LLM-BASED MULTI-AGENT POETRY GENERATION IN NON-COOPERATIVE ENVIRONMENTS

Ran Zhang
School of Business Informatics and Mathematics
University of Mannheim
B6 26, 68159 Mannheim Germany
ran.zhang@uni-mannheim.de

Steffen Eger
School of Business Informatics and Mathematics
University of Mannheim
B6 26, 68159 Mannheim Germany
steffen.eger@uni-mannheim.de

September 9, 2024

ABSTRACT

Despite substantial progress of large language models (LLMs) for automatic poetry generation, the generated poetry lacks diversity while the training process differs greatly from human learning. Under the rationale that the learning process of the poetry generation systems should be more human-like and their output more diverse and novel, we introduce a framework based on social learning where we emphasize non-cooperative interactions besides cooperative interactions to encourage diversity. Our experiments are the first attempt at LLM-based multi-agent systems in non-cooperative environments for poetry generation employing both TRAINING-BASED agents (GPT-2) and PROMPTING-BASED agents (GPT-3 and GPT-4). Our evaluation based on 96k generated poems shows that our framework benefits the poetry generation process for TRAINING-BASED agents resulting in 1) a 3.0-3.7 percentage point (pp) increase in diversity and a 5.6-11.3 pp increase in novelty according to distinct and novel n-grams. The generated poetry from TRAINING-BASED agents also exhibits group divergence in terms of lexicons, styles and semantics. PROMPTING-BASED agents in our framework also benefit from non-cooperative environments and a more diverse ensemble of models with non-homogeneous agents has the potential to further enhance diversity, with an increase of 7.0-17.5 pp according to our experiments. However, PROMPTING-BASED agents show a decrease in lexical diversity over time and do not exhibit the group-based divergence intended in the social network. Our paper 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.

Keywords poetry generation · social learning · multi-agent system

1 Introduction

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

Why poetry generation?

---

[1]The code, data and generated outputs are publicly available at https://github.com/zhangr2021/Multiagent_poetry.git
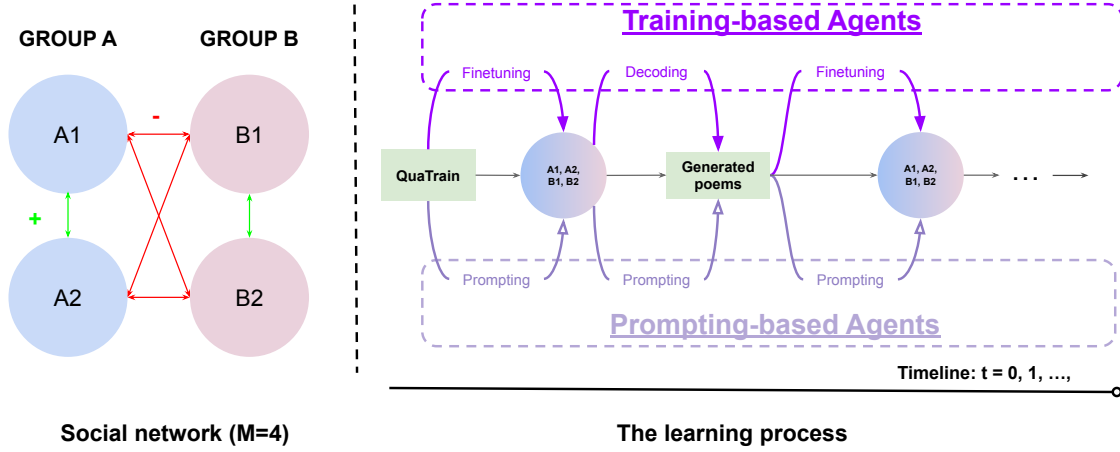
Figure 1: Illustration of the predefined social network ($M = 4$) and the high-level description of the learning process for TRAINING-BASED agents (GPT-2) and PROMPTING-BASED agents (GPT-3.5 and GPT-4). The green and red lines in the social network indicate cooperative (+) and non-cooperative (-) interaction between agents respectively.

> "Poetry is the rhythmical creation of beauty in words."
> – *Edgar Allan Poe*

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]. Models need to excel not only in linguistic competence such as understanding semantics and grammar but also in capturing stylistic elements such as rhyme, meter, imagery, and artistic flair to produce human-like poetry Zhipeng et al. [2019], Belouadi and Eger [2023], Ma et al. [2023]. Current challenges in poetry generation include: 1) limitations of finetuned models or pipelines that are tailored for specific styles or topics Lau et al. [2018], Van de Cruys [2020], Tian and Peng [2022]; 2) LLMs struggling to create diverse and aesthetically pleasing poetry in zero-shot and few-shot scenarios compared to human compositions Sawicki et al. [2023a,b]. The complexity and constraints in poetry generation make it a suitable scenario for multi-agent settings, given the known capacity of multi-agent systems to solve complex tasks and the potential to enhance poetry diversity by leveraging "mixtures of multiple poets" Yi et al. [2020].

Why multi-agent system and social learning?
The current process of generating poetry in machine learning and NLP differs substantially from the way in which humans learn and compose poetry. Different from the paradigm of training one single poetry generation model that learns from particular datasets, human beings learn within a social context where they interact with others through communication, be it spoken or written, formal or non-formal Jarvis [2012]. This raises the question of whether a more human-like generation approach could improve poetry production. Multi-agent systems, with rich applications in social simulation Park et al. [2023], Chuang et al. [2023], are suitable for our goal where human behavior, such as poetry composition, can be effectively replicated through LLM-based agents that can be situated well in social networks. Furthermore, as one of the most fundamental elements for creative composition, *"divergent"* thinking ability (to deviate from established norms) during the learning phase is crucial for both human and computational creativity Elgammal et al. [2017], Brinkmann et al. [2023], Wingström et al. [2023]. This serves as a rationale for us to adopt a social learning process that supports not only collaboration but also opposition as the main source of deviation in the process Eger [2016], Shi et al. [2019]. Moreover, unlike previous studies that focus on weak forms of opposition, such as debates and arguments, which emphasize refining differences through thought correction Chan et al. [2023], our proposed social learning process aims to amplify divergence to enhance diversity.

Why non-cooperative environments?
One aspect of human behavior is non-cooperative interactions in various contexts. For example, the political arena is often characterized by forms of opposition between individual parties or, in a wider context, 'counter-cultures' rebelling against the establishment. Moreover, one defining property of literature / art / philosophy is to distinguish oneself from previous or contemporary 'competitors'. To name some examples, Hemingway's 'iceberg' writing style differs

substantially from his predecessors, characterized by a more sentimental writing style Baker [1972]; Impressionist artists have challenged the standards of paintings set by the conventional art community with new contents and styles; philosopher Arthur Schopenhauer has strongly argued against the philosophy of Hegel and philosophers often form opposing groups, e.g., 'Kantians', 'Neoplatinists', etc 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 leveraged in modeling Gautier et al. [2022], Lei et al. [2022].

How do we utilize LLM-based agents for poetry generation?
We build our social learning framework upon a predefined social network that governs the interactions among agents shown in Figure 1. This network facilitates not only cooperative interactions ("Poets appreciate each other's work and learn from the others") but also non-cooperative interactions ("Poets dislike each other's work and deviate from each other"). We present the learning frameworks based on two different LLMs: 1) PROMPTING-BASED conversational agents (GPT-4) to answer *can our framework benefit in zero-shot or few-shot settings to produce diverse poetry?* and 2) TRAINING-BASED agents (GPT-2), where we investigate various training and decoding configurations such as training losses and the number of interactive agents to determine *which strategy is the most effective in non-cooperative environments for poetry generation?* Our framework comprises three main components: (1) the social network, (2) the learning process and (3) the learning strategy. Our main contribution lies in the development of a novel learning framework that integrates existing methods to extend their scope and effectiveness.

