, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 57–66, Association for Computational Linguistics, doi:10.18653/v1/2023.eacl-demo.8, https://aclanthology.org/2023.eacl-demo.8/.

Ridwan MAHBUB, Ifrad KHAN, Samiha ANUVA, Md Shihab SHAHRIAR, Md Tahmid Rahman LASKAR, and Sabbir AHMED (2023), Unveiling the essence of poetry: Introducing a comprehensive dataset and benchmark for poem summarization, in Houda BOUAMOR, Juan PINO, and Kalika BALI, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14878–14886, Association for Computational Linguistics, doi:10.18653/v1/2023.emnlp-main.920, https://aclanthology.org/2023.emnlp-main.920/.

R. Thomas MCCOY, Paul SMOLENSKY, Tal LINZEN, Jianfeng GAO, and Asli CELIKYILMAZ (2023), How much do language models copy from their training data? Evaluating linguistic novelty in text generation using RAVEN, *Transactions of the Association for Computational Linguistics*, 11:652–670, doi:10.1162/tacl_a_00567, https://aclanthology.org/2023.tacl-1.38.

Hugo Gonçalo OLIVEIRA (2012), PoeTryMe: A versatile platform for poetry generation, in Tarek R. BESOLD, Kai-Uwe KUEHNBERGER, Marco SCHORLEMMER, and Alan SMAILL, editors, *Computational Creativity, Concept Invention, and General Intelligence (Volume 1)*, pp. 18–23, https://www.lirmm.fr/ecai2012/images/stories/ecai_doc/pdf/workshop/W40_c3gi_pre-proceedings_20120803.pdf.

Joon Sung PARK, Joseph O'BRIEN, Carrie Jun CAI, Meredith Ringel MORRIS, Percy LIANG, and Michael S. BERNSTEIN (2023), Generative agents: Interactive simulacra of human behavior, in Sean FOLLMER, Jeff HAN, Jürgen STEIMLE, and Nathalie Henry RICHE, editors, *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, Association for Computing Machinery, doi:10.1145/3586183.3606763, https://doi.org/10.1145/3586183.3606763.

Jing QIAN, Li DONG, Yelong SHEN, Furu WEI, and Weizhu CHEN (2022), Controllable natural language generation with contrastive prefixes, in Smaranda MURESAN, Preslav NAKOV, and Aline VILLAVICENCIO, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2912–2924, Association for Computational Linguistics, doi:10.18653/v1/2022.findings-acl.229, https://aclanthology.org/2022.findings-acl.229/.

Nils REIMERS and Iryna GUREVYCH (2019), Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in Kentaro INUI, Jing JIANG, Vincent NG, and Xiaojun WAN, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Association for Computational Linguistics, doi:10.18653/v1/D19-1410, https://aclanthology.org/D19-1410.

Leonardo F. R. RIBEIRO, Yue ZHANG, and Iryna GUREVYCH (2021), Structural adapters in pretrained language models for AMR-to-text generation, in Marie-Francine MOENS, Xuanjing HUANG, Lucia SPECIA, and Scott Wen-tau YIH, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4269–4282, Association for Computational Linguistics, doi:10.18653/v1/2021.emnlp-main.351, https://aclanthology.org/2021.emnlp-main.351/.

Yu-Ping RUAN and Zhen-Hua LING (2023), Emotion-Regularized Conditional Variational Autoencoder for Emotional Response Generation, *IEEE Transactions on Affective Computing*, 14(01):842–848, doi:10.1109/TAFFC.2021.3073809, https://doi.ieeecomputersociety.org/10.1109/TAFFC.2021.3073809.

Piotr SAWICKI, Marek GRZES, Fabricio GOES, Dan BROWN, Max PEEPERKORN, and Aisha KHATUN (2023a), Bits of grass: Does GPT already know how to write like Whitman?, in Alison PEASE, Joao Miguel CUNHA, Maya ACKERMAN, and Daniel G. BROWN, editors, *Proceedings of the 14th International Conference for Computational Creativity*, pp. 312–321, https://computationalcreativity.net/iccc23/ICCC-2023-Proceedings.pdf.

