This is an expansion on the Flashbots research prize announcement thread and a followup thread.

- ***[Update on Jul 25]** page now include takeaways and results of the hackathon*

For any questions, contact sxysun on Twitter or Telegram @sxysun

The Prize

Flashbots is doing a **research partnership** with augmenthack.xyz by offering **15k Research Grants (FRPs)** to high-quality projects with a research angle coming out of the hackathon that are willing to have an in-depth engagement for R&D after the event.

Since this is a research grant, your project doesn't have to be traditional hackathon projects, it can be in the form of research-athon (think about the experiment section of a paper, something like grid-world games is a decent start!). Here are some previous FRP research projects that we supported for you to get a flavor of what kind of projects we like, we like projects that *ask deep questions*.

High-level Ideas

We are intersted in exploring how AIs can coordinate with each other using smart contracts, the "robot money."

Suzuha's tweet perfectly captures the spirit!

We believe in this thesis because traditional social technologies like policies and regulation are *inadequate for coordinating games with AI agents* (because the characteristic of a game with algorithmic pariticipants is that it's played very fast), while crypto-economic commitments can provide a key technology for the cooperation among AI agents and help to overcome some of the shortcomings of traditional social technologies.

For an expansion on the high-level ideas, see CredibleCommitments.WTF and its summary reading, but the TLDR is that we are interested in answering:

- **How can we better design blockchains (a technology for implementing commmitments) for coordination games with algorithmic agent participants (cooperative AI)?**

Specific Ideas

1. Generative Model Coordination

Can we create a decentralized system where different fine-tuned generative models coordinate via smart contracts to fulfill a user's general prompt? *This is already a huge problem for big AI corps, after all, user intent is high-level but they've too many models to choose from!*

For example you have one fine-tuned model for creating characters in a game and another fine-tuned model for reading science fiction. Now, you want to create a short film about a post-modern cyberpunk story, what do you do? You probably have a high-level prompt that neither of the two models can work perfectly to satisfy but need to work together.

We ask: *can decentralized technology enable better coordination between fine-tuned models than existing ways (which is to train another "middle manager" AI to use other fine-tuned models as tools using RL with user behavior data) of tackling this problem?*

2. Cooperation ability for Foundation models

As soon as we implement any cooperative AI with real incentives, we'll run into the problem of MEV (strategic behavior in the interaction between agents).

For example, when one party's learning algorithm is committed (transparent), the other party that is interacting with the algorithm could adversarially manipulate the inputs to gain an advantage in game play. We encounter such scenarios in reality a lot in pricing models in advertisements or online vendors.

As a result, **foundation models better embody the ability to use smart contracts (paper) to protect themselves from strategic behavior like MEV** (e.g., require a programmable privacy commitment to "dumb the game down" such that even if you expect the other player to be smarter she cannot do much because you are confident the game is too simple to manipulate).

We ask: **is it possible to embed the ability to use smart contracts in foundation models to make them robust to strategic behavior?** How much advantage can an adversary with strategic behavior get (how much MEV can it extract) from a vanilla foundation model agent? Can the addition of smart contracts help foundation model agents improve their ability to cooperate even in a hyper-adversarial environment?

One concrete direction to go could to be to implement a game of CATAN or Diplomacy where foundation model agents can play with the ability to use smart contracts (which unlocks more strategies, like options/futures/extortions). Of course, one would have to modify the goal of those to games to make them not zero-sum (e.g., change goal to maximize number of victory points after X turns).

Notice here we are considering foundation models that employ some strategic behavior, so not exactly LLMs, but LLMs or AgentGPT-style AIs are also interesting to study. **In fact**, one interesting project could be to compare the strategy-proofness of LLM-based agents and other foundation models.

3. Preference Expression in Natural Language

There's been buzz like "enabling diverse preference expression and satisfaction," but the elicitation cost of this preference is simply too high, because users don't actually know precisely what they want at a given point!

Idea (i): **one could fine-tune an LLM to interface with human language and translate it into a smart contract/smart transaction.** This idea is great because by translating the intent into a formal language, rather than directly dealing with it, the system could reason about strategic behaviors better (e.g., make sure the prompt is "aligned")!