Training-based agents in non-cooperative environments generate poetry of increasing diversity and novelty over time: Our evaluation in Section 5 suggests that our framework benefits the generation process resulting in increasing diversity and novelty according to distinct and novel n-grams for TRAINING-BASED agents. The generated poetry from TRAINING-BASED agents also exhibits group divergence in terms of lexicons, styles and semantics per the predefined group affiliation. Analysis from Section 6 also indicates that non-cooperative conditions boost diversity for PROMPTING-BASED agents and the potential benefits of employing a more diverse ensemble of models with non-homogeneous agents. But PROMPTING-BASED agents in our framework do not exhibit group-based divergence of any kind. Moreover, PROMPTING-BASED agents are prone to generating poetry of homogeneous styles over time.

2 Related work

Our research connects to 1) *the interaction of LLM-based agents* where we focus on the forms of interaction; 2) the corresponding methods to model interactions, i.e., *language model ensemble and controlled text generation* (CTG) where CTG techniques are required for our use case in 3) *poetry generation*.

The interaction of LLM-based agents The form of interactions among agents can be broadly categorized as cooperative and non-cooperative. Very often, agents communicate cooperatively where the aim is to make joint decisions through collaboration such as back-and-forth communication Li et al. [2024], majority voting Hamilton [2023] or a combination of both Zhuge et al. [2023]. Non-cooperative interactions, though less prevalent in comparison, can enhance the quality of responses through debates or arguments among agents Chan et al. [2023], Du et al. [2023]. Moreover, while most studies focus on static interaction among agents, some researchers delve into the dynamics of agents' interactions. Liu et al. [2023] propose a dynamic interaction architecture where they utilize an optimization algorithm to select the best agents at inference time. Autogen Wu et al. [2023] also enables dynamic group chat to guide the flow of interaction among agents during ongoing conversations. Others focus on the output dynamics during the interaction, instead. Chuang et al. [2023] find that LLMs tend to align with factual information regardless of their personas and initial states which limits the simulation of opinion dynamics using LLM-based agents. More recently, the dynamics of group interaction is also studied to show that incorporating chain-of-thought reasoning, detailed personas, and finetuning LLMs can enhance agents' ability to replicate human-like group dynamics Chuang et al. [2024]. Our work differs in that 1) we build a social network with group affiliations to obtain a more human-like learning process; 2) we propose a framework that involves two forms of interaction, especially with an emphasis on non-cooperative communication ; 3) we focus on the output dynamics of the generation process under finetuning for TRAINING-BASED agents and consecutive prompting using detailed personas for PROMPTING-BASED agents.

Language model ensemble and controlled text generation The modeling of our framework requires 1) an ensemble of multiple language models (LMs) and 2) the generated outputs to capture general poetic styles concerning the use case of poetry generation. This involves the areas of *LM ensemble* and *controlled text generation (CTG)*.

LM ensemble can be divided into 1) conversational ensemble which does not involve parameter training Wang et al. [2022a] and 2) finetuning-based ensemble i.e., neural network ensemble Shazeer et al. [2016] and output ensemble Dekoninck et al. [2023], Jiang et al. [2023]. Prompting-based conversational ensemble is often utilized for reasoning tasks where LLMs ensemble their own responses (i.e., self-ensemble) Wang et al. [2022a], Fu et al. [2022]. Very recently, Lu et al. [2024] combine multiple small conversational models in a parameter-efficient and interpretable way that outperforms ChatGPT according to their A/B test. Finetuning-based ensembles can operate at the neural network or output level. While *neural network ensemble* typically requires massive training or finetuning through extensive datasets and resources Shazeer et al. [2016], Jiang et al. [2024], the mixture of smaller modules such as adapters becomes a viable solution in resource-restricted situations Wang et al. [2022b], Chronopoulou et al. [2023].

CTG, the task of generating texts subject to attributes such as emotion Firdaus et al. [2022], Ruan and Ling [2021], topic Dathathri et al. [2019], Wang et al. [2019], toxicity avoidance Liu et al. [2021], style Belouadi and Eger [2023], Shao et al. [2021b], debiasing Dinan et al. [2020], Sheng et al. [2020], etc., is also relevant to our study. CTG can operate at the training/finetuning and inference stage. Similarly to LM ensemble, finetuning with additional modules such as task-related adapters Ribeiro et al. [2021], Lin et al. [2021] is also utilized to gain parameter-efficient controllability. Moreover, for CTG applications to reduce the probability of generating undesirable attributes, in addition to the standard CROSS-ENTROPY loss utilized for text generation finetuning, other loss functions are also explored. Qian et al. [2022] additionally include CONTRASTIVE loss which is crucial for the detoxification task but only partially improves the performance of the sentiment control task. Zheng et al. [2023] also employ a CONTRASTIVE loss on sequence likelihood to decrease the generation probability of negative samples. In comparison, operation at the inference stage is more viable in the era of LLMs Jiang et al. [2023], Wang et al. [2023b], Dekoninck et al. [2023]. Reranking the outputs is a popular solution. For example, Jiang et al. [2023] first rerank the complete candidate outputs from multiple LLMs and then fuse the top-K answers. Reranking the original next-token distribution during the decoding stage is also widely explored such as utilizing discriminators Dathathri et al. [2019] or combining opinions (i.e., output logits) from (anti-)expert models Liu et al. [2021], Dekoninck et al. [2023]. Operation during inference, especially at the decoding stage, offers strong controllability over the generated texts with less requirement on time and computational resources. However, it may cause a slight decrease in text quality Dathathri et al. [2019]. On the other hand, operations at the training/finetuning stage preserve high-quality text with weaker controllability Zhang et al. [2023].

For our use case of poetry generation, we consider both prompting- and finetuning-based ensemble methods. For the finetuning-based method, we operate jointly 1) at the training stage experimenting with standard CROSS-ENTROPY loss and (or) CONTRASTIVE loss to finetune with adapters for better quality texts and efficiency and 2) at the decoding stages to obtain better controllability in the non-cooperative environments.

Automatic poetry generation Early attempts to automatic poetry generation mainly rely on grammatical rules Oliveira [2012], statistical rules Jiang and Zhou [2008], Greene et al. [2010] and neural networks such 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 focus on transformer-based architectures Tian et al. [2021], Shao et al. [2021a]. Although variants of GPT models have demonstrated outstanding performance in many NLP tasks, mixed opinions are observed on their ability to generate poetry. Studies such as Bena and Kalita [2019], Liao et al. [2019], LC [2022] finetune GPT-2 with additional components such as emotion, form and theme and result in moderate to high-quality poems according to human evaluation. Köbis and Mossink [2021] claim that *zero-shot* GPT-2 can generate human-like poems where the best poem according to human selection can match human-written ones but without human preselection, machine-generated poems are easily identifiable. Wöckener et al. [2021] point out that similar to RNN-based models, GPT-2 faces problems learning poetry-specific characteristics such as rhyme and meter. To counter such deficiencies, Belouadi and Eger [2023] propose BYGPT5, an end-to-end token-free model conditioned on rhyme, meter and alliteration. The model can outperform larger models such as GPT-2, ByT5, and ChatGPT (GPT3-3.5) according to both automatic and human evaluation. They also construct a customized corpus QUATRAIN consisting of large-scale machine-labeled pseudo-quatrains to enlarge the finetuning dataset size. Moreover, Sawicki et al. [2023b] claim that GPT-3 finetuned on 300 poems can successfully generate high-quality poems in a specific author's style but GPT-3.5 without finetuning leads to undesirable poems written with similar styles. The findings from Sawicki et al. [2023a] also point out that GPT-3.5 and GPT-4, without finetuning, fail to generate poetry in desired styles. Recently, interactive poetry generation has attracted the attention of researchers as it facilitates human-machine collaboration to generate poetry of more diverse styles and better quality under specific constraints Zhipeng et al. [2019], Uthus et al. [2022]. Ma et al. [2023] propose a post-polishing system that fine-grains GPT-2 generation based on constraints from humans. COPOET from Chakrabarty et al. [2022] finetunes pretrained T5 with <instruction, generation> pairs to enable poetry generation according to human instructions. Their study shows that finetuned T5 model is not only competitive to the larger INSTRUCTGPT model but also successfully collaborates with humans to produce better poems. In our study, we utilize GPT-2 as our base model as it is a good trade-off

between parameter efficiency and language proficiency. Contrary to most poetry generation objectives that optimize few poetry-specific characteristics, we leverage poems with arbitrary styles where we initialize our models with randomly drawn samples from QUATRAIN corpus that contain pseudo poetic features. Additionally, we also explore the potential of GPT-3.5 and GPT-4 in an interactive environment enabled by a multi-agent system.