Piotr SAWICKI, Marek GRZES, Luis Fabricio GÓES, Dan BROWN, Max PEEPERKORN, Aisha KHATUN, and Simona PARASKEVOPOULOU (2023b), On the power of special-purpose GPT models to create and evaluate new poetry in old styles, in Alison PEASE, Joao Miguel CUNHA, Maya ACKERMAN, and Daniel G. BROWN, editors, *14th International Conference on Computational Creativity*, pp. 10–19, Association for Computational Creativity, https://computationalcreativity.net/iccc23/ICCC-2023-Proceedings.pdf.

Yizhan SHAO, Tong SHAO, Minghao WANG, Peng WANG, and Jie GAO (2021), A sentiment and style controllable approach for Chinese poetry generation, in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4784–4788, Association for Computing Machinery, doi:10.1145/3459637.3481964, https://doi.org/10.1145/3459637.3481964.

Noam SHAZEER, Azalia MIRHOSEINI, Krzysztof MAZIARZ, Andy DAVIS, Quoc LE, Geoffrey HINTON, and Jeff DEAN (2016), Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, in *International Conference on Learning Representations*, pp. 1–19, International Conference on Learning Representations, https://www.cs.toronto.edu/~hinton/absps/Outrageously.pdf.

Lei SHEN, Xiaoyu GUO, and Meng CHEN (2020), Compose like humans: Jointly improving the coherence and novelty for modern Chinese poetry generation, in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, doi:10.1109/IJCNN48605.2020.9206888, https://ieeexplore.ieee.org/document/9206888.

Emily SHENG, Kai-Wei CHANG, Prem NATARAJAN, and Nanyun PENG (2020), Towards Controllable Biases in Language Generation, in Trevor COHN, Yulan HE, and Yang LIU, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3239–3254, Association for Computational Linguistics, doi:10.18653/v1/2020.findings-emnlp.291, https://aclanthology.org/2020.findings-emnlp.291/.

Guodong SHI, Claudio ALTAFINI, and John S. BARAS (2019), Dynamics over signed networks, *SIAM Review*, 61(2):229–257, doi:10.1137/17M1134172, https://doi.org/10.1137/17M1134172.

Yixuan SU, Tian LAN, Yan WANG, Dani YOGATAMA, Lingpeng KONG, and Nigel COLLIER (2022), A contrastive framework for neural text generation, in

S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 21548–21561, Curran Associates Inc., https://dl.acm.org/doi/10.5555/3600270.3601836.

Guy Tevet and Jonathan Berant (2021), Evaluating the evaluation of diversity in natural language generation, in Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 326–346, Association for Computational Linguistics, doi:10.18653/v1/2021.eacl-main.25, https://aclanthology.org/2021.eacl-main.25.

Huishuang Tian, Kexin Yang, Dayiheng Liu, and Jiancheng Lv (2021), Anchibert: A pre-trained model for ancient Chinese language understanding and generation, in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, doi:10.1109/IJCNN52387.2021.9534342, https://ieeexplore.ieee.org/document/9534342.

Yufei Tian and Nanyun Peng (2022), Zero-shot sonnet generation with discourse-level planning and aesthetics features, in Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3587–3597, Association for Computational Linguistics, doi:10.18653/v1/2022.naacl-main.262, https://aclanthology.org/2022.naacl-main.262/.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan (2022), Memorization without overfitting: analyzing the training dynamics of large language models, in S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 38274–38290, Curran Associates Inc., https://dl.acm.org/doi/10.5555/3600270.3603043.

David Uthus, Maria Voitovich, and R.J. Mical (2022), Augmenting poetry composition with Verse by Verse, in Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pp. 18–26, Association for Computational Linguistics, doi:10.18653/v1/2022.naacl-industry.3, https://aclanthology.org/2022.naacl-industry.3.

Tim Van de Cruys (2020), Automatic poetry generation from prosaic text, in Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2471–2480, Association for Computational Linguistics,

doi:10.18653/v1/2020.acl-main.223,
https://aclanthology.org/2020.acl-main.223.

Guanzhi WANG, Yuqi XIE, Yunfan JIANG, Ajay MANDLEKAR, Chaowei XIAO, Yuke ZHU, Linxi FAN, and Anima ANANDKUMAR (2024a), Voyager: An open-ended embodied agent with large language models, *Transactions on Machine Learning Research*, pp. 1–44,
https://openreview.net/forum?id=ehfRiF0R3a.

