Simulating Social Media Platforms with Large Language Models: Latent MBTI Personality Projections in LLMs

Moein Taherinezhad
Politecnico di Milano
moein.taherinezhad@mail.polimi.it

Mohammad Javad Zandiyeh
Politecnico di Milano
mohammadjavad.zandiyeh@mail.polimi.it

Het Hargovind Ashar
Politecnico di Milano
hethargovind.ashar@mail.polimi.it

Hamidreza Saffari
Politecnico di Milano
hamidreza.saffari@mail.polimi.it

Abstract

This study proposes a machine learning framework for inferring Myers–Briggs Type Indicator (MBTI) personality types from natural language chat data. Using over 8,000 labeled user posts from Kaggle, each MBTI type is decomposed into four binary classification tasks along the Extraversion/Introversion, Sensing/Intuition, Thinking/Feeling, and Judging/Perceiving axes. Sentence-BERT embeddings are used to represent text, and independent logistic regression models are trained for each trait. The system is applied to real-world conversational JSON data, enabling personality prediction for agents across multiple chat sessions. Results show a bias toward introverted and intuitive types, reflecting dataset imbalance. The model provides a scalable, interpretable approach for conversational personality analysis, with potential applications in behavioral research and human–AI interaction.

1 Introduction

LLMs often reflect biases present in their training data, including social and cultural stereotypes, which can result in unfair, discriminatory, or unreliable outputs. This study focuses on one such bias by examining the innate personality projections of LLMs. The goal is to determine whether LLMs exhibit consistent latent behavioral patterns when placed in controlled situational contexts shaped by predefined basic personality traits. To explore this, a simulated social media network is constructed, where LLM agents engage in conversations within a specific topical domain. Each agent's participation is then analyzed to infer its personality profile, using the MBTI framework to classify the generated texts across different scenarios.

Two distinct approaches are used to simulate the social media network. The first, **Manual Network Creation**, involves explicitly defining the network structure while delegating only the conversation generation to LLM agents, allowing full control over the foundational parameters. The second, **Automatic Network Creation**, relies on a single LLM agent to generate both the network and the conversations, enabling rapid and scalable simulation while placing emphasis on prompt design to guide behavior.

2 Code

Our Github repository can be accessed from here.

3 Related Work

LLM-powered social simulation systems. Several recent studies have explored the use of large language models to simulate social behavior in networked environments. The S3 framework (Social-network Simulation System) introduces a structured simulation system where LLM agents are embedded in a social network and interact with one another to mimic realistic social processes (Gao et al., 2025). It emphasizes the flexibility and scalability of using LLMs to represent individuals in various social settings. Similarly, the Y Social project builds an LLM-powered digital twin of a social media platform, allowing researchers to test different interventions and observe emergent behaviors at scale (Rossetti et al., 2024). It highlights how LLM agents can represent diverse users and support policy experimentation in silico. Another study, Agent-Based Modelling Meets Generative AI in Social Network Simulations, investigates how generative AI can enhance traditional agent-based modeling by integrating LLMs to simulate complex human-like interactions and decisions across evolving social networks (Ferraro et al., 2025).

Evaluation of news feed algorithms. Another line of work investigates how LLMs can simulate user behavior to evaluate algorithmic outcomes. One such study uses LLM agents in a social media environment to test how different news feed algorithms affect content exposure and opinion dynamics (Törnberg et al., 2023). By assigning distinct personas to agents and simulating their content preferences and interactions, the authors demonstrate that LLM-based simulations can help assess algorithmic fairness and potential echo chamber effects. Together, these works demonstrate the growing value of LLM-based simulations for studying social systems, media dynamics, and algorithmic impact.

4 Manual Network Creation

In the Manual Network Creation approach, social media conversations are created by first defining the personalities of each agent and then manually setting up how they interact in a fixed network. Unlike the Automatic Network Creation method, where an LLM generates both the network and the conversations on its own, this approach gives more control over how the agents behave and who they talk to.