Idea (ii): **one could also create a recommender system that learns the user's preference and then expresses that preference as on-chain data.**

4. Emergent Behavior

We know RL agents could coordinate with each other via the use of outcome-contigent contracts, now, what if they use smart contracts? what kind of cooperative behavior would emerge? e.g., in zero-sum games, due to gas, would they learn to not use blockchain? when we offer them different kind of contracting ability, how will they behave differently? will game-theoretic results accurately predict their behavior?

More interestingly, do we expect them to not use the contracts if the blockchain is not MEV-proof? One could imagine implementing such a multi-agent system in Minecraft/any grid-world game and augmenting the game by giving the agents ability to use smart contracts.

Potential recipe for action:

1. Identify simple but representative Cooperative AI problems and implement a version of them.
    - e.g., games on github
2. Design a protocol for agents to commit (with smart contract or ZKML) to a model as a strategy, and for contract proposers to randomly requests proofs of cooperation, at a sustainable rate based on the benchmark results obtained during the previous step.
3. Design and conduct experiments with new types of contracts, potentially involving multiple rounds or multiple proposers.

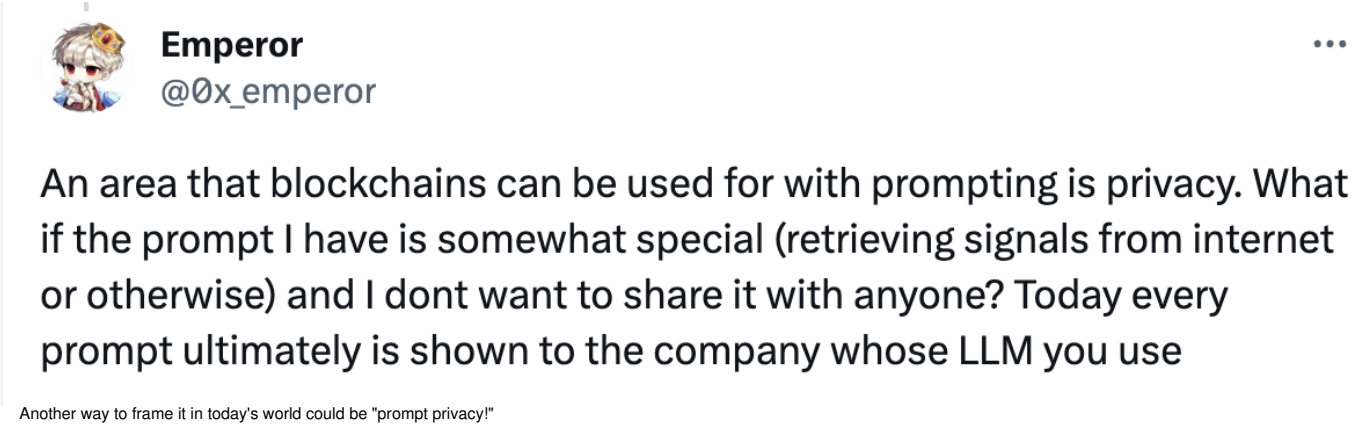
This direction is heavily tied to **2. Cooperation ability for Foundation models.**

5. Negotiation

Negotiation is interesting because users don't really know their intents, *when you use Twitter you are not telling Twitter what kind of tweet you want to see, you are letting Twitter tell you.*

However, in a world without decentralized AI, Twitter is telling you too much and your preferences are overly engineered. For crypto to be relevant to AI, we must encourage development of "personal delegation AI" that acts in YOUR own interest.

A cool hackathon project would be to **see if it is possible to combine a personal assistant AI with a recommender system by making them both "commit," or, sign smart contracts, when they interact.**



Another way to frame it in today's world could be "prompt privacy!"

6. Automated Mechanism Design

We know that, using deep learning, it is possible to automatically design mechanisms.