Hongyi WANG, Felipe Maia POLO, Yuekai SUN, Souvik KUNDU, Eric P. XING, and Mikhail YUROCHKIN (2024b), Fusing models with complementary expertise, in *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, pp. 1–23, https://neurips.cc/virtual/2023/80510.

Wenlin WANG, Zhe GAN, Hongteng XU, Ruiyi ZHANG, Guoyin WANG, Dinghan SHEN, Changyou CHEN, and Lawrence CARIN (2019), Topic-guided variational auto-encoder for text generation, in Jill BURSTEIN, Christy DORAN, and Thamar SOLORIO, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pp. 166–177, Association for Computational Linguistics, doi:10.18653/v1/N19-1015,
https://aclanthology.org/N19-1015.

Xuezhi WANG, Jason WEI, Dale SCHUURMANS, Quoc V LE, Ed H. CHI, Sharan NARANG, Aakanksha CHOWDHERY, and Denny ZHOU (2023), Self-consistency improves chain of thought reasoning in language models, in *The Eleventh International Conference on Learning Representations*, pp. 1–24, International Conference on Learning Representations,
https://openreview.net/forum?id=1PL1NIMMrw.

Yaqing WANG, Sahaj AGARWAL, Subhabrata MUKHERJEE, Xiaodong LIU, Jing GAO, Ahmed Hassan AWADALLAH, and Jianfeng GAO (2022), AdaMix: Mixture-of-adaptations for parameter-efficient model tuning, in Yoav GOLDBERG, Zornitsa KOZAREVA, and Yue ZHANG, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5744–5760, Association for Computational Linguistics,
doi:10.18653/v1/2022.emnlp-main.388,
https://aclanthology.org/2022.emnlp-main.388/.

Zhe WANG, Wei HE, Hua WU, Haiyang WU, Wei LI, Haifeng WANG, and Enhong CHEN (2016), Chinese poetry generation with planning based neural network, in Yuji MATSUMOTO and Rashmi PRASAD, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1051–1060, The COLING 2016 Organizing Committee,
https://aclanthology.org/C16-1100.

Roosa WINGSTRÖM, Johanna HAUTALA, and Riina LUNDMAN (2023), Redefining creativity in the era of AI? Perspectives of computer scientists and new media artists, *Creativity Research Journal*, 36(2):1–17, `https://doi.org/10.1080/10400419.2022.2107850`.

Jörg WÖCKENER, Thomas HAIDER, Tristan MILLER, The-Khang NGUYEN, Thanh Tung Linh NGUYEN, Minh Vu PHAM, Jonas BELOUADI, and Steffen EGER (2021), End-to-end style-conditioned poetry generation: What does it take to learn from examples alone?, in Stefania DEGAETANO-ORTLIEB, Anna KAZANTSEVA, Nils REITER, and Stan SZPAKOWICZ, editors, *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pp. 57–66, Association for Computational Linguistics, doi:10.18653/v1/2021.latechclfl-1.7, `https://aclanthology.org/2021.latechclfl-1.7`.

Qingyun WU, Gagan BANSAL, Jieyu ZHANG, Yiran WU, Beibin LI, Erkang (Eric) ZHU, Li JIANG, Xiaoyun ZHANG, Shaokun ZHANG, Ahmed AWADALLAH, Ryen W. WHITE, Doug BURGER, and Chi WANG (2024), AutoGen: Enabling next-gen LLM applications via multi-agent conversation, in *First Conference on Language Modeling*, pp. 1–46, `https://openreview.net/forum?id=BAakY1hNKS`.

Rui YAN (2016), I, poet: automatic poetry composition through recurrent neural networks with iterative polishing schema, in Gerhard BREWKA, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2238–2244, AAAI Press, `https://www.ijcai.org/Proceedings/16/Papers/319.pdf`.

Shunyu YAO, Dian YU, Jeffrey ZHAO, Izhak SHAFRAN, Thomas L. GRIFFITHS, Yuan CAO, and Karthik NARASIMHAN (2023), Tree of thoughts: deliberate problem solving with large language models, in A. OH, T. NAUMANN, A. GLOBERSON, K. SAENKO, M. HARDT, and S. LEVINE, editors, *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 11809–11822, Curran Associates Inc., `https://papers.nips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf`.