4.1 Persona Modeling

Each agent in the network is given a simple persona based on a few key demographic and social traits. These include their name, gender, age group, economic status (such as Working Class or Upper-Middle Class), and current job. A diverse set of personas is created in advance to represent different backgrounds and points of view. This helps make the social interactions more realistic and varied.

4.2 Network Structure and Group Formation

Agents are represented as nodes in an undirected graph using NetworkX. Groups are created manually by choosing sets of agents to take part in conversations. The `create_group` method is used to form each group. It randomly connects the selected agents and sets up `Group` object that manages how they interact.

4.3 Conversation Generation

Group conversations are simulated using a turn-based system. In each round, every agent sends one message, and this continues for a set number of rounds. Messages are created using a selected LLM model specifically, `DeepSeek-V3-0324` accessed through the OpenRouter API. To handle API rate limits, several OpenAI clients are used, and the system rotates between them to keep the conversation going without stopping.

The conversation is shaped by well-designed prompts that set the topic (such as work–life balance, reflections on failure, or content preferences). A memory window controls how much of the past chat each agent can access either the full history or just a limited number of recent messages (like the last 10).

Each message is influenced by both the agent's persona and the recent conversation. The conversations are saved in a structured JSON format that includes who sent the message, who received it, and the message itself. This format makes it easier to analyse and use the data for tasks like personality classification.

4.4 Experiment Execution Loop

The experiment execution loop systematically runs communication simulations across various configurations of agent groups, initial prompts, and memory constraints. For each configuration, the system generates and stores the corresponding chat history to facilitate further analysis.

4.5 Process Overview

The loop iterates over all possible combinations of the following parameters:

- **Group:** A subset of agent indices representing the participants in the conversation.
- **Prompt Initialization (`prompt_start`):** A dictionary containing a *title* and a *prompt* used to initialize the dialogue.
- **Memory Length (`len_chat_memory`):** The number of preceding messages included as conversational context. This may be a fixed value or `None` to indicate full memory.

4.6 Workflow per Experiment

Each experiment follows the steps outlined below:

1. **Naming:** A unique identifier for the experiment is generated based on the selected group, prompt title, and memory length.
2. **Group Creation:** The function `social_network.create_group()` is invoked to instantiate a group of agents corresponding to the current configuration.
3. **Network Visualization:** The agent connections within the group are visualized by calling `social_network.draw_network()`.

4. **Communication Simulation:** The function `group_obj.communicate()` initiates the dialogue among agents using the defined prompt and memory context.

5. **Result Saving:** The complete chat history is saved to a `.json` file named according to the unique experiment identifier.

For each experimental run, the system visualizes the agent connections, simulates multi-round group communication, and saves the resulting dialogue history with a uniquely identifiable filename. This enables reproducible analysis and comparison across configurations.

5 Automatic Conversation Creation

In this approach, we aim to extend the methodology proposed by Chang et al. (2025). While the original study focused on generating social media networks in a single step by prompting an LLM to generate the connections for a given set of personas, our goal is to enhance this by simulating dynamic conversations between individuals. This extension incorporates variable rate limits, allowing us to test a range of LLMs without requiring multiple prompts for each conversation. As a result, this approach facilitates the evaluation of a broader set of LLMs in a more efficient and scalable manner.

5.1 Persona Modeling

To create realistic social media interactions, we model personas using a set of key demographic and socio-economic attributes. Each persona is defined by their basic demographics (name, age, and gender), economic status (categorized into *Upper Class*, *Upper Middle Class*, *Middle Class*, and *Working Class*), and their current occupation. This simplified but effective persona model allows us to capture essential characteristics that influence social media interactions while maintaining computational efficiency. Each persona's attributes are carefully chosen to create diverse perspectives in conversations, ensuring realistic and meaningful interactions.

