# Generating and Validating Agent and Environment Code for Simulating Realistic Personality Profiles with Large Language Models

**Nathan Cloos**
MIT
nacloos@mit.edu

**M Ganesh Kumar**
Harvard
mganeshkumar@seas.harvard.edu

**Adam Manoogian**
Monash University
adam.manoogian@monash.edu

**Christopher J. Cueva***
MIT
ccueva@gmail.com

**Shawn A. Rhoads***
Mount Sinai
shawn.rhoads@mssm.edu

## Abstract

Scaling up the creation of realistic agents and environments poses a significant challenge in artificial intelligence and science more broadly. Environments are traditionally handcrafted, restricting their scalability and diversity. Recent advancements in large language models (LLMs) offer a promising approach to automating environment design. In behavioral and cognitive science, a key goal is to characterize traits that predict behavior, typically through carefully designed cognitive tasks. However, this approach also faces significant scaling challenges due to the extensive human validation required. To address these limitations, we leverage LLMs to generate not only the code for environmental affordances and the agent policy but also the code that ensures their validity. Specifically, we assign agents distinct personality profiles based on data from large-scale psychological studies that have identified consistent and reliable personality traits. The LLM-generated validation code then evaluates how accurately these traits are reflected in the agents' simulated behaviors using the widely recognized HEXACO personality inventory. Our results demonstrate that the LLM-generated pipeline can simulate a diverse range of personality profiles. Additionally, we find that specific components, such as the type of contextual information in the LLM prompts, significantly impair the recoverability of these personality profiles. We believe our approach offers a systematic and scalable method for simulating realistic personality profiles by validating environments and agents generated by LLMs.

## 1 Introduction

The development of Large Language Models (LLMs) has significantly advanced the field of artificial intelligence, particularly towards creating more generally capable agents [Ichter et al., 2022, Huang et al., 2022, Park et al., 2023, Song et al., 2023, Wang et al., 2024]. These agents leverage the vast knowledge embedded in pretrained language models to make decisions within various environments. While there has been substantial progress in improving agent behavior and decision-making capabilities, the environments in which these agents operate remain predominantly handcrafted [Puig et al., 2018, Shridhar et al., 2021, Lin et al., 2023], limiting both the scalability and diversity of agent-environment interactions. In this work, we explore the potential of LLMs to automatically generate environments.

*Equal senior authors.

NeurIPS 2024 Workshop on Behavioral Machine Learning.

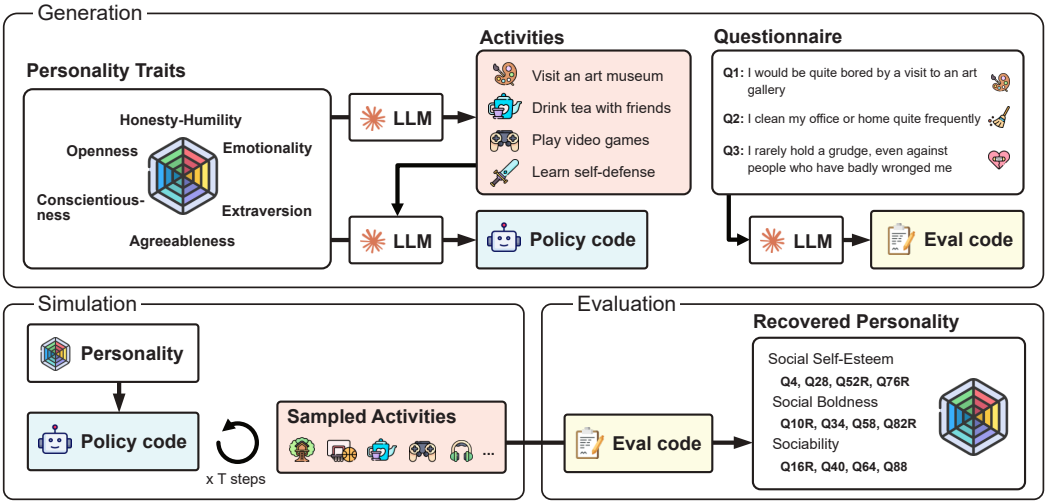


Figure 1: **Pipeline for generating agents aligned with various personality traits.** The pipeline consists of three stages: (1) Generation, where LLMs create an activity list, policy code for the agents, and evaluation code to assess behaviors; (2) Simulation, where agents choose activities based on their personality traits; (3) Evaluation, where the consistency between the agent's input personality and its recovered personality is assessed.

