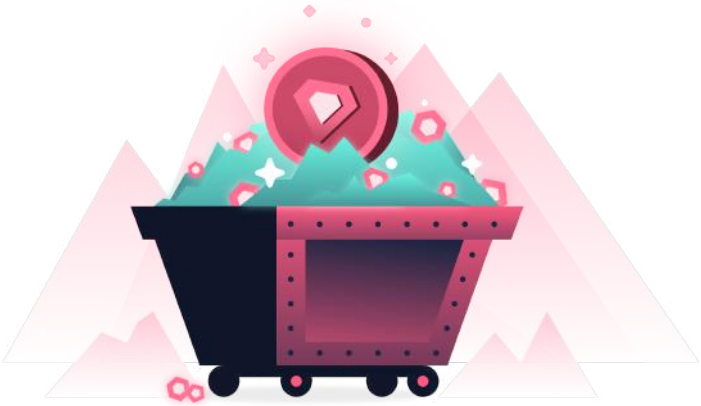
Draft Version 0.07

Gems Protocol

Decentralized Mechanical Turk powered by Ethereum

Gems is a protocol for contracting workers to perform micro tasks. Workers stake tokens in order to prove validity of their tasks and earn a reusable computed trust score, enhancing the cost-efficiency of the network while democratizing access to scalable micro task workers.



Rory O'Reilly Kieran O'Reilly
rory@gems.org kieran@gems.org

November 7, 2017

Contents

1 Introduction

1.1 Background

Micro tasks are small tasks that require human judgment, can be completed independently over the internet, and are part of a larger unified project. Micro tasks aid organizations that need humans to complete tasks computers otherwise cannot. Micro tasks are used in a wide range of large scale distributions, whether by machine learning researchers to gather structured labeled data for training purposes[1], or by large companies, such as Facebook, Google, Twitter, Ebay, and Linkedin, to refine their production ready solutions and provide further model training[2]. Billions of micro tasks are completed each year. The Chinese government alone uses 2 million micro task workers to aid in censoring the internet[3].

Amazon, in 2005, created Amazon Mechanical Turk (MTurk), an online marketplace where requesters can crowdsource data collection by paying workers to perform micro tasks online. Since then, demand for micro task workers has increased. In a World Bank Report in 2015, industry experts concluded the largest crowdsourcing platforms Amazon Mechanical Turk and CrowdFlower, a venture backed company that has publically raised over $58 million[4] and is focused on enriching data used for AI, would quadruple their revenue from 2013 to 2016[5].

Unfortunately, the current centralized marketplaces are wrought with problems. Amazon charges requesters a minimum 20% fee on any amount paid out to workers and an additional 5% for requesting workers that have a good track record[6]. Furthermore, tasks with 10 or more assignments are charged an additional 20% fee, totaling upwards of 40% in total fees[7]. With over half of MTurkers reportedly earning less than $5 an hour[8], MTurk has been described as the sweatshop of online crowdsourced labor[9]. To use Crowdflower, on the most basic plan requesters must pay a $3000 USD onboarding fee and $1500/month continuation fee with an annual contract[10]*. This does not include payment to workers, which is paid separately. Furthermore, there is an additional 15% fee on any payments given to workers.

Additionally, manually verifying the accuracy of results from workers is difficult. To ensure quality results, requesters currently rely on redundancy, oftentimes having 5-15 workers perform the same task to form a consensus[11]. This is undoubtedly wasteful. Requesters pay 5-15x as much per project, and workers lose out on a potential increase in payment.

Non-managed platforms do not supply interfaces needed to accomplish specific tasks. Currently, requesters will either build their own tools[12] or pay large fees to startups looking to tackle the enterprise market[13], wasting millions of dollars.

Micro task workers also have issues with the system. The entirety of the labor pool does not have access to current micro task marketplaces. A staggering 38%[24] of the world's population does not have a bank account, hurting their ability to work in and contribute in current

systems. Furthermore, for those that do have bank accounts, problems with payments, lost checks, etc. come into play[14]. This does not even include the fees and cut that central institutions take away from the marketplace.

Lastly, the signup and approval process for workers is nebulous and privacy invasive, leaving many wondering if they will even be able to contribute to this marketplace. The unofficial subreddit for workers on MTurk is riddled with users asking when they will be approved and many others offering condolences saying it may take months if ever.

1.2 Overview

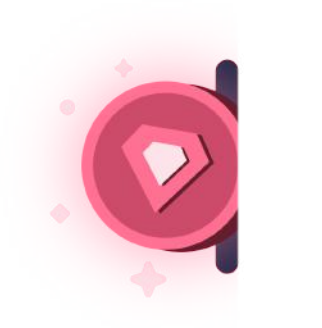
The solution is Gems. In this paper we introduce Gems, a decentralized, open-sourced, human task crowdsourcing protocol built on top of the Ethereum blockchain. Using the Gems Protocol, anyone can tap into the power of scalable micro task workers without needing to worry about task verification, trust, or payments. Gems is designed to disincentivize malicious actors and reward fair players. The Gems Protocol is comprised of a staking mechanism to ensure task completion, a trust mechanism to track worker integrity, and a payment system to reduce transaction fees. The GEM token, a multi-utility token, fuels the Gems Protocol. Gems uses its token mechanism to enforce the behavior of all participants, instead of being regulated by a single operator.