3 Social learning framework for poetry generation

This section introduces our social learning framework for poetry generation. The recent development of LLMs has incentivized various attempts to simulate the social learning processes of human individuals via LLM-based agents Li et al. [2023], Chuang et al. [2023], Gao et al. [2023]. Our framework, inspired by such processes, applies a social network-based approach to poetry generation. We investigate whether a more human-like learning process (i.e., social learning) can facilitate poetry generation. We differ from the previous studies in two aspects: 1) We base our architecture on a signed network where agents not only interact in cooperative but also in non-cooperative manners; 2) we introduce the learning framework for PROMPTING-BASED agents (GPT-3.5 and GPT-4) and TRAINING-BASED agents (GPT-2). Our framework consists of three parts: (1) the social network, (2) the learning process and (3) the learning strategy. We describe these below.

3.1 The social network

We consider a signed social network with M LLM-empowered agents where each link between two agents is associated with a positive or negative sign Leskovec et al. [2010], Eger [2016], Shi et al. [2019]. We divide the M agents into two groups as shown 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. We expect two types of interaction between agents based on group affiliation: 1) 'in-group' agents cooperate closely with one another as 'friends' (positive sign); 2) 'out-group' agents are 'foes' and they adjust their 'opinions' in a non-cooperative manner (negative sign). We call the learning process associated with 'in-group' agents **positive learning** and 'out-group' associated learning is **negative learning**. Simultaneous learning from both 'in-group' and 'out-group' is called **joint learning** by us.

In our application for poetry generation, the agents are pretrained LLMs. We view the LLMs as different poets belonging to two groups. The 'in-group' poets appreciate each other's work and aim to learn from their styles. Conversely, the 'out-group' poets dislike each other's work and aim to differentiate their works.

3.2 The learning process

Variable	Definition
$a_1, a_2, \ldots$	agents belong to group A
$b_1, b_2, \ldots$	agents belong to group B
M	the total number of agents
$\mathcal{A}_i$	the target agent $\mathcal{A}_i \in \{a_1, b_1, a_2, b_2, \ldots\}$ where $i \in \{1, 2, \ldots, M\}$
$\mathcal{A}_i, \mathcal{A}^+, \mathcal{A}^-$	the agent tuple: (the target agent, the 'in-group' agents of the target agent, the 'out-group' agents). E.g., $(a_1, [a_2, a_3], [b_1, b_2])$
$P_{\mathcal{A}_i}, P_{*^+}, P_{*^-}$	the conditional probability distribution for the next token of agent $\mathcal{A}_i$, agents $*^+ \in \mathcal{A}^+$ and agents $*^- \in \mathcal{A}^-$
N	the total number of generated poems per iteration per agent
T	the total number of iterations
t	the iteration number $t \in \{1, 2, \ldots, T\}$
$o^{\mathcal{A}_i}$	a generated poem by agent $\mathcal{A}_i$
$O_t^{\mathcal{A}_i}$	the set of poems generated by agent $\mathcal{A}_i$ at iteration t
S_t	the set of all generated poems at iteration t
F_{generate}	a generation function of agent tuple $(\mathcal{A}_i, \mathcal{A}^-, \mathcal{A}^+)$
F_{update}	an updating 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
#$\mathcal{A}$	the number of interactive agents at the decoding stage

Table 1: Notations

---

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

Algorithm 1: Social learning process for poetry generation

Now, we describe the learning process among agents based on the social network shown in Figure 1 and Algorithm 1. We summarize all the notations mentioned below in Table 1. The learning process represents the high-level communication procedure among agents. 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 of agent initialization). We further divide the learning process into two phases: the UPDATE phase and the GENERATE phase, defining the learning strategy $(F_{\text{update}}, F_{\text{generate}})$. The two functions jointly organize the positive, negative and joint learning process. F_{generate} is a generation function that outputs poems O. F_{update} is a learning function that equips agents with the latest knowledge based on the generated poems. We denote an agent as $\mathcal{A}_i$ where $i \in \{1, 2, \dots, M\}$. The 'out-group' agents for $\mathcal{A}_i$ are referred to as $\mathcal{A}^-$ and the 'in-group' agents as $\mathcal{A}^+$. The generation function is thus $F_{\text{generate}}(\mathcal{A}_i, \mathcal{A}^-, \mathcal{A}^+)$.

At each iteration t, the learning process starts with the GENERATE phase. Each agent $\mathcal{A}_i$ generates a set of N poems through function $F_{\text{generate}}(\mathcal{A}_i, \mathcal{A}^-, \mathcal{A}^+)$. We iterate over all agents and collect a set of poems S_t which consists of all generated poems O_t from the current iteration t. Then, we let agent $\mathcal{A}_i$ update their knowledge and learn cooperatively, non-cooperatively or jointly from each other based on poems from the current iteration S_t and the previous iteration S_{t-1}.[2] We denote the update function as $F_{\text{update}}(S_t, S_{t-1})$. We iterate over all agents $\mathcal{A}_i$ for $i \in \{1, 2, \dots, M\}$ until all agents have been updated. We conduct the learning process T times.

3.3 The learning strategy

As mentioned in Section 3.2, the learning process involves positive, negative and joint learning which operates at the GENERATE and the UPDATE phase. We now discuss the detailed learning strategy for TRAINING-BASED agents and PROMPTING-BASED agents. For TRAINING-BASED agents, the learning strategy contains finetuning strategies for the UPDATE phase & decoding strategies for the GENERATE phase. For PROMPTING-BASED agents, both phases are conducted via prompting. Table 2 summarizes the learning strategies for both types of agents.

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

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

---

[2]We utilize the generation from the current and previous iteration to expand the dataset size of the finetuning step where the agent can update to the latest knowledge and prevent potential catastrophic forgetting Biesialska et al. [2020].

3.3.1 Training-based agents

We first introduce the decoding strategy for the GENERATE phase where we utilize reranking techniques. We then detail the finetuning strategy for the UPDATE phase where we employ the CONTRASTIVE loss and the standard CROSS-ENTROPY loss.

Decoding strategy at the GENERATE phase

We adopt the DEXPERT framework, where models learn through a comparison and contrast mechanism Liu et al. [2021]. By reranking the probability distribution of the next token, the target agent $\mathcal{A}_i$ generates the next token by jointly considering the probability distribution of the target agent $\mathcal{A}_i$ itself, its 'in-group' agents $\mathcal{A}^+$ and 'out-group' agents $\mathcal{A}^-$. At the decoding stage, the number of interactive agents involved can vary depending on the specific subset chosen from the sets $\mathcal{A}^+$ and $\mathcal{A}^-$, denoted as $\mathcal{A}_\#^+$ and $\mathcal{A}_\#^-$. The number of interactive agents is thus $\#\mathcal{A}$.[3] This flexibility allows the system to dynamically adjust the number of interactive agents participating in the decoding process, offering adaptability based on the requirements or constraints of the task. The detailed formulation of the decoding strategy is shown below.