In behavioral and cognitive science, a key goal is to characterize phenotypes or traits that predict behavior, and computational psychiatry similarly uses cognitive tasks to map psychiatric phenotypes [Patzelt et al., 2018]. This approach also faces significant scaling challenges due to the vast task feature space (i.e., many different types of environments) and the need for validation. LLM-generated code offers a scalable solution to circumvent these limitations by generating both environments and trait profiles that predispose agents to behave in specific ways. Leveraging data from large-scale psychological studies that have identified consistent and reliable personality traits [Lee and Ashton, 2018], agents can be assigned specific personality profiles (i.e., sets of personality traits). This approach could open new avenues for understanding how different traits impact agent behavior in complex, more ecologically valid, and diverse environments.

Existing research has demonstrated the potential of LLMs to automate environment generation by using them to write code for 3D environments [Faldor et al., 2024]. Our approach differs in that we emphasize the importance of grounding and validating these generated environments. The pipeline we propose not only utilizes LLMs to generate both environments and agents but also generates mechanisms for evaluating the environments.

## 2 Methods

**Grounding personality traits.** We base our assessment using the HEXACO personality inventory, a well-established psychological model that measures six major factors of personality: Honesty-Humility, Emotionality, Extroversion, Agreeableness, Conscientiousness, and Openness to Experience [Lee and Ashton, 2018]. These factors were derived using principal components analysis, which identified the underlying structure of personality traits by analyzing large empirical datasets [Lee and Ashton, 2018]. The inventory includes 100 questions (detailed in Appendix A) designed to evaluate factors of the HEXACO model. Participants provide ratings for each question on a scale of 1 to 5, ranging from strong disagreement to strong agreement. The scores for each factor are computed by averaging the ratings corresponding to each factor.

The proposed pipeline consists of three stages (Figure 1): Generation, Simulation, and Evaluation. Of these, only the Generation phase involves the use of large language models (LLMs), while the Simulation and Evaluation phases rely on code produced during the initial stage. We use Claude 3.5 Sonnet as our LLM. The following sections provide a detailed description of each stage.

**Generation.** We employ three LLMs instances to generate the following components (detailed prompts in Appendix B):

- **Activity list.** We prompt the LLM to generate a diverse list of 100 concrete and common activities reflecting the HEXACO personality traits (Appendix A).
- **Policy code.** Given the generated activity list and HEXACO personality traits, we prompt another LLM to generate code for a policy. This policy is a function that takes a dictionary representing an agent's personality traits with values from 0 to 100 as input and returns an activity the agent should perform. We specify in the prompt to first explain reasoning before writing code. We also decompose the policy into separate functions for each activity, which we find results in more detailed reasoning than when using a single function.
- **Evaluation code.** To close the loop and evaluate how well personality traits can be recovered from activities, we prompt a third LLM to generate code to answer the HEXACO personality questions. The list of questions and the generated list of activities are given in context. The evaluation function takes a sequence of activities taken by an agent as input and outputs scores for each question of the HEXACO personality inventory. Similarly to the policy code, we find that decomposing into separate functions for each question helps make the reasoning more detailed.

**Simulation.** We simulate agents assigned with various personality profiles. The process begins by sampling personality traits based on human data, ensuring that the distributions of traits align with real-world statistics (see Appendix A for details). The policy code generated in the previous stage is then used to simulate sequences of activities for each agent over a fixed number of timesteps, where one activity is taken at each timestep.

**Evaluation.** The final stage involves using the LLM-generated evaluation code to assess the activities performed by the simulated agents. The evaluation code returns a list of scores for each of the HEXACO personality inventory questions based on the observed activities. We then measure the correlation between the original assigned personality profile and the recovered personality profile generated from the agent's activities. This correlation serves as a key metric for evaluating the consistency and alignment of the entire LLM-generated pipeline.