5.2 Conversation Generation

The conversation generation process is controlled by several parameters that shape the nature and flow of interactions. The topic of conversation is explicitly defined, ranging from economic issues like Universal Basic Income to social concerns like *Climate Change*. The number of participants is variable, allowing for both one-on-one discussions and group conversations. Each message is limited to 350 characters to maintain realistic social media post lengths, and the number of messages per person is randomly determined between 10 and 20 to simulate natural conversation flow. The system generates conversations in a structured JSON format, including persona identification, message content, and timestamps for each message.

5.3 Models

In this experiment, we used Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen2.5-7B-Instruct (Qwen et al., 2025) as our models. The Hugging Face Inference API was utilized to access these models. To ensure consistent results, we set the temperature to zero for all evaluations. Throughout our experiments, we refer to these models by their shortened names: Mistral, Qwen.

5.4 Prompt

Table 2 shows the prompt used in this experiment. We employed a zero-shot approach, meaning the prompt did not include any examples. The prompt structure includes detailed persona descriptions with their demographic and socio-economic attributes, conversation parameters and constraints, and specific output format requirements. This structured approach ensures that the generated conversations maintain consistency with the defined personas and conversation parameters.

6 MBTI Personality Classification Model

The goal of this model is to predict the Myers–Briggs Type Indicator (MBTI) personality type of a person based on their written chat messages. MBTI classification is framed as a multi-label classification task, where each of the four personality dimensions is predicted independently.

6.1 Collab Link

You can access the Colab notebook used for training and prediction at: `https://colab.research.google.com/drive/1qzcEE0G0U84U4xXQ3GrnuI67iT0EM-LO?usp=sharing`

6.2 Dataset

The model is trained on the widely used MBTI dataset (Community, 2017), which contains 8,675 rows. Each row includes a user's MBTI personality

Parameter	Type	Description
name	string	The persona’s full name
gender	string	Gender identity (e.g., "Male", "Female")
age	integer	Age in years
economic_status	string	Socio-economic class (e.g., "Upper Class", "Working Class")
occupation	string	Current professional role

Table 1: Persona Class Parameters in Automatic Conversation Creation Model

Generate a social media conversation between these personas:

{persona_descriptions}

Conversation parameters:

- Topic: {config.topic}

- Maximum messages: {config.max_messages_per_person}

- Maximum message length: {config.max_message_length} characters

- When Creating the messages, incorporate the personas characteristics so that the messages of each reflect their personas.

Return the conversation in the following JSON format:

{

"conversation": [{

"persona": "name",

"message": "message content",

"timestamp": "ISO format timestamp"

}] }

Table 2: Prompt instruction for the Automatic Conversation Generation.

type (e.g., INFP, ENTP) and a set of social media posts concatenated by delimiters.

6.3 Preprocessing

Each user’s posts are split using the ||| delimiter and concatenated into a single string, truncated to the first 1000 characters. The MBTI type is then decomposed into four binary trait labels:

- **E/I (Extraversion / Introversion)**
- **S/N (Sensing / Intuition)**
- **T/F (Thinking / Feeling)**
- **J/P (Judging / Perceiving)**

These labels are used as targets for four independent binary classifiers.

6.4 Feature Representation

Text inputs are transformed into dense vector representations using the `all-MiniLM-L6-v2` model from the `sentence-transformers` (Reimers and Gurevych, 2019) library. This model is a lightweight variant of BERT that provides meaningful sentence embeddings and balances speed with performance.

6.5 Model Architecture

Four binary classifiers are trained independently, one for each MBTI trait dimension (Wolf et al., 2020). Each classifier is implemented using `LogisticRegression` from `scikit-learn` (Pedregosa et al., 2011). To address the class imbalance in the dataset, the classifiers are trained using the `class_weight='balanced'` setting.