Xiaoyuan YI, Ruoyu LI, Cheng YANG, Wenhao LI, and Maosong SUN (2020), Mixpoet: Diverse poetry generation via learning controllable mixed latent space, in *Proceedings of the AAAI Conference on Artificial Intelligence (Volume 34)*, 05, pp. 9450–9457, AAAI Press, doi:10.1609/aaai.v34i05.6488, `https://ojs.aaai.org/index.php/AAAI/article/view/6488`.

Hanqing ZHANG, Haolin SONG, Shaoyu LI, Ming ZHOU, and Dawei SONG (2023), A survey of controllable text generation using transformer-based pre-trained language models, *ACM Computing Surveys*, 56(3):1–37, doi:10.1145/3617680, `https://doi.org/10.1145/3617680`.

Ran ZHANG, Jihed OUNI, and Steffen EGER (2024), Cross-lingual cross-temporal summarization: Dataset, models, evaluation, *Computational*

Linguistics, 50(3):1001–1047, doi:10.1162/coli_a_00519, https://aclanthology.org/2024.cl-3.5/.

Xingxing ZHANG and Mirella LAPATA (2014), Chinese poetry generation with recurrent neural networks, in Alessandro MOSCHITTI, Bo PANG, and Walter DAELEMANS, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 670–680, Association for Computational Linguistics, doi:10.3115/v1/D14-1074, https://aclanthology.org/D14-1074.

Chujie ZHENG, Pei KE, Zheng ZHANG, and Minlie HUANG (2023), Click: Controllable text generation with sequence likelihood contrastive learning, in Anna ROGERS, Jordan BOYD-GRABER, and Naoaki OKAZAKI, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1022–1040, Association for Computational Linguistics, doi:10.18653/v1/2023.findings-acl.65, https://aclanthology.org/2023.findings-acl.65.

Guo ZHIPENG, Xiaoyuan YI, Maosong SUN, Wenhao LI, Cheng YANG, Jiannan LIANG, Huimin CHEN, Yuhui ZHANG, and Ruoyu LI (2019), Jiuge: A human-machine collaborative Chinese classical poetry generation system, in Marta R. COSTA-JUSSÀ and Enrique ALFONSECA, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 25–30, Association for Computational Linguistics, doi:10.18653/v1/P19-3005, https://aclanthology.org/P19-3005.

Andrew ZHU, Lara MARTIN, Andrew HEAD, and Chris CALLISON-BURCH (2023), Calypso: LLMs as dungeon masters' assistants, in Markus EGER and Rogelio Enrique CARDONA-RIVERA, editors, *Artificial Intelligence and Interactive Digital Entertainment*, pp. 380–390, AAAI Press, doi:10.1609/aiide.v19i1.27534, https://doi.org/10.1609/aiide.v19i1.27534.

Mingchen ZHUGE, Haozhe LIU, Francesco FACCIO, Dylan R. ASHLEY, Róbert CSORDÁS, Anand GOPALAKRISHNAN, Abdullah HAMDI, Hasan Abed Al Kader HAMMOUD, Vincent HERRMANN, Kazuki IRIE, Louis KIRSCH, Bing LI, Guohao LI, Shuming LIU, Jinjie MAI, Piotr PIĘKOS, Aditya A. RAMESH, Imanol SCHLAG, Weimin SHI, Aleksandar STANIĆ, Wenyi WANG, Yuhui WANG, Mengmeng XU, Deng-Ping FAN, Bernard GHANEM, and Jürgen SCHMIDHUBER (2025), Mindstorms in natural language-based societies of mind, *Computational Visual Media*, 11(1):29–81, doi:10.26599/CVM.2025.9450460, https://ieeexplore.ieee.org/document/10903668.

Ran Zhang

0009-0004-2655-7031

ran.zhang@uni-mannheim.de

School of Business Informatics and Mathematics

University of Mannheim

Steffen Eger

0000-0003-4663-8336

steffen.eger@utn.de

Department Engineering

University of Technology Nuremberg (UTN)

Ran Zhang and Steffen Eger (2025), LLM-based multi-agent poetry generation in non-cooperative environments, *Journal of Language Modelling*, 13(2):261–318
https://dx.doi.org/10.15398/jlm.v13i2.432