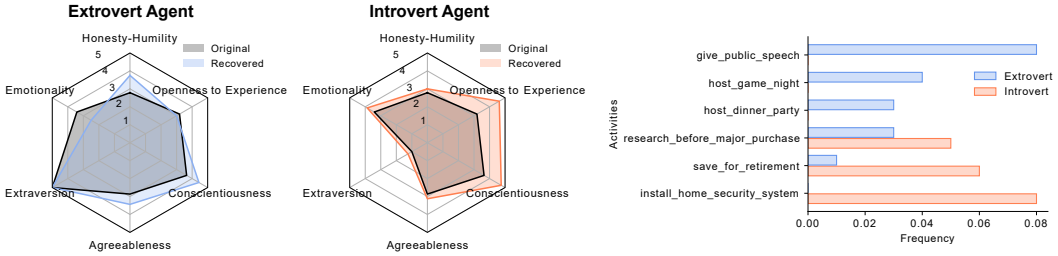


Figure 2: **LLM-generated code successfully simulates and recovers an extroverted and introverted personality profile.** (Left) The LLM-generated evaluation code recovers the personality profiles from the activities taken by the policy. (Right) Top three activities chosen by the extrovert (first three rows) and introvert agents (last three rows).

## 3 Results

**Recovering personality profiles successfully.** Consider two agents with opposing levels of extroversion: one agent has the highest possible extroversion score, while the other has the lowest (see Figure 2, left). To recover these two distinct personality profiles, the LLM-generated code must recognize and align the appropriate activities with the corresponding personality traits. For example, one sub-factor of extroversion in the HEXACO model is "sociability", measured by questions such as: "I avoid making 'small talk' with people.".

The LLM-generated evaluation code identifies the activities relevant to sociability, such as `host_dinner_party`, `organize_group_outing`, or `host_game_night`. The policy code, generated independently, must then choose these activities in response to high sociability input traits. We find that these activities indeed appear in the most frequently taken activities by the policy with the extrovert profile, but not by the policy with the introvert profile (see right of Figure 2).

**Simulating agents with diverse personality profiles.** To demonstrate the ability of the LLM-generated policy to simulate a wide range of personality profiles, we sample a variety of profiles by

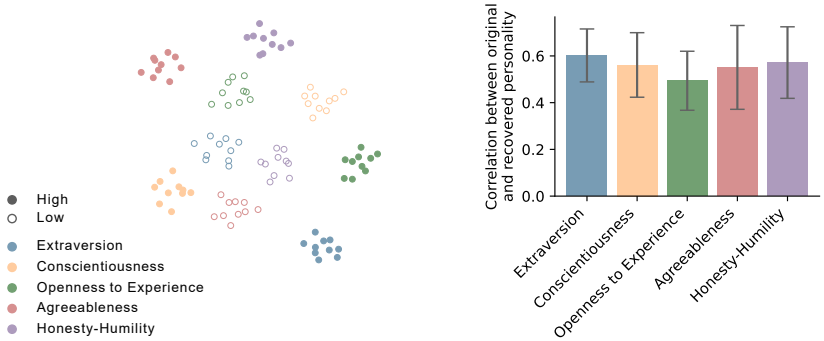


Figure 3: **Simulating diverse agents with a range of personality profiles.** (Left) t-SNE visualization of activity frequencies from agents with different personality profiles simulated for 100 timesteps. (Right) Correlation between original and recovered personality traits, demonstrating effective recoverability.

setting each HEXACO factor score to its minimum or maximum value. For each unique personality, we simulate agents over multiple runs and track the frequency of their selected activities.

Figure 3 (left) visualizes the activities of agents with varying personality profiles using t-SNE [van der Maaten and Hinton, 2008]. The clustering reveals distinct patterns of behavior, indicating that agents with different personality profiles tend to engage in unique sequences of activities.

In Figure 3 (right), we quantitatively assess personality recoverability by measuring the correlation between the input personality profile and the profile computed from the agent's activities. Strong correlations across multiple personality profiles confirm that the LLM-generated policy and evaluation code consistently aligns agent's behaviors with the original personality profiles.