6.6 Training Procedure

For each trait axis, the dataset is split into training and test sets using an 80/20 ratio. Sentence embeddings are precomputed to optimize training time. The logistic regression models are trained with a maximum of 1000 iterations.

6.7 Inference

During inference, a new chat transcript is tokenized and embedded using the same sentence transformer. Each trait classifier outputs a binary prediction, which is then assembled into a complete MBTI type. For example, the predicted outputs `[I, N, T, J]` would yield the type `INTJ`.

6.8 Limitations and Bias

Due to the skewed distribution in the training dataset particularly toward Introverted and Intuitive types the model tends to overpredict types that begin with `IN`. While balancing techniques are employed, this limitation highlights the importance of using a more representative dataset or applying additional regularization and threshold calibration during prediction (Ando and Zhang, 2005)

Axis	Trait 1	Trait 2	Skew Toward
E/I	**E:** 23.0%	**I:** 77.0%	Introversion (I)
S/N	**S:** 13.8%	**N:** 86.2%	Intuition (N)
T/F	**T:** 45.9%	**F:** 54.1%	Balanced
J/P	**J:** 39.6%	**P:** 60.4%	Perceiving (P)

7 Results

This section presents the observed evolution of MBTI (Myers-Briggs Type Indicator) types assigned to individual agents within social network simulations, across different models and network creation methodologies. The simulations involved conversations between five agents: Frank, Karen, Kayla Agent, Leo, and Morgan.

7.1 Manual Network Creation: DeepSeek-V3-0324 Model

The first set of 30 experiments utilized the DeepSeek-V3-0324 model under the "Manual Network Creation" approach, where the network structure was explicitly defined, and the LLM agents were solely responsible for conversation generation (1). In this setup, Frank primarily exhibited INTP and INTJ types across various content, reflection, and work-life balance files. Karen displayed a significant presence of INTP and INTJ, with occasional INFJ and INFP. Kayla Agent was predominantly INFP, with two instances of ENFP. Leo consistently appeared as INFP and INTP, with some INFJ assignments. Morgan showed a mix of INTJ, INTP, and INFJ. Overall, the DeepSeek-V3-0324 model, when operating within a pre-defined network, demonstrated a relatively diverse distribution of MBTI types across agents, with INTP, INTJ, and INFP being common.

7.2 Automatic Network Creation: Qwen Model

The "Automatic Network Creation" approach involved a single LLM agent generating both the network structure and the conversations. Ten experiments were conducted for each model in this category. The results for the Qwen model are presented in 3. For the Qwen model, Frank was predominantly assigned as INTP, with one instance of INFJ and one ENFP. Karen primarily exhibited INTP and INTJ types, with single instances of INFJ and INFP. Kayla Agent largely remained INTP, with one ENFP and one INFJ assignment, but was absent in several files. Leo appeared as INFP and was largely absent in many files. Morgan was predominantly INTP and INTJ, with one INFP. In contrast to the manual network creation, the Qwen model demonstrated a notable shift towards INTP and INTJ types across most agents when present. This suggests that when the LLM is also responsible for defining the network, it tends to favor these analytical and logical personality types in the generated conversational dynamics.

7.3 Automatic Network Creation: Mistral Model

The results for the Mistral model, also under the "Automatic Network Creation" approach, are detailed in 2, encompassing 10 experiments. In the Mistral simulations, Frank primarily exhibited INFP, INFJ, and INTP. Karen displayed a mix of INFJ, INTJ, and INTP. Kayla Agent predominantly showed INFP and INFJ assignments, but was frequently absent. Leo appeared as INTP, INFP, and INFJ, but was also often absent. Morgan was consistently assigned as INFJ, INFP, and INTJ. The Mistral model, while also operating in an automatic network creation mode, produced a broader, yet still patterned, distribution of MBTI types compared to the Qwen model. While INFJ, INFP, INTJ, and INTP were common across agents, there was less clear-cut dominance of one or two types for specific agents compared to the Qwen model.