Given the input sequence at generation time t_g (g indicates the *generation* stage), denoted as $\boldsymbol{x}_{<t_g}$, we predict the next token x_{t_g} generated by the target agent $\mathcal{A}_i$ through combining the outputs from $\mathcal{A}_\#^+$ and $\mathcal{A}_\#^-$. We first obtain the conditional logit scores of all models denoted by $l_{\mathcal{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g}), l_{*^+}(x_{t_g}|\boldsymbol{x}_{<t_g}), l_{*^-}(x_{t_g}|\boldsymbol{x}_{<t_g})$, where $*^+ \in \mathcal{A}_\#^+$ is an agent belonging to the interactive 'in-group' agents $\mathcal{A}_\#^+$ and $*^- \in \mathcal{A}_\#^-$; $l_*(x_{t_g}|\boldsymbol{x}_{<t_g}) \in \mathbb{R}^{|\mathcal{V}|}$ and $\mathcal{V}$ is the vocabulary. The probability distribution of the next token over the vocabulary $\mathcal{V}$ is $P_*(x_{t_g}|\boldsymbol{x}_{<t_g}) = \text{softmax}[l_*(x_{t_g}|\boldsymbol{x}_{<t_g})]$. The probability distribution of the next token $\hat{P}_{\mathcal{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g})$ is thus given by

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

The next token x_{t_g} is assigned with high probability if the probability is high under both $P_{\mathcal{A}_i}$ and P_{*^+} and low under P_{*^-}. Moreover, if we replace P_{*^+} with $P_{\mathcal{A}_i}$, the process considers 'out-group' agents $\mathcal{A}^-$ only which solely models negative learning. This decoding strategy can model both negative learning ($P_{\mathcal{A}_i}, P_{*^-}$) and joint learning ($P_{\mathcal{A}_i}, P_{*^-}, P_{*^+}$) as shown in Table 2.

Finetuning strategy at the UPDATE phase We discuss the finetuning strategies based on the learning relationships, i.e., positive and joint learning as shown in Table 2.

- *Positive learning*: we utilize the conventional finetuning method with CROSS-ENTROPY loss ($\mathcal{L}_{\text{CE}}$) to finetune agent $\mathcal{A}_i$ with poems $o \in O_{\mathcal{A}_i} \cup O_{\mathcal{A}^+}$ (poems generated from $\mathcal{A}_i$ and $\mathcal{A}^+$). The loss function for the jth poem o_j in a mini-batch is thus

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

  where $\mathcal{T}$ denotes the number of tokens for the poem o_j. $P_{\mathcal{A}_i}(x_{t_g}|\boldsymbol{x}_{<t_g})$ is the conditional distribution of the token at time t_g for poem o_j given the previous sequence $\boldsymbol{x}_{<t_g}$.

- *Joint learning with* CONTRASTIVE *loss*: Our social network design well suits the setting for CONTRASTIVE learning where poems from 'in-group' agents are positive samples and poems from 'out-group' agents are negative samples. We adopt CONTRASTIVE learning to pull closer the semantic representation of 'in-group' samples and push apart that of 'out-group' samples. We implement the CONTRASTIVE loss SIMCSE proposed by Gao et al. [2021]. For a mini-batch containing samples from $\mathcal{A}_i, \mathcal{A}^+, \mathcal{A}^-$, let $(o_j^{\mathcal{A}_i}, o_j^{\mathcal{A}^+}, o_j^{\mathcal{A}^-})$ denote the jth paired triple and $(\boldsymbol{h_j}, \boldsymbol{h_j}^+, \boldsymbol{h_j}^-)$ be its representation. The CONTRASTIVE loss is thus

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

---

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

where Q is the size of mini-batch, τ is the temperature and $sim(h_1, h_2)$ is the cosine similarity $\frac{h_1^\top h_2}{\|h_1\| \cdot \|h_2\|}$. Here, we experiment with CONTRASTIVE loss (a) alone and (b) jointly with $\mathcal{L}_{CE}$ for 'in-group' poems.

- *Joint learning with conditioned* CROSS-ENTROPY *loss*: We utilize the style constraints Belouadi and Eger [2023] using conditions <positive> and <negative> for poems generated by 'in-group' agents and 'out-group' agents. We then finetune the agent $\mathcal{A}_i$ with CROSS-ENTROPY loss.

3.3.2 Prompting-based agents

The learning strategy for PROMPTING-BASED agents relies on prompting. Our prompts are constructed with three modules: 1) a profile module, which defines the role of $\mathcal{A}_i$; 2) a memory module, which stores the generated poems; and 3) an action module, which completes the generation task. Table 8.1 in the appendix contains the prompts for both prompting strategies. For PROMPTING-BASED agents, the UPDATE and GENERATE phases do not function in isolation. F_{update} updates the profile module during prompting based on the generated poems from previous iterations. Similar to the design of TRAINING-BASED agents, we can update the knowledge of agents based on different learning relationships:

- The chain-prompting strategy for isolated positive & negative learning: For chain-prompting, we update the knowledge of $\mathcal{A}_i$ based on its relationships with other agents in a separate manner. At iteration t, we first update the profile of $\mathcal{A}_i$ with poems generated from $\mathcal{A}^+$ at time $t-1$. The process is denoted as $F_{update}(\mathcal{A}_i, \mathcal{A}^+)$. $\mathcal{A}_i$ thus generates a poem $o^{\mathcal{A}_i}$ based on the positive learning results (example prompt: "Please read the poems from your friends carefully and compose similarly to your friend."). We then update the profile of $\mathcal{A}_i$ with the poem $o^{\mathcal{A}_i}$ and a poem $o^{\mathcal{A}^-}$ sampled from the previous iteration $t-1$. This process is denoted as $F_{update}(\mathcal{A}_i, \mathcal{A}^-)$. $\mathcal{A}_i$ thus recomposes the poem $o^{\mathcal{A}_i}$ based on the negative learning results (example prompt: "Please rewrite your poem to compose dissimilarly to your foe.").

- Joint prompting for joint learning: Joint prompting updates the profile with poems generated by $\mathcal{A}^+$ and $\mathcal{A}^-$ at the same time, denoted as $F_{update}(\mathcal{A}_i, \mathcal{A}^-, \mathcal{A}^+)$.

4 Experiments

4.1 Agent initialization

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

Table 3: Instances from QUATRAIN corpus.

Initialization for TRAINING-BASED agents We choose $M = 4$ for our initial experiments. We first pretrain GPT-2 (medium) with a subset of the random QUATRAIN corpus of size 123K (nearly 1/6 of QUATRAIN corpus). QUATRAIN consists of machine-labeled pseudo-quatrains that are consecutive sequences with four lines extracted from real human poems as shown in Table 3. The poems follow poetic characteristics, i.e., rhyme, meter and alliteration. We pre-select the QUATRAIN instances excluding those that are semantically similar (> 0.7) with each other based on pairwise cosine similarity of sentence embeddings calculated using SBERT Reimers and Gurevych [2019]. We first pretrain GPT-2 further with 720 training steps using the pre-selected QUATRAIN dataset. More details of the pretraining and the loss curve are shown in Section 8.2. We then finetune the pretrained model with four randomly selected subsets of size 7.5k from our 123K subcorpus to obtain different initializations of four agents. The initialization step prepares

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

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

models with a preliminary understanding of poetic structures and characteristics essential for subsequent learning phases.

Initialization for PROMPTING-BASED agents For PROMPTING-BASED agents, we randomly sample QUATRAIN instances and initialize the agent with chain-prompting and joint-prompting under the predefined profile shown in Table 12 and Table 13.

4.2 Experimental setup

For TRAINING-BASED agents, we design experiments to explore how the parameters from the finetuning stage (i.e., loss functions) and the decoding stage (number of agents and the scaling parameter) affect the dynamics of generation. We summarize the detailed setup in Table 4.

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

4.3 Evaluation

We first conduct automatic evaluation where we study the generation dynamics of our framework from lexical perspectives. We study lexical novelty and diversity, as novelty and diversity are crucial indicators for creative tasks such as poetry generation. We then study the dynamics from semantic (semantic similarity) perspectives. Lastly, we evaluate the poems qualitatively where we directly compare the generated poetry.