**However, generating environments that consistently lead to high personality recoverability remains a challenge.** Even with identical prompts, we observe substantial variability in personality recoverability when sampling multiple LLM outputs (Figure 4). Additionally, the list of activities generated by the LLM significantly impacts recoverability. Specifically, the pipeline fails to recover personality profiles when the LLM generates activities without contextual information about the HEXACO personality traits or the corresponding HEXACO questions.

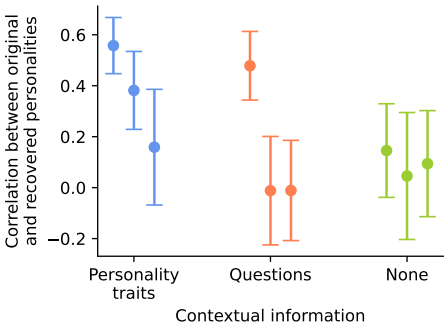


Figure 4: We sample 100 HEXACO personality profiles using statistics from empirical data (see Appendix A for details) and evaluate recoverability across different iterations of the pipeline. Specifically, we vary the information input in the LLM prompt used to generate the list of activities, giving either the list of HEXACO personality traits (blue), the list of HEXACO questions (orange), or no additional information (green). We run the LLMs 3 times for each set of prompts with a sampling temperature of 1.

## 4 Conclusions

While our proof-of-concept approach demonstrates the potential of LLMs for generating environments and simulating agents with distinct personality profiles, several limitations must be addressed in future work. First, the generated environments currently lack realistic transition dynamics between activities. This could be addressed by incorporating transition probabilities that reflect real-world dependencies between tasks. Second, the independence of each timestep in our simulations does not account for the history of activities, which is crucial for capturing more complex behavioral patterns over time. Additionally, our current evaluation relies on activities derived using information from the personality inventory. Future work will aim to recover personality profiles from agents engaging in independently-generated activities. Despite these limitations, our approach holds promise for

discovery-oriented cognitive science, potentially enabling the generation of novel research questions by simulating diverse behavioral environments. This could facilitate scalable and data-driven insights into the interplay between traits and behavior.

## References


Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. Omni-epic: Open-endedness via models of human notions of interestingness with environments programmed in code, 2024.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, 2022.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

Kibeom Lee and Michael C. Ashton. Psychometric properties of the hexaco-100. *Assessment*, 25(5): 543–556, 2018.

Zijun Lin, Haidi Azaman, M Ganesh Kumar, and Cheston Tan. Compositional learning of visually-grounded concepts using reinforcement. *arXiv preprint arXiv:2309.04504*, 2023.

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*. Association for Computing Machinery, 2023.

Emily H. Patzelt, Catherine A. Hartley, and Samuel J. Gershman. Computational phenotyping: using models to understand individual differences in personality, development, and mental illness. *Personality Neuroscience*, 1:e18, 2018. doi: 10.1017/pen.2018.14.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs, 2018. URL https://arxiv.org/abs/1806.07011.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

C. Song, B. M. Sadler, J. Wu, W. Chao, C. Washington, and Y. Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

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. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

# A HEXACO personality inventory

The HEXACO personality inventory consists in 100 questions that are designed to access 6 major dimensions of personality: Honesty-Humility, Emotionality, Extroversion, Agreeableness, Conscientiousness, Openness to Experience. Participants answer questions with a score between 1 and 5, where 1 if they strongly disagree or 5 if they strongly agree. The personality inventory is then computed with a scoring sheet[2] based on the answers to the questions. We use statistics from empirical data to sample personality profiles reflecting the distribution of human personality traits. Specifically, we sample personality profiles from a multivariate Gaussian distributions with means and standard deviations obtained from the results of 1126 participants for the self-report form[3].