We ask: **is it possible to encode the credibility or the MEV-proofness of smart contracts and transactions as a constraint on the learning?** What about different notions of credibility? Can we automatically learn something akin to verifiable sequencing rules that either SUAVE or PEPC could implement to provide better UX for crypto?

A concrete project would be based on simulations!

7. ZKML for cooperation in multi-agent settings

following is from a note written by Jonathan.

In a recent paper titled "Equipping MARL Agents with Contracts to Solve Social Dilemma", the authors explore how contracts can be used to address social dilemmas in multi-agent reinforcement learning (MARL). Contracts, in this context, refer to an agent transferring a payoff to other agents when a specific event or action occurs. Section 3 of the paper highlights the challenges involved in accurately identifying deviations from socially optimal play and punishing them. The paper states, *"Moreover, the proposer needs to be able to infer which agents deviated from socially optimal play, in order to accurately punish deviation. Without any further assumption, this requires knowledge of the current state and the actions of all players."* ZKML offers a solution by allowing a contract proposer to easily infer deviations from the committed strategy by requesting a proof to unlock the contract's reward. This approach eliminates the need for the proposer to possess complete knowledge of the present state and actions of all players. ZKML has the potential to enable more complex contracts spanning multiple rounds of interaction.

ZKML facilitates accurate punishment decisions by allowing agents to prove compliance or deviation from socially optimal play without directly revealing their private actions or states. It enhances privacy, maintains the confidentiality of sensitive information, and fosters cooperation by providing a verifiable mechanism for detecting and addressing deviations from cooperative behavior.

7. Your Idea!

Learnings from the hackathon

Dynamic pricing

Query auction: Graph, which serves blockchain data queries, runs a query auction where users bid for query execution and indexers compete to execute the queries. Since it is hard for indexers to price complex queries, a dynamic pricing model based on RL was implemented. It was interesting to see the performance and convergence of such pricing models.

Market maker rebates: one project was about dynamic pricing for rebates for market makers. Along same lines but applied in DeFi settings, one could imagine implementing some kind of dynamic pricing model for transaction fees in AMMs or lending protocols, e.g., giving higher fees for LPs that provide liquidity longer, or when volatility is larger, or based on the counterparty (orderflow origination).

Natural language intent

Some participants implemented a "natual language to on-chain txn" tool (Brian). This means it's now easy for AI agents to use smart contracts and get easy access to financial markets. It would be intersting to just plug AgentGPT into it and then we can have agents that interact on-chain. For example, this is equivalent to giving algorithmic agents an on-chain bank account that they own and can freely engage in financial transactions in satisfaction of their goals.

Automated smart contracts

One project (NuroContract) is: using past on-chain data about users' intents and the outcome of executing those intents, have a LLM generate a smart contract that would improve the execution of those intents, kind of like a crypto application generator.

"Ok hear me out, personalized generative smart contracts.""

For example, knowing that that all of the previous users have intents about exchanging assets, and they interacted with Uniswap to get some outcome on the asset settlement, can the AI learn a better mechanism such that given those intents of exchanging assets, gives the users a better settlement result, for example, the Cowswap protocol (where users are first allowed to settle against each other before settling against the liquidity provider).

The original idea of the project was: based on group chat data, generate a smart contract for the group to coordinate on what they are talking about, e.g., a meetup.

This is the project that was given the prize.

Some future directions of this could be:

- bridging this generative model with the traditional study of automated mechanism design in Econ. It would be fun to compare the two approaches.
- *strategy-proof mechanisms*: the AI that is generating the contract is nothing but another agent that has been delegated actions of faithfully executing the mechanism, and it can often be algorithmic (regretNet, etc,.). It would be important to study automated mechanism design in a strategic setting where other automated mediators are present. For example, can we make the generative smart contract agent learn a MEV-free smart contract, as defined in Clockwork finance.

Others

supplychain management system: one team implemented a system where factories can lend idle production lines.

In retrospect, I think we could have done better if we throw out githubs with half-built projects beforehand, cuz many participants does not have a deep knowledge in MEV, so my prompts were kinda hard to understand, but it's still cool.