We further introduce the first application using the Gems Protocol, aptly named the Gems Platform. The Gems Platform connects those who want work done (henceforth known as "requesters") and human workers (henceforth known as "miners"). By using the Gems Protocol, the Gems Platform removes socioeconomic barriers that exist in centralized alternatives (e.g. large fees, market inefficiencies, need for bank accountants, etc.). Anyone can build on top of the Gems Platform, creating "modules" that are interfaces for particular human tasks. The first module the Gems team will build focuses on labeling data for AI.

In this paper, we describe the Gems Protocol, the GEM token mechanism, the current problem with existing platforms, the Gems Platform, and the reasons we initially focus on AI tasks. The Gems Protocol, Gems Platforms, and modules built on top of it are collectively known as the Gems Network.

In summary, the Gems Network addresses the following:

1. Removing the middleman taking a large fee
2. Verifying accuracy of results from crowdsourced tasks
3. Supplying and building reusable interfaces
4. Removing the need for existing banking infrastructure
5. Properly incentivizing and disincentivizing miners and requesters



2 Gems Network Overview

The Gems Network facilitates the extension and efficient operation of the micro task community by allowing organizations to reliably deploy micro task miners. The Gems Protocol, through validating task completions with the Gems Staking mechanism and enabling miners to have a reusable trust mechanism through the Gems Trust Score, enforces compliance of network participants. The Gems Protocol enables the creation of various platforms that are built on top of the Gems Protocol. Platforms that are created have no intrinsic fee imposed by the Gems Protocol, broaden the scope of the labor supply, and eliminate inefficiencies in the market place. Gems allows any application to utilize efficient online scalable workforces.

2.1 Gems Staking Mechanism

The Gems Staking mechanism provides disincentive for malicious actors, enhancing the efficiency of the Gems Network. Through the Gems Staking mechanism, currently created through Ethereum-based smart contracts, miners, requesters, and verifiers stake tokens on the validity of their work and against the validity of others' work, providing a palpable disincentive for doing tasks incorrectly.

2.2 Gems Trust Score

The Gems Trust Score is an indicator of how reliable an individual on the network is. By using the individual's history of completing tasks accurately, efficiently, and consistently, the Gems Trust Score is formed and linked to the network participant's Ethereum wallet address. Because it is easy to create a new address, trust scores are not easily bootstrapped; miners need a long proven track record to obtain a high score.

Miners with very high scores are eligible to verify the work of other members on the network, allowing them to increase the overall accuracy of the system while earning extra money at a

higher hourly rate. Unreliable miners will be removed from the network, keeping the quality of work high.

2.3 Gems Platform and Modules

The first application using the protocol is the Gems Platform, a marketplace for matching miners and requesters. The Gems Platform charges no central fee, and utilizing the GEM Token and Gems Protocol, eliminates existing economic inefficiencies. Modules, reusable interfaces for completing specific tasks, are interfaces that are built on top of the Gems Platform.

2.4 Gems Payment System

Transactions on the Ethereum network are not free; they require gas. To allow for micro payments and staking without using gas, the Gems Protocol uses a system of payment channels. Payment channels allow for secure off chain payments without using gas and for the grouping of those payments on the blockchain at a later date. The Gems Payment System will be used by requesters, verifiers, and miners to increase the cost efficiency of the network.

3 Gems Platform

3.1 Existing Market Landscape

Existing micro task marketplaces include MTurk, currently dominant in the marketplace, and Crowdflower. MTurk and Crowdflower both cater to academics and researchers[15]; tasks created on MTurk can range from clicking a link, going to a webpage and annotating an image, to completing a survey, where as Crowdflower is focused on enriching AI data.

Current micro task platforms all work in a similar way: workers sign up, requesters submit work to be done, workers do the work, and workers get paid by the requesters. In very few instances do the central organizations provide tangible benefit throughout the process, excluding simply building the initial infrastructure. Centralized organizations, like Crowdflower and MTurk, act as a rent seeking middleman between the two interested parties, which leads to numerous inefficiencies in the market.

The limitations with current platforms are outlined below, as well as the Gems solution. The Gems solution increases pay for workers, decreases cost for requesters, enables participation for those unbanked, and introduces a quality management solution, while facilitating the broadening of the network.

3.2 Problem 1: Fees

Both Crowdflower and MTurk charge requesters who submit tasks to be performed, adding downward pressure on the amount workers can potentially be paid.

To analyze and conceptualize the effect of a network fee on requesters and miners, we analyze the competitive labor market in equilibrium without a network fee, and then compare this to the competitive labor market with a network fee. In a competitive labor market, workers prefer higher wages and employers prefer lower wages. The labor market balances out these competitive desires and reaches an equilibrium point for wage and workers employed. In this example, we use a simplified model for the elasticity of the supply and demand curves.

Figure 1 represents the labor market in equilibrium without a network fee. The labor market is in equilibrium when the supply of labor, S, and the demand for labor, D_0, are equal. As shown in figure 1, when in equilibrium, workers are paid W_0, and E_0 workers are employed. The coral shaded region represents the employer's surplus, and the turquoise shaded region represents the worker's surplus. The combined of the shaded regions produces the total welfare of the actors in this market.