Metric for lexical diversity and novelty. We measure the lexical diversity using the percentage of distinct uni-grams (*distinct-1*) and bi-grams (*distinct-2*) following the definition by Su et al. [2022], Tevet and Berant [2021]. The formulation is given as: $\frac{\text{unique n-grams}(O)}{\text{total n-grams}(O)}$, where O is the set of generated poems to be evaluated. We adopt the measure of novelty (*novelty-1* and *novelty-2*) by McCoy et al. [2023], Shen et al. [2020] where we calculate the number of new uni-/bi-grams that do not stem from the pretraining set and rescale them with the total number of generated tokens. Thus, novelty reflects the lexical difference between the generated poems and the training set, while diversity indicates the token variety among the generated poems.

Metric for group-based semantic similarity. Following our group affiliation, we examine the group dynamics of the agents by their semantics. For any pair of poems sampled from the same iteration t, we calculate the *semantic similarity* of the paired instances by computing the cosine similarity of the embeddings retrieved from SBERT Reimers and Gurevych [2019]. We then aggregate the similarity scores per iteration by their group affiliation (i.e., 'in-group' and 'out-group') defined in Figure 1.
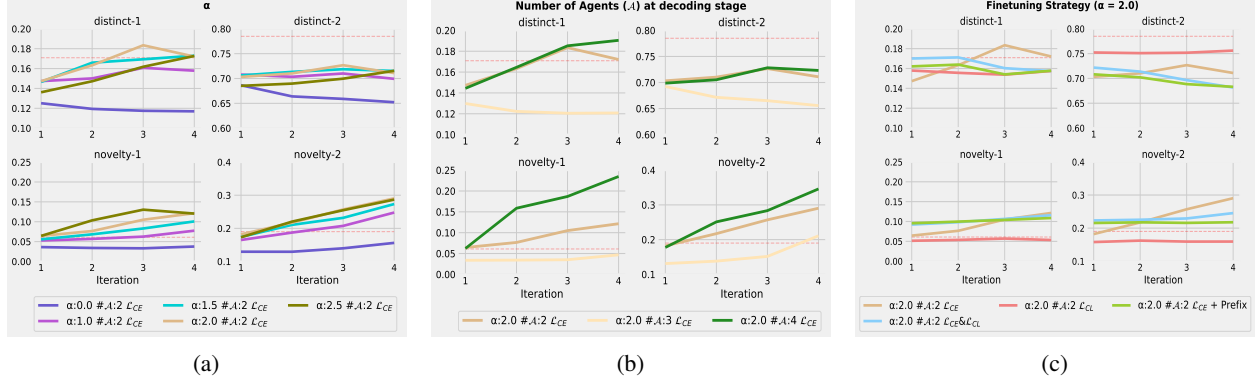
Figure 2: Dynamics of agent diversity and novelty over varying training parameters. The degree of diversity is measured by the percentage of distinct uni-grams (*distinct-1*) and bi-grams (*distinct-2*) in the generated poems. The degree of novelty is measured by the number of novel uni-grams (*novelty-1*) and bi-grams (*novelty-2*) in the generated poems compared to that in 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 #$\mathcal{A}$ during the decoding stage. (c) The effect of finetuning strategies: $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{CL}}$ indicate CROSS-ENTROPY loss and CONTRASTIVE loss. Prefix refers to the conditioned finetuning. The horizontal red dashed line indicates the state of initial agents at iteration 0.

5 Experiment results

5.1 Automatic Evaluation: the generation dynamics of agents

We generate 400 poems using the same set of decoding parameters for TRAINING-BASED agents or the same prompt templates for PROMPTING-BASED agents from each agent for every iteration.[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.

Our main findings are: 1) according to lexical level comparison (i.e., distinct and novel uni-/bi-grams), our framework benefits TRAINING-BASED agents resulting in increasing diversity and a higher level of novelty; 2) according to pairwise semantic similarity averaged per group affiliation, we observe group divergence in semantics for TRAINING-BASED agents, especially 'out-group' divergence due to operation at the decoding stage; 3) PROMPTING-BASED agents generate poems of more diverse lexicons at $t = 1$ but they tend to output poems of homogenous styles over time.[5]

5.1.1 Increasing diversity and novelty according to distinct and novel n-grams for TRAINING-BASED agents

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 and Table 5 shows the results of different experimental setups for diversity and novelty averaged over all iterations. We observe that:

- the effect of negative decoding strategy with the scaling parameter α.
  - *Diversity* Negative decoding combined with positive finetuning ($\alpha > 0$, $\mathcal{L}_{\text{CE}}$) 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) suggest that compared to the case without negative decoding (i.e., $\alpha = 0$), negative decoding strategy under varying α ranging from 1 to 2.5 yields 3.4 to 4.7 percentage points (pp) increase in *distinct-1* and 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

---

[4]See Section 8.2 in the appendix for more details on parameter setting.

[5]Due to resource limit, we do not conduct multiple runs for all experiments. We study the stability of our statistics in Section 6.1. This section is based on one single run of experiments.

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

Table 5: Diversity and novelty results in aggregative 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; #$\mathcal{A}$ is the number of interactive agents at decoding stage; $\mathcal{L}$ represents the loss function during finetuning. *Initialization* indicates the states of initial agents at iteration 0.

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. Worth noting is 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 positive finetuning ($\alpha > 0$, $\mathcal{L}_{\text{CE}}$) 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 aggregative mean by a maximum of 7.0 pp in *novelty-1* and a 9.8 pp increase 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 (#$\mathcal{A}$) involved at the decoding stage

  - **Diversity** As shown in Table 5, #$\mathcal{A} = 4$ yields the highest diversity level according to *distinct-1* and *distinct-2* with #$\mathcal{A} = 2$ achieving similar performance. Dynamically, diversity increases over iteration for paired agents (#$\mathcal{A} = 2$ or 4) at the decoding stage. However, for #$\mathcal{A} = 3$, we observe a decreasing trend in diversity with much lower level of diversity compared to the case for #$\mathcal{A} = 2$ or 4.

  - **Novelty** We observe a greater gain for novelty at #$\mathcal{A} = 4$. This is evident: 1) Table 5 shows 6.9 pp increase in aggregative mean for #$\mathcal{A} = 4$ compared to #$\mathcal{A} = 2$; 2) Figure 2b indicates a shaper increasing trend at #$\mathcal{A} = 4$ 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 time. However, we observe less novelty for #$\mathcal{A} = 3$, which is similar to the case for diversity.

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

To sum up, our framework can lead to increasing diversity and a higher level of novelty: 1) negative decoding combined with positive finetuning ($\alpha > 0$, $\mathcal{L}_{\text{CE}}$) is the most effective combination of decoding and finetuning strategies; 2) the

increase in an even number of agents can improve the results, especially for novelty; 3) in our experiment, positive finetuning (i.e., finetuning using CROSS-ENTROPY loss alone) is more effective overall both in aggregative mean and dynamically compared to other finetuning strategies.

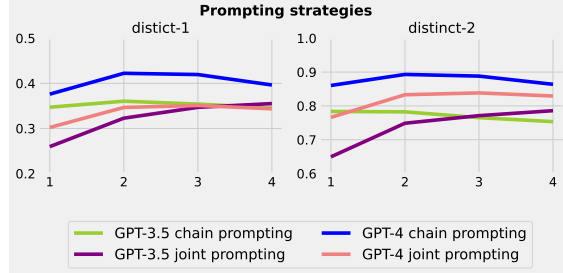**RQ2: how do different prompting strategies affect the diversity of PROMPTING-BASED agents?** As

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

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

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

PROMPTING-BASED agents do not involve further pretraining, novelty metrics, which involve comparison to the pretraining dataset, are thus 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 PROMPTING-BASED agents similar to that of the TRAINING-BASED agents? Different from the trend we observe for TRAINING-BASED agents, PROMPTING-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 yield slightly decreasing trends for nearly all experiments. GPT-3.5 under joint-prompting is an exceptional case where the increasing trend continues mildly. We examine the effect of positive and negative learning strategies in separation 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 aggregative mean in Table 6 where GPT-3.5 under chain-prompting strategy delivers the second best performance according to *distinct-1*.