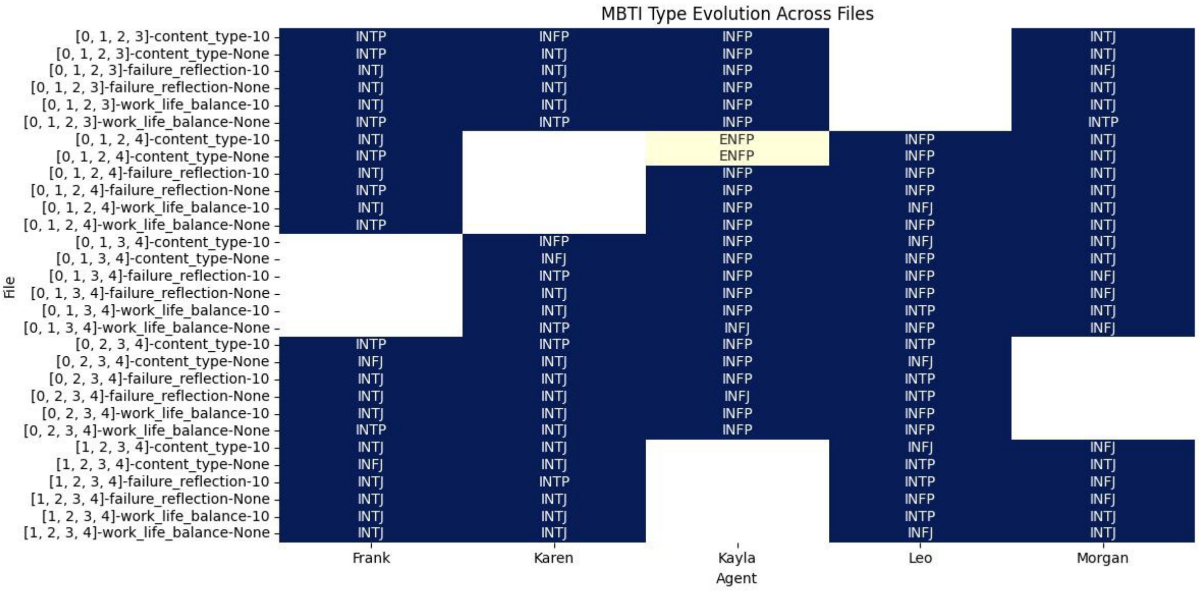

Figure 1: The MBTI analysis for agents across different conversation settings for DeepSeek.