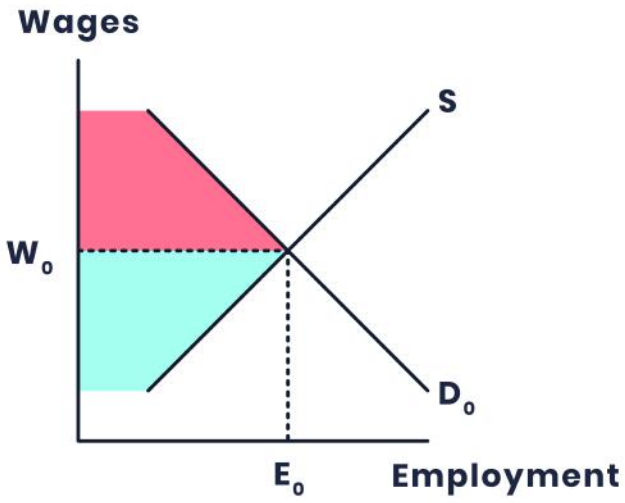


Figure 1

Figure 2 demonstrates the effect when a \$1 fee per hour of wage is imposed on employers. In Figure 1, it was shown the employers were willing to pay a total of W_0 to hire E_0 workers. Intuitively, because of this new \$1 fee per hour of wage, employers are now willing to pay a total of $W_0 - 1$ to hire E_0 workers. Therefore, when a network fee is introduced that targets employers, demand for labor shifts down from D_0 to D_1, as shown in Figure 2. The network fee imposed moves the market to a new equilibrium point, where wage falls to W_1, and employment falls to E_1. While the wage of the worker is W_1, the total cost of hiring this worker is $W_1 + 1$.

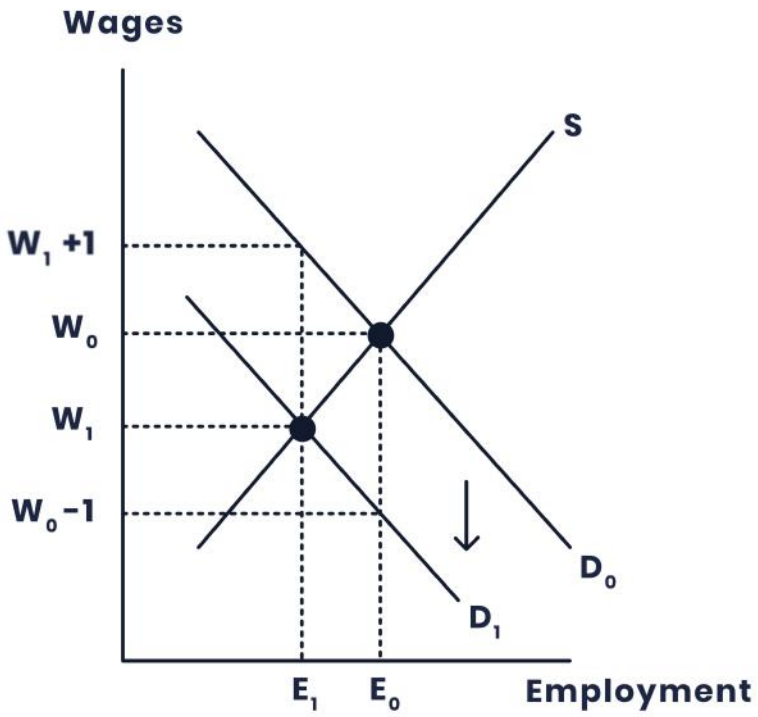


Figure 2

Figure 3 shows the repercussions of the introduced fees. The yellow shaded region represents the total fee that is collected, which is the product of employment, E_1 and the fee wedge introduced, 1. The gray shaded region represents deadweight loss in the market. Deadweight loss is introduced by economic inefficiency, where by total economic surplus is "lost" in the market. As shown in Figure 3, both employers and workers lose surplus.