For PROMPTING-BASED agents, our framework only benefits the generation process in a limited manner (when $t = 1, 2$) according to lexical diversity. Worth noting is that PROMPTING-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 PROMPTING-BASED agents.

5.1.2 Group divergence in semantics

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 $\#\mathcal{A}$ involved during the decoding stage and different finetuning 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 $\#\mathcal{A}$. 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 finetuning 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 yield an 8.8 pp decrease in
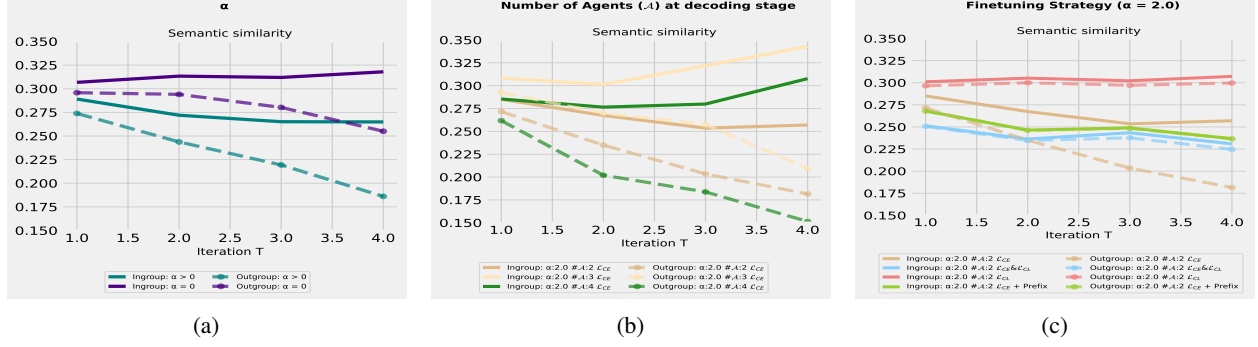
Figure 4: Divergence of TRAINING-BASED agents measured by mean of pairwise semantic similarity over varying training parameters. (a) The effect of scaling parameter α in Equation (1). (b) The effect of the number of interactive agents $\#\mathcal{A}$ during the decoding stage. (c) The effect of finetuning strategies: $\mathcal{L}_{CE}$ and $\mathcal{L}_{CL}$ indicate CROSS-ENTROPY loss and contrastive loss. Prefix refers to the conditioned finetuning. The solid line and dashed line represent semantic similarity measured for 'in-group' and 'out-group' affiliations respectively.
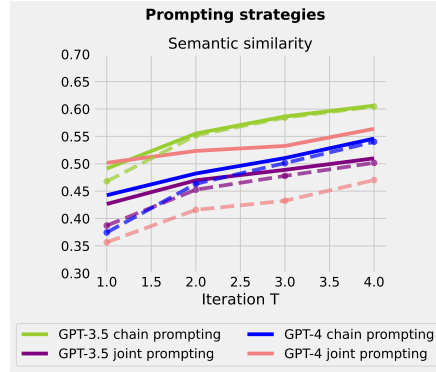

Figure 5: Group divergence of PROMPTING-BASED agents measured by the mean of pairwise semantic similarity over varying prompting strategies and base model. The solid line and dashed line represent semantic similarity measured for 'in-group' and 'out-group' affiliations respectively.

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. $\#\mathcal{A} = 4$ 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, $\#\mathcal{A} = 2$ results in an increase with 2.8 pp for 'in-group' and 9 pp decrease for 'out-group' (11.8 divergence in total). Overall, $\#\mathcal{A} = 4$ exhibits a lower level of similarity compared to $\#\mathcal{A} = 2$.

- *Inseparable 'in-group' and 'out-group' dynamics resulting from other joint finetuning strategies.* Figure 4c shows the outcome for different finetuning strategies involving multiple losses $\mathcal{L}$ and conditioned finetuning (i.e., Prefix). Except for the case using $\mathcal{L}_{CE}$ alone as the finetuning loss (i.e., positive finetuning defined in Table 4), all other cases with joint finetuning exhibit inseparable dynamics between 'in-group' and 'out-group' similarity. We suspect that for contrastive learning ($\mathcal{L}_{CL}$), 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 finetuning as well.

RQ4: how do different prompting strategies affect the group dynamics of PROMPTING-BASED agents?

Undesirable increasing semantic similarity from 'out-group' agents Figure 5 shows the group divergence of PROMPTING-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 observed at $t = 1$. Over time, the agents tend to generate semantically similar poetry for both GPT-3 and

t	group A	group B
	TRAINING-BASED *agents* ($\alpha = 2, \#\mathcal{A} = 2, \mathcal{L}_{CE}$)	
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
	PROMPTING-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**.

Table 7: Generated poetry based on group affiliation from TRAINING-BASED and PROMPTING-BASED agents at different iteration t. For PROMPTING-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.

GPT-4. Moreover, we also notice that PROMPTING-BASED agents generate poetry of homogeneous styles over time which coincides the finding of Sawicki et al. [2023a]. We discuss this point in more detail in Section 5.2.

5.2 Qualitative evaluation

Table 7 contains examples of generated poetry from TRAINING-BASED agents under positive finetuning and negative decoding strategy (i.e., $\alpha = 2, \#\mathcal{A} = 2, \mathcal{L}_{CE}$) and from PROMPTING-BASED agents using GPT-3.5 under the chain-prompting strategy. We select examples composed with similar themes, i.e., child or youth. Poems generated at $t = 0$ are considered the baselines for both TRAINING-BASED and PROMPTING-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 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 $\alpha = 2, \#\mathcal{A} = 2, \mathcal{L}_{CE}$ 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 languages 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

	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)
PROMPTING-BASED	GPT-3.5 chain-prompting	0.007 (.322)	0.007 (.755)	-	-

Table 8: Stability of three simulation results measured by standard deviation. The mean values of all three simulations are reported in brackets.

group A and group B also suggests such divergence. For example, words such as "mind, thy, thee, nature, art, power, happy, hath, young, pleasant, 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, dream, ...". For PROMPTING-BASED agents, in contrast, we observe more diverse lexicons for both groups compared to TRAINING-BASED agents but the two groups in PROMPTING-BASED setting hardly diverge in terms of vocabulary or topics when $t > 1$. This is also suggested by Figure 5. Moreover, PROMPTING-BASED agents tend to generate poems of homogenous styles over time. As shown in Table 7, poems generated from PROMPTING-BASED agents excessively focus on rhymes which makes the generated poetry merely superficially human-like. This also suggests a poor understanding of poetry for GPT-3.5 and GPT-4 in a zero-shot setting. 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' for rhyming, GPT-3.5 and GPT-4 also tend to generate poems using similar beginning phrases such as "Beneath/Under XXX, In the XXX, 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.

6 Discussion and analysis

6.1 How stable are the simulation results?

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, i.e., $\alpha = 0$ and $\alpha = 2$, and one experiment setting for PROMPTING-BASED agents, i.e., GPT-3,5 chain-prompting. We rerun the experiments three times under the same parameters (or prompt templates for PROMPTING-BASED agents) and initialization. We then yield three sets of statistics and calculate the standard deviation as our stability measure. We study the stability from two perspectives: 1) stability of the aggregative mean and 2) dynamic stability.

Stability of the aggregative mean. The stability results of the aggregative mean are shown in Table 8. We observe a low level of variation with less than 0.7 pp for both TRAINING-BASED and PROMPTING-BASED agents.

Dynamic stability. The stability results for our dynamic statistics are shown in Table 9. 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 PROMPTING-BASED agents, we observe a greater level of variation with the highest standard deviation to the level 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 PROMPTING-BASED agents.