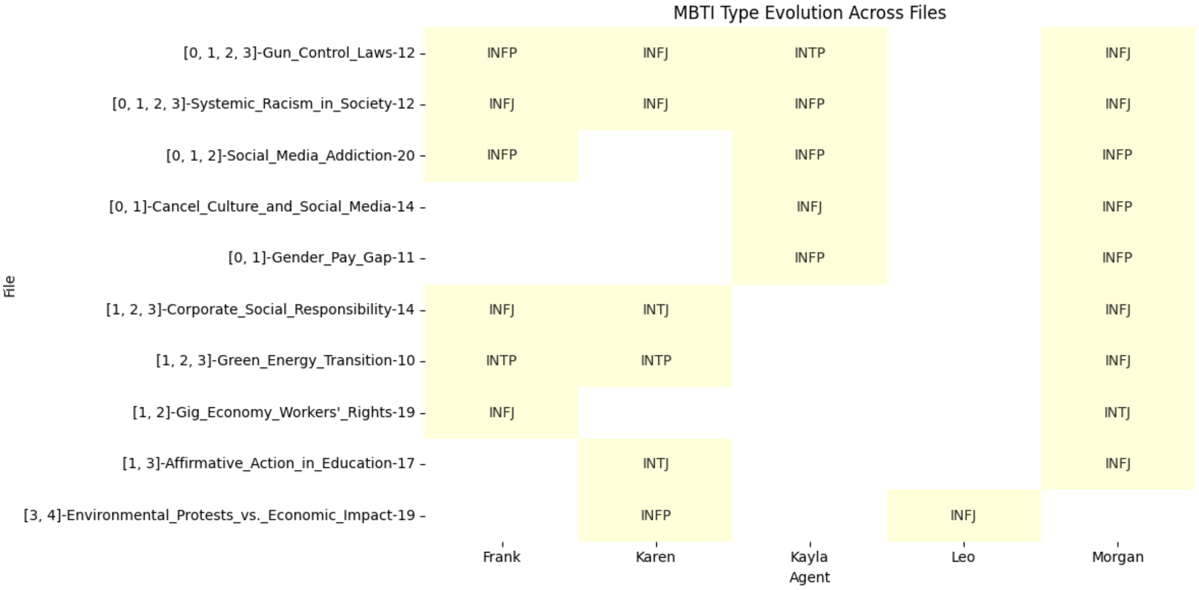

Figure 2: The MBTI analysis for agents across different conversation settings for Mistral.

7.4 Discussion of Trends

Analysis across the three experimental setups reveals several key trends regarding MBTI type evolution in simulated social networks.

Firstly, each LLM model exhibited distinct propensities in assigning MBTI types. DeepSeek-V3-0324 demonstrated a varied distribution with a strong presence of INTP, INTJ, and INFP. Qwen showed a strong inclination towards INTP and INTJ, particularly for consistently present agents, while Mistral provided a more balanced distribution across INFJ, INFP, INTJ, and INTP. Secondly, the "Automatic Network Creation" approach, particularly evident with the Qwen model, appeared to narrow the diversity of assigned MBTI types, concentrating on a smaller set. While Mistral also operated in this mode, it maintained a slightly wider, though consistent, type distribution for its agents. Thirdly, despite variations across models and approaches, certain agents demonstrated tendencies towards specific MBTI types or clusters of types within their respective simulations. Lastly, across all experiments, INTP, INTJ, INFP, and INFJ were the most frequently assigned MBTI types. This suggests that the LLMs, irrespective of the under-