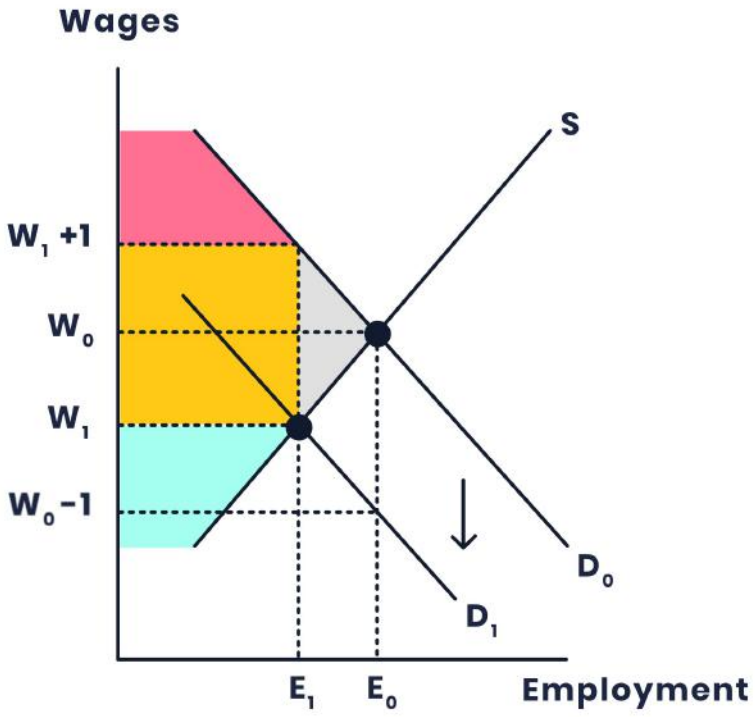


Figure 3

While the network fee is imposed on employers, the fee affects workers as well. The labor market shifts part of this burden to workers, as denoted by wage falling from W_0 to W_1. As shown, network fees lower workers' and employers' surplus, takes surplus away from the market in the form of fees, and introduces economic inefficiencies in the form of deadweight loss.

GEM Solution: No Fees

Gems does not charge a centralized fee for transactions on the platform. The only cost on

the Gems Platform is gas used on the Ethereum network. As shown above, workers and employers will likely share this benefit and return to a fair equilibrium as noted in Figure 1. The end result of having no fees on the platform is a more efficient marketplace for both miners and requesters and the removal of the rent seeking middlemen.

3.3 Problem 2: Participation

Workers in the micro task economy currently need to be approved before they can start work. For example, the approval process for MTurk demands identification [16], which introduces delays in approving workers and human/technical errors in identification.

Crowdflower forces you to jump through multiple portals and eventually asks you to link up with your Facebook account, which shares all of your public information with Crowdflower. For those that do not have a Facebook or do not want to share that much information, Crowdflower is not a viable solution.

The World Bank reported that workers and academics in India indicate that an increase in the stringency of identity requirements for worker registration is a major constraint to the registration of new workers for micro task platforms[17].

GEM Solution

On the Gems Platform, users do not need to verify their identity to complete tasks. The reason for verifying identity is rooted in preventing malicious actors from abusing the network. The Gems Protocol, and specifically the staking token mechanism (outlined below), makes the Gems Platform less susceptible to such attacks. In the future, as the Gems Platform branches into areas where information about the miners (geographical, demographical, etc.), may be helpful to requesters, we may allow miners to **voluntarily** verify in order to access these specific tasks. Furthermore, if we elect to allow members to voluntarily verify certain information, we can utilize emerging decentralized protocols (e.g. Bloom) to promote an accurate, speedy, and network contributive approach. For those that verify their identity, we may elect to mint Gems for new miners and allow them to hook into an Ethereum faucet to cover transaction costs, in order to facilitate the broadening of the Gems Network early on.

3.4 Problem 3: Payments

The payment processes on existing centralized platforms are not conducive for the unbanked. To receive payments on MTurk, workers need to have a bank account for transfers, be willing to wait for checks (to deposit into their bank), or be willing to be paid in some form of gift card that is not necessarily useful for their current situation. Furthermore, the processes even if one does have a bank account aren't perfect. A 2017 Wired article titled "*Amazon's Turker Crowd Has Had Enough*" detailed how workers' checks for a sizable period of time

were not being delivered correctly[14].

Crowdflower makes payments via Paypal. While Paypal is a slightly better option than gift cards and mailed checks, it too has its problems. Paypal takes a 2.9% fee + 0.30 per transaction. With many Crowdflower workers earning only 0.03 a task, 0.30 + 2.9% results in a sizeable pay reduction. For those who are able to become Crowdflower Elite and avoid the Paypal fee, there are still similar inefficiencies and shortcomings in a centralized organization. For example, funds may not be sent on a timely manner or at all[18].

Inherently, these restrictions on payments and accessibility, mainly from needing a firm banking infrastructure, limit the amount of workers that can be on the network. We believe labor that has access to good internet, but not a good banking system, should be able to work in the labor market. We believe tapping into this labor pool and enabling those who can and are willing to work will have consequential socioeconomic effects for Gems and beyond.

GEM Solution: Paid on the BlockChain via GEMs

With the advent of blockchain technology, formerly necessary institutions that introduce inefficiences, like banks, credit cards, and verification methods of the aforementioned can become a thing of the past. Miners are rewarded for successfully accomplishing a task with GEMs, the reward system of the Gems Platform. Storing GEMs, just like any other ERC20 token, does not involve detailed information about your bank, allowing those who are unbanked and would like to participate to easily do so. Blockchain payments, through the Gems Payment Channels, will reduce necessary transaction fees, which will make transactions on the platform incredibly cost efficient.

As blockchain technology gets further disseminated at an increasing pace, being paid on the blockchain will become increasingly attractive. Furthermore, as OTC protocols such as 0x begin to gain momentum, the barrier to transfer between GEMs and other tokens decreases dramatically.

3.5 Problem 4: Usability

Designing for crowdsourced micro tasks is difficult. This is in part due to maintaining an interface that doesn't necessitate high computer literacy. With that being said, the interface can certainly be improved. It is widely known that MTurk's UI/UX is quite dated. A Microsoft study determined that the interface was severely undeserving of its user base at the time[19]. Furthermore, requesters wanting to use MTurk for tasks will typically build their own tool or website and ask the worker to visit it or have it iframed on MTurk's interface. Crowdflower's Crowdflower Markup Language (CML) does little to save requesters time in setting up simple tasks, while more difficult tasks are left again to the requesters themselves to build.