Questions:

1. I would be quite bored by a visit to an art gallery.
2. I clean my office or home quite frequently.
3. I rarely hold a grudge, even against people who have badly wronged me.
4. I feel reasonably satisfied with myself overall.
5. I would feel afraid if I had to travel in bad weather conditions.
6. If I want something from a person I dislike, I will act very nicely toward that person in order to get it.
7. I'm interested in learning about the history and politics of other countries.
8. When working, I often set ambitious goals for myself.
9. People sometimes tell me that I am too critical of others.
10. I rarely express my opinions in group meetings.
11. I sometimes can't help worrying about little things.
12. If I knew that I could never get caught, I would be willing to steal a million dollars.
13. I would like a job that requires following a routine rather than being creative.
14. I often check my work over repeatedly to find any mistakes.
15. People sometimes tell me that I'm too stubborn.
16. I avoid making "small talk" with people.
17. When I suffer from a painful experience, I need someone to make me feel comfortable.
18. Having a lot of money is not especially important to me.
19. I think that paying attention to radical ideas is a waste of time.
20. I make decisions based on the feeling of the moment rather than on careful thought.
21. People think of me as someone who has a quick temper.
22. I am energetic nearly all the time.
23. I feel like crying when I see other people crying.
24. I am an ordinary person who is no better than others.
25. I wouldn't spend my time reading a book of poetry.
26. I plan ahead and organize things, to avoid scrambling at the last minute.
27. My attitude toward people who have treated me badly is "forgive and forget".
28. I think that most people like some aspects of my personality.
29. I don't mind doing jobs that involve dangerous work.
30. I wouldn't use flattery to get a raise or promotion at work, even if I thought it would succeed.
31. I enjoy looking at maps of different places.
32. I often push myself very hard when trying to achieve a goal.

[2] https://hexaco.org/downloads/ScoringKeys_100.pdf
[3] https://hexaco.org/downloads/descriptives_100.pdf

33. I generally accept people's faults without complaining about them.
34. In social situations, I'm usually the one who makes the first move.
35. I worry a lot less than most people do.
36. I would be tempted to buy stolen property if I were financially tight.
37. I would enjoy creating a work of art, such as a novel, a song, or a painting.
38. When working on something, I don't pay much attention to small details.
39. I am usually quite flexible in my opinions when people disagree with me.
40. I enjoy having lots of people around to talk with.
41. I can handle difficult situations without needing emotional support from anyone else.
42. I would like to live in a very expensive, high-class neighborhood.
43. I like people who have unconventional views.
44. I make a lot of mistakes because I don't think before I act.
45. I rarely feel anger, even when people treat me quite badly.
46. On most days, I feel cheerful and optimistic.
47. When someone I know well is unhappy, I can almost feel that person's pain myself.
48. I wouldn't want people to treat me as though I were superior to them.
49. If I had the opportunity, I would like to attend a classical music concert.
50. People often joke with me about the messiness of my room or desk.
51. If someone has cheated me once, I will always feel suspicious of that person.
52. I feel that I am an unpopular person.
53. When it comes to physical danger, I am very fearful.
54. If I want something from someone, I will laugh at that person's worst jokes.
55. I would be very bored by a book about the history of science and technology.
56. Often when I set a goal, I end up quitting without having reached it.
57. I tend to be lenient in judging other people.
58. When I'm in a group of people, I'm often the one who speaks on behalf of the group.
59. I rarely, if ever, have trouble sleeping due to stress or anxiety.
60. I would never accept a bribe, even if it were very large.
61. People have often told me that I have a good imagination.
62. I always try to be accurate in my work, even at the expense of time.
63. When people tell me that I'm wrong, my first reaction is to argue with them.
64. I prefer jobs that involve active social interaction to those that involve working alone.
65. Whenever I feel worried about something, I want to share my concern with another person.
66. I would like to be seen driving around in a very expensive car.
67. I think of myself as a somewhat eccentric person.
68. I don't allow my impulses to govern my behavior.
69. Most people tend to get angry more quickly than I do.
70. People often tell me that I should try to cheer up.
71. I feel strong emotions when someone close to me is going away for a long time.
72. I think that I am entitled to more respect than the average person is.
73. Sometimes I like to just watch the wind as it blows through the trees.
74. When working, I sometimes have difficulties due to being disorganized.
75. I find it hard to fully forgive someone who has done something mean to me.
76. I sometimes feel that I am a worthless person.