Overall, our simulations indicate high stability over the statistics, especially for TRAINING-BASED agents. The PROMPTING-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).

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)
PROMPTING-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)	-	-

Table 9: 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 brackets.

model	strategy	distinct-1	distinct-2
GPT-4	positive	0.286	0.598
GPT-4	negative	0.313	0.653
GPT-4	joint (positive + negatives)	0.336	0.817

Table 10: Diversity results in aggregative mean for PROMPTING-BASED agents under different learning strategies. *Distinct-1* and *distinct-2* are the percentage of distinct uni-/bi-grams.

6.2 The effect of different learning strategies and heterogeneous models for PROMPTING-BASED agents

Non-cooperative environments boost diversity. To examine the effect of learning strategies for PROMPTING-BASED agents, we utilize the same experimental parameters as in Section 4 and conduct generation under positive-only, negative-only, and joint learning strategies (joint-prompting), respectively. As shown in Table 10, the joint learning strategy, which employs both positive and negative steps, is the most effective in terms of the diversity of the generated poetry, yielding a 5.0 pp increase in *distinct-1* and over a 20 pp increase in *distinct-2* compared to the positive-only strategy. Moreover, the negative-only strategy enhances diversity compared to the positive-only strategy, but to a lesser extent than the joint approach.

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

Table 11: Diversity results in aggregative mean for PROMPTING-BASED agents with heterogeneous models.

Heterogeneous models can boost the diversity of the system. To test the effect of using non-homogeneous agents, we use a combination of various models to conduct experiments using joint-prompting defined in Section 4. As shown in Table 11, when GPT-4 is combined with GPT-3.5, 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 along with GPT-4 and GPT-3.5 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. This demonstrates the potential benefits of employing a more diverse ensemble of models.

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

As discussed in Section 5, the framework built with PROMPTING-BASED agents does not exhibit any group-based behavior as expected. Considering that we initialize the agents with random samples drawn from the QUATRAIN corpus, we suspect this may cause a high resemblance among the initialized 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 school children 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. Slightly surprisingly, we observe a very similar trend for both diversity and semantic divergence to that of random initialized PROMPTING-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 7. 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 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 observation coincides with the work from Chuang et al. [2023], Tirumala et al. [2022] that larger models suffer more from memorization.

7 Concluding remarks

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 PROMPTING-BASED agents employing GPT-3.5 and GPT-4. Our evaluation with 96k generated poems shows that for TRAINING-BASED agents: 1) non-cooperative environments encourage diversity and novelty over iteration measured by distinct and novel n-grams; 2) TRAINING-BASED agents demonstrate group divergence in terms of lexicons, styles and semantics in accordance to the predefined group affiliation. Our results also indicate that for PROMPTING-BASED agents: 3) the generated poetry contains very few grammatical errors with a more diverse lexicon; 4) the PROMPTING-BASED framework benefits from non-cooperative environments and heterogeneous model in terms of aggregated diversity; 5) dynamically, PROMPTING-BASED framework barely improves lexical diversity after the first iteration and unlike TRAINING-BASED agents, PROMPTING-BASED agents do not show group-based divergence as expected; 6) PROMPTING-BASED agents are prone to generating poetry of more homogeneous styles over time, presumably suggesting the memorization problems of LLMs.

Nowadays, more researchers have raised concerns 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 such as RLHF (i.e., reinforcement learning from human feedback) produce less diverse text Kirk et al. [2023], 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 thus 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. [2022a].

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. [2023] and thus boost diversity and novelty to a greater level. For PROMPTING-BASED agents, involving more complex reasoning methods such as tree-of-thought Yao et al. [2024] into the prompting might be helpful. Extending the current framework to include an interactive combination of both TRAINING-BASED and PROMPTING-BASED agents might be interesting to explore in which a diverse network of LLMs might bring additional generation diversity to the system.

References

Abdollah Amirkhani and Amir Hossein Barshooi. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935, 2022.

Carlos Baker. *Hemingway, the writer as artist*. Princeton University Press, 1972.

Jonas Belouadi and Steffen Eger. 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)*, pages 7364–7381, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.406. URL https://aclanthology.org/2023.acl-long.406.

Brendan Bena and Jugal Kalita. Introducing aspects of creativity in automatic poetry generation. In *Proceedings of the 16th International Conference on Natural Language Processing*, pages 26–35, 2019.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, 2020.

Levin Brinkmann, Fabian Baumann, Jean-François Bonnefon, Maxime Derex, Thomas F Müller, Anne-Marie Nussberger, Agnieszka Czaplicka, Alberto Acerbi, Thomas L Griffiths, Joseph Henrich, et al. Machine culture. *Nature Human Behaviour*, 7(11):1855–1868, 2023.

Tuhin Chakrabarty, Arkadiy Saakyan, and Smaranda Muresan. Don't go far off: An empirical study on neural poetry translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7253–7265, 2021.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 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*, pages 6848–6863, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.460. URL https://aclanthology.org/2022.emnlp-main.460.

Tuhin Chakrabarty, Vishakh Padmakumar, He He, and Nanyun Peng. Creative natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 34–40, 2023.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Yanran Chen, Hannes Gröner, Sina Zarrieß, and Steffen Eger. Evaluating diversity in automatic poetry generation. *arXiv preprint arXiv:2406.15267*, 2024.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. Adaptersoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, 2023.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy T Rogers. Simulating opinion dynamics with networks of llm-based agents. *arXiv preprint arXiv:2311.09618*, 2023.

Yun-Shiuan Chuang, Nikunj Harlalka, Siddharth Suresh, Agam Goyal, Robert D Hawkins, Sijia Yang, Dhavan V Shah, Junjie Hu, and Timothy T Rogers. The wisdom of partisan crowds: Comparing collective intelligence in humans and llm-based agents. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2019.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations*, 2023.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, 2020.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Steffen Eger. Opinion dynamics and wisdom under out-group discrimination. *Mathematical Social Sciences*, 80: 97–107, 2016.

Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks generating "art" by learning about styles and deviating from style norms. In *8th International Conference on Computational Creativity, ICCC 2017*. Georgia Institute of Technology, 2017.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. Emosen: Generating sentiment and emotion controlled responses in a multimodal dialogue system. *IEEE Transactions on Affective Computing*, 13(3): 1555–1566, 2022. doi: 10.1109/TAFFC.2020.3015491.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2022.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. S^3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*, 2023.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Anna Gautier, Alex Stephens, Bruno Lacerda, Nick Hawes, and Michael Wooldridge. Negotiated path planning for non-cooperative multi-robot systems. 2022.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system. In Mohit Bansal and Heng Ji, editors, *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://aclanthology.org/P17-4008`.

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 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*, pages 524–533, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `https://aclanthology.org/D10-1051`.

Sil Hamilton. Blind judgement: Agent-based supreme court modelling with gpt. In *The AAAI-23 Workshop on Creative AI Across Modalities*, 2023.

Will Hipson and Saif M. Mohammad. 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*, pages 1578–1589, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.196`.

Christopher Janaway. *Schopenhauer: A very short introduction*. OUP Oxford, 2002.

Peter Jarvis. *Towards a comprehensive theory of human learning*. Routledge, 2012.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, 2023.

Long Jiang and Ming Zhou. Generating chinese couplets using a statistical mt approach. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 377–384, 2008.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *The Twelfth International Conference on Learning Representations*, 2023.

Nils Köbis and Luca D Mossink. Artificial intelligence versus maya angelou: Experimental evidence that people cannot differentiate ai-generated from human-written poetry. *Computers in human behavior*, 114:106553, 2021.

Maria Kuteeva and Marta Andersson. Diversity and standards in writing for publication in the age of ai—between a rock and a hard place. *Applied Linguistics*, page amae025, 2024.

Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 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)*, pages 1948–1958, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1181. URL `https://aclanthology.org/P18-1181`.