Poor user interfaces (UIs) and user experiences (UXs) vastly increase the time it takes to perform a given task. The result is workers performing fewer tasks per hour and requesters paying for time that isn't used constructively. Furthermore, frustrating and unintuitive UI/UX leads to tasks that are performed incorrectly. Lastly, having requesters build their own UI/UX is a waste of time and resources.

There are numerous browser extensions that attempt to fill holes in Amazon's capabilities[20]. For example, Turkopticon has over 30,000 downloads, with its main functionality being the ability to see how workers rated the task/requester.

GEM Solution: Open Sourced UI/UX and Modules

4 Gems Modules

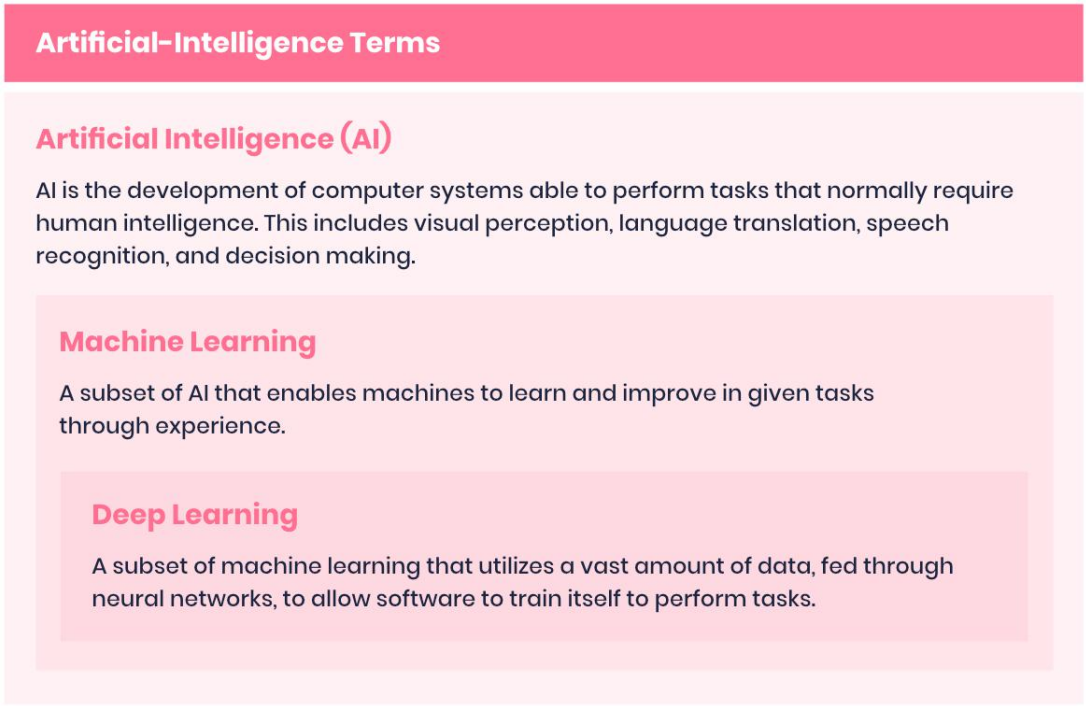
Gems Modules are interfaces built on top of the Gems Platform. Modules are the tools miners use to perform tasks. The Gems team will work hands-on with early requesters to build reusable UI/UX open source modules to be used on the Gems Platform. Furthermore, the Gems team will build and open source reusable components of modules that will allow the ease of use for anyone to build another module.

Centralized institutions oftentimes believe in closed sourced software or open sourcing bits and pieces that are not necessarily important for their primary business. However, we, as a decentralized platform where we take no ongoing network cut, believe otherwise.

We believe having open sourced and reusable components in the Gems environment will help promote better network effects within the Gems Platform, allowing for miners to make more money per hour, requesters to get their jobs completed faster, and wasted time to be eliminated on both fronts.

4.1 First Module: AI

The Gems team will first build Modules that revolve around AI tasks. We're starting with AI tasks as we believe there is a large need for labeled data in this growing industry. In the AI world, data is the currency of the future. The need for data and specifically scalable annotation services comes from the wide adoption and use of machine learning and specifically deep learning, both subsets of Artificial Intelligence.



Why is data needed for machine learning?

Learning phases can be broken down into two main categories, supervised learning and unsupervised learning. The fundamental difference between supervised learning and unsupervised learning is that in supervised learning, there are known input variables x and corresponding output variables y. In a supervised learning process, an algorithm analyzes the input variables and corresponding output variables, attempting to infer a mapping function for new examples $y = f(x)$. Supervised learning is presently much more widespread than unsupervised learning, as supervised learning is currently far less complex. During the training phase, a neural network is fed thousands of labeled inputs, idealistically learning to classify them. Therefore, well labeled datasets are paramount during the supervised training process of a neural network.

Why focus on annotating AI tasks first?