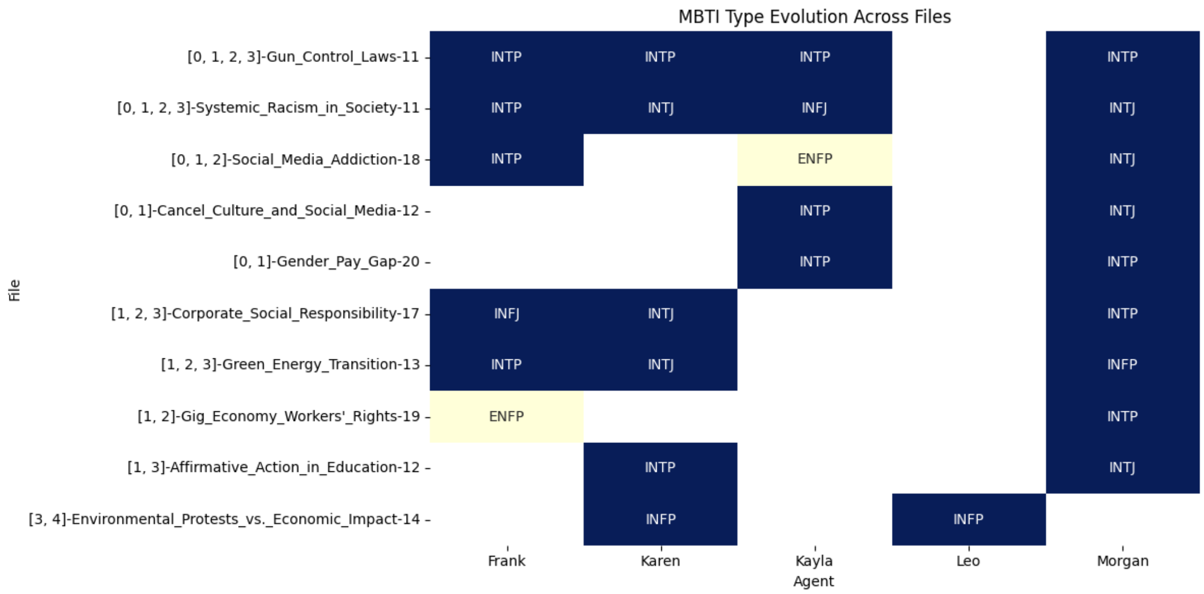

Figure 3: The MBTI analysis for agents across different conversation settings for Qwen.

lying prompt design, tended to generate conversations and agent behaviors that align with these introverted and intuitive personality profiles. The ENFP type was assigned less frequently and only for Frank and Kayla in certain scenarios.

These findings underscore the significant impact of both the choice of LLM and the network creation methodology on the simulated personalities of agents in social network simulations. The observed patterns highlight the critical role of model capabilities and prompt engineering in shaping the emergent characteristics and dynamics of the simulated interactions.

8 Limitations

Despite providing valuable insights into LLM agent personalities in social network simulations, this study faces several limitations. Firstly, relying on the MBTI framework for personality assessment is inherently problematic due to its established scientific limitations in reliability and validity for human personalities. Applying it to LLM agents further complicates interpretation, as the assigned types are inferences from generated text rather than reflections of stable, intrinsic traits, potentially oversimplifying the agents' complex behaviors.

Secondly, each network creation methodology presents distinct constraints. Manual network creation offers precise control but is not scalable for large or complex simulations, and risks introducing human biases that might inadvertently constrain emergent social dynamics. Conversely, automatic network creation, while scalable, suffers from reduced transparency and control, making it difficult to understand the rationale behind generated networks or conversational patterns. This approach is also highly sensitive to prompt design, which can hinder reproducibility and the consistent long-term evolution of agent personalities.

Finally, the inherent limitations of LLM technology and the scope of the current experiments warrant consideration. LLMs can perpetuate biases from their training data and fundamentally lack true understanding, consciousness, or real-world experience, which restricts their ability to fully replicate the nuances of human communication. Furthermore, the simulations were conducted with a relatively small number of agents and specific conversational topics. Future work should explore larger-scale simulations and a broader array of social contexts to enhance the generalizability of these findings.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Serina Chang, Alicja Chaszczewicz, Emma Wang, Maya Josifovska, Emma Pierson, and Jure Leskovec. 2025. Llms generate structurally realistic social networks but overestimate political homophily. *Preprint*, arXiv:2408.16629.

Kaggle Community. 2017. Mbti personality type dataset. https://www.kaggle.com/datasnaek/mbti-type.

Antonino Ferraro, Antonio Galli, Valerio La Gatta, Marco Postiglione, Gian Marco Orlando, Diego Russo, Giuseppe Riccio, Antonio Romano, and Vincenzo Moscato. 2025. Agent-based modelling meets generative ai in social network simulations. In *Social Networks Analysis and Mining*, pages 155–170, Cham. Springer Nature Switzerland.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2025. S^3: Social-network simulation system with large language model-empowered agents. *Preprint*, arXiv:2307.14984.

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992.

Giulio Rossetti, Massimo Stella, Rémy Cazabet, Katherine Abramski, Erica Cau, Salvatore Citraro, Andrea Failla, Riccardo Improta, Virginia Morini, and Valentina Pansanella. 2024. Y social: an llm-powered social media digital twin. *Preprint*, arXiv:2408.00818.

Petter Törnberg, Diliara Valeeva, Justus Uitermark, and Christopher Bail. 2023. Simulating social media using large language models to evaluate alternative news feed algorithms. *Preprint*, arXiv:2310.05984.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.