RAY LC. Imitations of immortality: Learning from human imitative examples in transformer poetry generation. In *10th International Conference on Digital and Interactive Arts*, ARTECH 2021, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450384209. doi: 10.1145/3483529.3483537. URL `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. Interacting with non-cooperative user: A new paradigm for proactive dialogue policy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 212–222, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3532001. URL `https://doi.org/10.1145/3477495.3532001`.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1361–1370, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. doi: 10.1145/1753326.1753532. URL `https://doi.org/10.1145/1753326.1753532`.

Chao Li, Xing Su, Chao Fan, Haoying Han, Cong Xue, and Chunmo Zheng. Quantifying the impact of large language models on collective opinion dynamics. *arXiv preprint arXiv:2308.03313*, 2023.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36, 2024.

Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. Gpt-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151*, 2019.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. The adapter-bot: All-in-one controllable conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16081–16083, 2021.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *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)*, pages 6691–6706, 2021.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023.

Xiaoding Lu, Adian Liusie, Vyas Raina, Yuwen Zhang, and William Beauchamp. Blending is all you need: Cheaper, better alternative to trillion-parameters llm. *arXiv preprint arXiv:2401.02994*, 2024.

Jingkun Ma, Runzhe Zhan, and Derek F. Wong. 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*, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-demo.8. URL `https://aclanthology.org/2023.eacl-demo.8`.

Ridwan Mahbub, Ifrad Khan, Samiha Anuva, Md Shihab Shahriar, Md Tahmid Rahman Laskar, and Sabbir Ahmed. Unveiling the essence of poetry: Introducing a comprehensive dataset and benchmark for poem summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14878–14886, 2023.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 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, 2023. doi: 10.1162/tacl_a_00567. URL `https://aclanthology.org/2023.tacl-1.38`.

Hugo Gonçalo Oliveira. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21, 2012.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, 2022.

Nils Reimers and Iryna Gurevych. 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)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410`.

Leonardo FR Ribeiro, Yue Zhang, and Iryna Gurevych. Structural adapters in pretrained language models for amr-to-text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4269–4282, 2021.

Yu-Ping Ruan and Zhenhua Ling. Emotion-regularized conditional variational autoencoder for emotional response generation. *IEEE Transactions on Affective Computing*, 2021.

Piotr Sawicki, Marek Grzes, Fabricio Goes, Dan Brown, Max Peeperkorn, and Aisha Khatun. Bits of grass: Does gpt already know how to write like whitman? In *Proceedings of the 14th International Conference for Computational Creativity*, 2023a.

Piotr Sawicki, Marek Grzes, Luis Fabricio Góes, Dan Brown, Max Peeperkorn, Aisha Khatun, and Simona Paraskevopoulou. On the power of special-purpose gpt models to create and evaluate new poetry in old styles. 2023b.

Yizhan Shao, Tong Shao, Minghao Wang, Peng Wang, and Jie Gao. A sentiment and style controllable approach for chinese poetry generation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4784–4788, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3481964. URL `https://doi.org/10.1145/3459637.3481964`.

Yizhan Shao, Tong Shao, Minghao Wang, Peng Wang, and Jie Gao. A sentiment and style controllable approach for chinese poetry generation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4784–4788, 2021b.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2016.

Lei Shen, Xiaoyu Guo, and Meng Chen. Compose like humans: Jointly improving the coherence and novelty for modern chinese poetry generation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9206888.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. Towards controllable biases in language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3239–3254, 2020.

Guodong Shi, Claudio Altafini, and John S Baras. Dynamics over signed networks. *SIAM Review*, 61(2):229–257, 2019.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=V88BafmH9Pj`.

Guy Tevet and Jonathan Berant. 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*, pages 326–346, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.25. URL `https://aclanthology.org/2021.eacl-main.25`.

Huishuang Tian, Kexin Yang, Dayiheng Liu, and Jiancheng Lv. Anchibert: A pre-trained model for ancient chinese language understanding and generation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

Yufei Tian and Nanyun Peng. Zero-shot sonnet generation with discourse-level planning and aesthetics features. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3587–3597, 2022.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35: 38274–38290, 2022.

David Uthus, Maria Voitovich, and R.J. Mical. 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*, pages 18–26, Hybrid: Seattle, Washington + Online, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-industry.3. URL `https://aclanthology.org/2022.naacl-industry.3`.

Tim Van de Cruys. 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*, pages 2471–2480, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.223. URL `https://aclanthology.org/2020.acl-main.223`.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Hongyi Wang, Felipe Maia Polo, Yuekai Sun, Souvik Kundu, Eric Xing, and Mikhail Yurochkin. Fusing models with complementary expertise. In *Annual Conference on Neural Information Processing Systems*, 2023b.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 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)*, pages 166–177, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1015. URL `https://aclanthology.org/N19-1015`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2022a.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, 2022b.

Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 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*, pages 1051–1060, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `https://aclanthology.org/C16-1100`.

Roosa Wingström, Johanna Hautala, and Riina Lundman. Redefining creativity in the era of ai? perspectives of computer scientists and new media artists. *Creativity Research Journal*, pages 1–17, 2023.

Jörg Wöckener, Thomas Haider, Tristan Miller, The-Khang Nguyen, Thanh Tung Linh Nguyen, Minh Vu Pham, Jonas Belouadi, and Steffen Eger. 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*, pages 57–66, Punta Cana, Dominican Republic (online), November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.latechclfl-1.7. URL `https://aclanthology.org/2021.latechclfl-1.7`.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Rui Yan. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *International Joint Conference on Artificial Intelligence*, 2016. URL `https://api.semanticscholar.org/CorpusID:14079825`.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li, and Maosong Sun. Mixpoet: Diverse poetry generation via learning controllable mixed latent space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9450–9457, 2020.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37, 2023.

Ran Zhang, Jihed Ouni, and Steffen Eger. Cross-lingual cross-temporal summarization: Dataset, models, evaluation. *Computational Linguistics*, pages 1–44, 2024.

Xingxing Zhang and Mirella Lapata. 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)*, pages 670–680, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1074. URL `https://aclanthology.org/D14-1074`.

Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. 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*, pages 1022–1040, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.65. URL `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. 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*, pages 25–30, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3005. URL `https://aclanthology.org/P19-3005`.

Andrew Zhu, Lara Martin, Andrew Head, and Chris Callison-Burch. Calypso: Llms as dungeon master's assistants. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, pages 380–390, 2023.

Mingchen Zhuge, Haozhe Liu, Francesco Faccio, Dylan R Ashley, Róbert Csordás, Anand Gopalakrishnan, Abdullah Hamdi, Hasan Abed Al Kader Hammoud, Vincent Herrmann, Kazuki Irie, et al. Mindstorms in natural language-based societies of mind. *arXiv preprint arXiv:2305.17066*, 2023.

8 Appendix

8.1 Prompt template for PROMPTING-BASED agents

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 12: Prompt template for chain-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:

Table 13: Prompt template for joint-prompting strategy

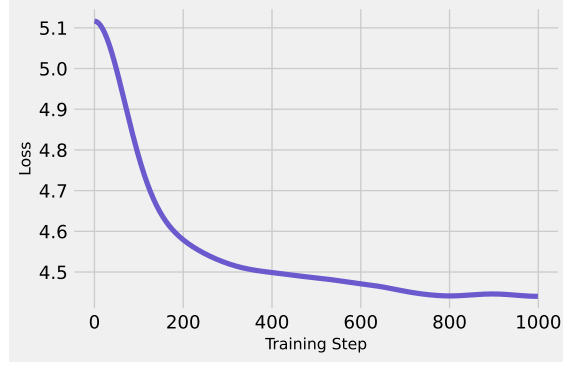Table 12 shows the prompt templates for chain-prompting. Table 13 shows the prompt templates for joint-prompting.

Figure 6: Loss of QUATRAIN data during pretraining.

8.2 Hyperparameters

8.2.1 Loss cure during pretraining

We present the loss curve of QUATRAIN data during pretraining in Figure 6.

8.2.2 Decoding parameter