AI is one of the fastest growing industries[21]. Companies such as Google, Facebook, and Amazon, guided by the allure of using artificial intelligence and machine learning to improve their products and generate more revenue, are investing more into AI than ever before. Netflix estimates that improving search results and suggestions using AI saves them over $1Billion per year [22]. These companies' investments in AI are further shown by their willingness to spend large sums of money to acquire AI teams and companies. McKinsey estimates that tech giants have spent between $20B-30B on AI in 2016[23]. The potential for what AI, and specifically deep learning, can do is immense. Whether it's autonomous vehicles,

algorithmic trading, or even Amazon product suggestions, AI is starting to play a large role in our everyday lives and will continue to do so for years to come.

5 Gems Staking Mechanism

As miners accomplish micro tasks, the validity of their work must be verified. The two distinctions of what causes errors are malicious actors, who are looking to game the network by receiving a reward and not doing micro tasks correctly, and those who simply misunderstand the task.

5.1 Staking Method for Miners

Requesters can require miners to stake a token, or a fraction thereof, on a given task, defined by variable M_s by using our staking smart contract. When staking M_s, the miner asserts that the work they have completed is done accurately and for the best possible results. In order to properly disincentivize malicious actors and not severely penalize miners who can improve, we propose that miners stake a relatively small portion of token. However, to prevent malicious miners causing financial harm to requesters, M_s should be large enough to pay for new verifiers and the gas used by the network.

5.2 Staking Method for Verifiers

Instead of having an abundance of miners duplicate tasks, we propose a method of "verifiers" that look through tasks and verify the validity of the work. Requesters specify whether they require verifiers and how many verifications are needed before the work is considered accurate. Verifiers, as opposed to miners, stake a smaller portion of token, denoted by variable V_s. This is because verifiers are highly qualified in their decision making process, and are less likely to be malicious actors.

The verifiers' stake ensures verifiers still have a palpable incentive to be correct in their assertions. The verifiers' stake is governed by a smart contract. Verifiers lose their stake if other verifiers, or the requester, overturn their work. Verifiers are returned their stake if their work is not overturned.

Verifiers, similar to miners, receive incentivization in the form of tokens allocated from requesters. Let the reward paid to be miners by denoted by M_r. Let the reward paid to verifiers be denoted by V_r. Let z be a number from 1 to 10 that denotes how simple (measured by time) verifying the task is. 1 being the longest, and 10 being shortest.

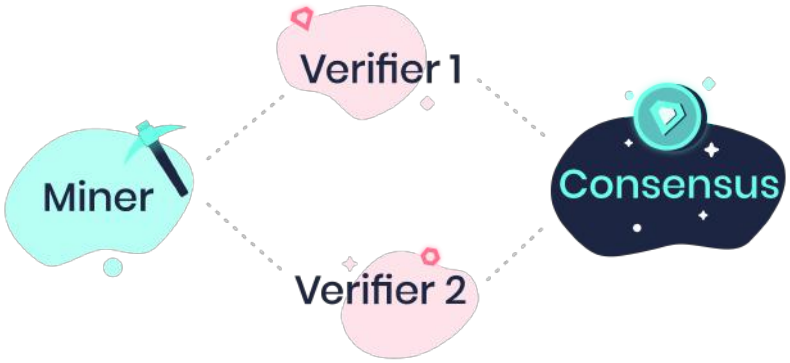
$$V_r = \min\left(M_r, \max\left(\frac{M_r}{z}, 0.2\right)\right)$$

The reward paid to verifiers is at most the reward paid to miners. Intuitively, this is because the time verifiers spend on work should be $\leq$ to the time miners spent, and therefore $V_r \leq M_r$. For simple tasks, for example a simple image annotation task, we can expect 2 things:

1. The verifier will put in the same amount of work as the miner. The verifier will have to redo the simple task. In the case of image annotation, to label a cat as a cat, there is no other way but to look at the image and decide it is a cat.
2. M_r will be less than the equivalent of \$0.20 in the equivalent of Gems. Simple tasks will pay an amount no more than \$0.20 because of their simple nature. This number can be adjusted to market rates.