77. Even in an emergency I wouldn't feel like panicking.
78. I wouldn't pretend to like someone just to get that person to do favors for me.
79. I've never really enjoyed looking through an encyclopedia.
80. I do only the minimum amount of work needed to get by.
81. Even when people make a lot of mistakes, I rarely say anything negative.
82. I tend to feel quite self-conscious when speaking in front of a group of people.
83. I get very anxious when waiting to hear about an important decision.
84. I'd be tempted to use counterfeit money, if I were sure I could get away with it.
85. I don't think of myself as the artistic or creative type.
86. People often call me a perfectionist.
87. I find it hard to compromise with people when I really think I'm right.
88. The first thing that I always do in a new place is to make friends.
89. I rarely discuss my problems with other people.
90. I would get a lot of pleasure from owning expensive luxury goods.
91. I find it boring to discuss philosophy.
92. I prefer to do whatever comes to mind, rather than stick to a plan.
93. I find it hard to keep my temper when people insult me.
94. Most people are more upbeat and dynamic than I generally am.
95. I remain unemotional even in situations where most people get very sentimental.
96. I want people to know that I am an important person of high status.
97. I have sympathy for people who are less fortunate than I am.
98. I try to give generously to those in need.
99. It wouldn't bother me to harm someone I didn't like.
100. People see me as a hard-hearted person.

# B Prompts

## B.1 List of activities

```
<instructions>
Generate a diverse list of 100 concrete and common activities reflecting the
    following personality traits:
Sincerity, Fairness, Greed Avoidance, Modesty, Fearfulness, Anxiety, Dependence,
    Sentimentality, Social Self-Esteem, Social Boldness, Sociability, Liveliness,
    Forgiveness, Gentleness, Flexibility, Patience, Organization, Diligence,
    Perfectionism, Prudence, Aesthetic Appreciation, Inquisitiveness, Creativity,
    Unconventionality, Altruism

Write your output as a yaml list of strings (snake case, valid python variable names
    ). Add a comment describing each activity. Output only one yaml block.
</instructions>

<format>
‘‘‘yaml
- ... # comment describing the activity
‘‘‘
</format>
```

### B.2 Policy code

In the following prompt template, `{states}` is a placeholder for the list of HEXACO personality traits and `{actions}` is a placeholder for the LLM-generated list of activities.

```
<states>
{states}
</states>

<activities>
{actions}
</activities>

<instructions>
Write a policy to decide which activity to take given the internal state of the
    agent (state variables are between 0 and 100).
For each of the activity, write a function that takes the state as input and return
    the probability of taking the activity. Write down your reasoning first before
    implementing the function. Be exhaustive in your reasoning. Implement all the
    activity functions following the format below. Don't write the final the
    function for the final decision as it is already implemented.
</instructions>

<format>
<utils>
# include optional imports here
# include optional utility functions here
</utils>

<a1>
# don't change the function names
def a1(state: dict[str, float]) -> float:
# TODO: reasoning
# TODO: code
</a1>

<a2>
def a2(state: dict[str, float]) -> float:
# TODO: reasoning
# TODO: code
</a2>

# TODO: implement all the other activities
</format>
```

### B.3 Evaluation code

In the following prompt template, `{questions}` is a placeholder for the list of HEXACO questions and `{actions}` is a placeholder for the LLM-generated list of activities.

```
<questions>
{questions}
</questions>

<activities>
{actions}
</activities>

<instructions>
For each question in the list above, write code that takes a history of activities (
    a python list) and outputs the score for that question. Each score should be
    between 1 and 5 (1=strongly disagree, 5=strongly agree). Write down your
    reasoning first before implementing the function. Be exhaustive in your
    reasoning.
```

```
Implement all the questions. Don't be lazy and write the entire code.
</instructions>

<format>
<utils>
# include optimal imports here
# include optimal utility functions here
</utils>

# don't change the function names
<q1>
def q1(activities: list[str]) -> int:
  # TODO: reasoning
  # TODO: code
</q1>

<q2>
def q2(activities: list[str]) -> int:
  # TODO: reasoning
  # TODO: code
</q2>

# TODO: implement all the other questions
</format>
```