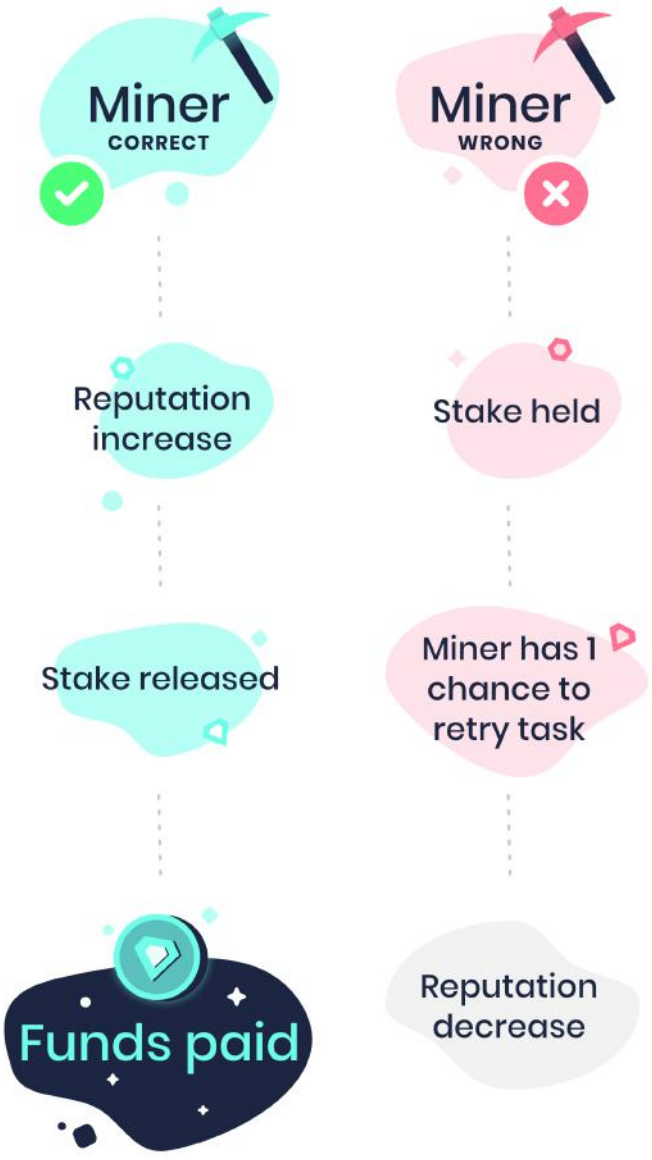
Knowing this, let's say for an image annotation task, a miner Tim earns \$0.10 in the equivalent of GEMs and the requester determines a fair z of 10. The verifier will have to put in the same amount of work to verify the image annotation as Tim did when working. The M_r in this example is \$0.10. Following the above equation, the verifier will earn $\min(0.10, \max(\frac{0.10}{10}, 0.2))$ or $\min(0.10, 0.20)$ which equals \$0.10 in the equivalent of GEMs. In this case the verifier and the miner put in the same work and received the same reward. Also noteworthy, the z in this case does not matter because the M_r is low. Requesters will not be able to pay verifiers less for the same work because of this hard constraint.

Verifiers should be paid less than miners when their verification of miner work is less time intensive than the task itself. This should have a benefit of saving overall money for requesters by diminishing the total amount of miners needed to correctly label a task, allowing requesters to allocate more money per task. For example, in a task like image segmentation where a miner has removed the background from an image, the verifier can look at the results and know if it is correct without completing the job. Let's say this job earns Tim \$0.75 in the equivalent of GEMs per task completion and the requester determines a fair z of 5. Following the above equation, the verifier will earn $\min(0.75, \max(\frac{0.75}{5}, 0.2))$ or $\min(0.75, 0.20)$ which equals \$0.20 in the equivalent of GEMs. In this case the verifier put in less work and received a proportionate less amount of the reward.



If verifiers correctly verify the miner's accuracy (e.g. 2 or more verifiers reach consensus), the miner's staked claim, M_s, is returned to the original miner, and the miner is rewarded with M_r. If the verifiers reach consensus and say the work is inaccurate, the miner has the opportunity to redo the work within a given timeframe or lose their original stake. If the miner successfully completes completes the task on the second try, the miner is penalized V_r, and their total reward is $M_r - V_r$.

If verifiers do not reach consensus on the miner's accuracy, a new set of verifiers are asked to verify the task. The new verifiers will be paid through the lost stake of the verifier who was wrong in their previous assessment.



5.3 Staking Method for Requesters

To prevent requesters who are malicious actors, requesters stake their GEMs incentivization pool for each task. Having requesters essentially put their reward in escrow ensures that requesters do not act in bad faith by reporting successfully completed tasks as incorrect.

The process of having both miners and verifiers further reduces the cost for requesters and provide new avenues for miners to make money. Miners with high trust scores have the ability to earn more money by verifying other tasks. Because of the staking mechanism, miners have a disincentive to perform a task poorly, and thus, we assume that requesters do not have to hire as many miners for a particular dataset, lowering their total output cost or allowing them to label more data with the same financial constraints.

6 Gems Trust Score

6.1 Trust Mechanism

Every participant will have a Trust Score linked to their Ethereum address. The trust mechanism is integral for permitting miners to earn the privilege of being a verifier. The trust mechanism will also protect miners and verifiers from poorly designed or fraudulent requesters. Furthermore, the trust mechanism helps the network discern between malicious actors and simple mistakes. Malicious actors detected through the trust mechanism will be banned from the platform.

When calculating the Gems Trust Score for miners, we use a confidence interval to take into account the proportion of successful task completions and the number of tasks completed. To calculate our Gems Trust Score, we first calculate the lower bound of the binomial proportion confidence interval as calculated by the Wilson score interval. The lower bound, c_1, is defined by the below algorithm where $\hat{p}$ is the fraction of positive task completions, n is the total number of task completions, and $z_{\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution.

$$c_1 = \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}$$

As the proportion of positive tasks, $\hat{p}$, increases, the confidence level increases. As the number of tasks n increases, the maximum value of c increases. After obtaining the initial confidence score c_1, we calculate c_2, which represents the confidence interval for tasks completed successfully on the first try, through the same algorithm as c_1.

$$c_2 = \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}$$

However, when we calculate c_2, $\hat{p}$ will be equal to the fraction of tasks completed successfully on the first try. We will then average c_1 and c_2 to get the final confidence score. If we did not calculate c_2, it would be possible for miners to harm the network by always failing on the first try and then succeeding on the second try. This is possible because they don't have financial harm if they succeed on the second attempt. However, by calculating the fraction correct on the first attempt, miners will lose trust by failing on the first attempt.

Information on requesters history is crucial to ensure miners and verifiers do not lose funds to abuse. We propose a holistic report on each requester available to miners and verifiers before they accept a micro task. Helpful stats in the report include the aggregate number of tokens requesters have paid out on the network, the total number of times the requester accepted a verifier's work, and the total number of times the requester reverted verifier's work. Additionally, the Gems Platform can measure the requesters promptness in releasing funds, their fairness based on estimated pay per hour for miners and verifiers, and their overall communication based on ratings by the community.

7 Gems Payment System

The Gems Payment System groups payments in as few transactions as possible to allow for cost-efficient staking and micro payments. As work is approved through the Gems Network, earnings and stakes are released through a payment channel to miners and verifiers. Additionally, when starting a task, miners and verifiers send their stake via payment channels. At any time, the miner or verifier can receive the balance they are owed from their payment channel. Likewise, requesters can receive the balance of the stakes they are owed from their payment channel.

8 Conclusion and Summary

As presented in this paper, Gems provides economic efficiency to network participants, properly incentivizes miners to accomplish tasks, disincentivizes malicious actors, opens access to those who are unbanked, and opens access to the labor supply of micro task workers without intrinsic network fees.

Figure 4 below details the direct differences between Gems and existing solutions.

Figure 4

8.1 Looking Forward

The Gems Staking mechanism and Gems Trust Score are initial components of the open sourced Gems Protocol. Phase 1 of the Gems Protocol will have the above constraints. As new mechanisms are developed by the Gems team and contributors, verifiers will have multiple options on how to verify their work and reward miners. For example, a group of academic contributors may create a solution where miners are rewarded based on how well their trust scores contribute to a high confidence level the work is completed correctly. Verifiers will be able to decide which mechanism is most apt for their verification use case. We are looking forward to continuing to conduct research, grow, and create new approaches for solving trust and verification problems. We look forward to working with others to further facilitate the evolution of the Gems Protocol.

Acknowledgements

Many thanks to those that helped to review this white paper:

Austin Baird
Luis Iván Cuende
Sol Kim
Joey Krug
Jesse Leimgruber
Lamar Letts
Kamal Obbad
Joe Urgo
Devon Zuegel

References

[1] http://www.nytimes.com/2007/03/25/business/yourmoney/25Stream.html

[2] https://www.crowdflower.com/success-stories/

[3] http://www.cnn.com/2013/10/07/world/asia/china-internet-monitors/index.html

[4] https://www.crunchbase.com/organization/crowdflower

[5] http://documents.worldbank.org/curated/en/138371468000900555/pdf/ACS14228-ESW-white-cover-P149016-Box391478B-PUBLIC-World-Bank-Global-OO-Study-WB-Rpt-FinalS.pdf

[6] https://requester.mturk.com/pricing

[7] https://docs.google.com/presentation/d/1Y_lvecsOefCfkXkkrdFtW1GH-NkoZidKzKXqFVJaLH4/edit#slide=id.ga3bec7adb_0_199

[8] http://www.pewinternet.org/2016/07/11/research-in-the-crowdsourcing-age-a-case-study/

[9] http://www.utne.com/science-and-technology/amazon-mechanical-turk-zm0z13jfzlin

[10] https://www.crowdflower.com/pricing/ —*Crowdflower has since obfuscated their pricing, asking inquirers to contact directly

[11] http://neerajkumar.org/writings/mturk/

[12] https://requester.mturk.com/help/faq

[13] https://www.crowdflower.com/use-cases/

[14] https://www.wired.com/story/amazons-turker-crowd-has-had-enough/

[15] http://www.pewinternet.org/2016/07/11/the-requesters-a-mix-of-academics-and-businesses/

[16] http://www.behind-the-enemy-lines.com/2017/01/why-was-my-amazon-mechanical-turk.html

[17] http://documents.worldbank.org/curated/en/138371468000900555/pdf/ACS14228-ESW-white-cover-P149016-Box391478B-PUBLIC-World-Bank-Global-OO-Study-WB-Rpt-FinalS.pdf

[18] https://www.paypal-community.com/t5/Job-board/Payment-Issue-on-CrowdFlower-Elite/td-p/1254696

[19] https://www.microsoft.com/en-us/research/wp-content/uploads/2010/12/DEV2010-turk-final.pdf

[20] https://www.reddit.com/r/HITsWorthTurkingFor/comments/sdyuo/turking_scripts_and_addons/

[21] https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/how-artificial-intelligence-can-deliver-real-value-to-companies

[22] http://www.businessinsider.com/netflix-recommendation-engine-worth-1-billion-per-year-2016-6

[23] https://www.forbes.com/sites/louiscolumbus/2017/07/09/mckinseys-state-of-machine-learning-and-ai-2017/#430a863175b6

[24] http://documents.worldbank.org/curated/en/187761468179367706/pdf/WPS7255.pdf