

# Hive: Collective Design Through Network Rotation

NILOUFAR SALEHI, UC Berkeley, USA

MICHAEL S. BERNSTEIN, Stanford University, USA

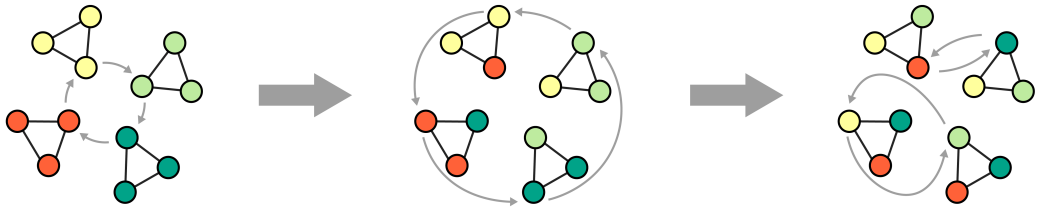


Fig. 1. Hive facilitates engagement with diverse viewpoints by rotating team membership in a collective over time. We introduce algorithmically-mediated *network rotation* to manage who should move, and when, to bring positive external influence to a team.

Collectives gather online around challenges they face, but frequently fail to envision shared outcomes to act on together. Prior work has developed systems for improving collective ideation and design by exposing people to each others' ideas and encouraging them to intermix those ideas. However, organizational behavior research has demonstrated that intermixing ideas does not result in meaningful engagement with those ideas. In this paper, we introduce a new class of collective design system **that intermixes people instead of ideas**; instead of receiving mere exposure to others' ideas, participants engage deeply with other members of the collective who represent those ideas, increasing engagement and influence. We thus present Hive: a system that organizes a collective into small teams, then intermixes people by rotating team membership over time. At a technical level, Hive must balance two competing forces: (1) networks are better at connecting diverse perspectives when network efficiency is high, but (2) moving people diminishes tie strength within teams. Hive balances these two needs through *network rotation*: an optimization algorithm that computes who should move where, and when. A controlled study compared network rotation to alternative rotation systems which maximize only tie strength or network efficiency, finding that network rotation produced higher-rated proposals. Hive has been deployed by Mozilla for a real-world open design drive to improve Firefox accessibility.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: Design; online collaboration; participatory design; innovation; teams.

## ACM Reference Format:

Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 151 (November 2018), 26 pages. <https://doi.org/10.1145/3274420>

Authors' addresses: Niloufar Salehi, UC Berkeley, School of Information, South Hall, Berkeley, CA, 94720, USA, [nsalehi@berkeley.edu](mailto:nsalehi@berkeley.edu); Michael S. Bernstein, Stanford University, 353 Serra Mall, Stanford, CA, 94305, USA, [msb@cs.stanford.edu](mailto:msb@cs.stanford.edu).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2573-0142/2018/11-ART151 \$15.00

<https://doi.org/10.1145/3274420>

## 1 INTRODUCTION

People with shared challenges often gather online to discuss and envision shared outcomes. Examples include grassroots Twitter movements to advocate for new harassment policies, Mechanical Turk workers advocating for fairer treatment [64], and competitions to address global grand challenges [24, 36]. For a collective to make progress, its members must deliberate to understand and articulate the problem, then collectively imagine goals and paths forward to alternative futures. How might social systems support large scale collective design?

Open design efforts have long sought to bring together collectives online to ideate, but in practice, participants rarely work together to understand others' viewpoints or build constructively on each others' ideas. Recent work has aimed to increase collaboration by intermixing ideas, for example by grouping and combining submissions [67, 68, 82], by encouraging contributors to use each others' submissions [5, 43], or by exposing contributors to each others' high-level approaches [78]. This work all shares an underlying assumption that the information — the idea itself — is central. However, knowledge is situated and developed in the context of social relations, and organizational behavior research has demonstrated that mere exposure to others' ideas does not result in meaningful engagement with those ideas [11, 61, 63]. The result is open design efforts with far less intermixing and collaboration than originally envisioned [4].

In this paper, we suggest an alternative approach: rather than intermixing ideas, intermix the people who bring the ideas. Research shows that collaboration with people who bring new perspectives results in engagement and increases the influence of new ideas, in a way that abstract exposure to those ideas does not [11, 61]. A new person's addition to a group forces existing group members to re-evaluate hidden assumptions [37] and overcome fixation [11] by engaging with the new member. Higher tie strength, trust, and acceptance between actors increase that influence [63].

Our goal with this research is to introduce a new class of social computing system that iteratively weaves a collaboration network by bringing members into contact with diverse perspectives over time. We present *Hive*, a system that organizes a collective into teams and rotates people in the collective across teams at regular intervals as they work towards a shared goal. *Hive* connects people together from distant parts of the collaboration network to facilitate the spread of viewpoints within the collective. Underlying *Hive* is a new approach that we term *network rotation*, which changes team membership while maintaining high levels of tie strength within teams. To support network rotation, *Hive* models a collaboration network as a weighted graph where each edge captures the tie strength between two people. We use this collaboration network to identify who should be moved onto other teams and when.

Intuitively, we prefer a tightly knit network where most people are connected through a small number of links to a disconnected network with silos and bridges. To achieve a good network structure, we want to make a small number of moves that tighten up the collaboration network by bringing together people who are far away from each other. We frame this as an optimization problem where we maximize a weighted sum of two functions. First is *network efficiency*, or the average path length between two nodes in the collaboration graph, which captures how efficiently viewpoints and ideas can spread in the network. Second is *tie strength*, or the average edge weight between people who are in the same team, which captures how familiar team members are. The optimization framework considers them jointly: moving people between teams increases network efficiency but reduces tie strength, so the algorithm must trade these off. This is a large combinatorial search space, and cannot be explored through traditional means. We introduce a stochastic search algorithm that finds an effective solution by exploring thousands of possible moves.

Our evaluation asked whether *Hive* and network rotation led to effective design, both in a field experiment and in a public deployment with accessibility advocates at Mozilla. For the field

experiment, we recruited 115 participants from Amazon Mechanical Turk to work together over three days to design proposals for an open-ended design challenge: How might we design a neighborhood common space to be a place that brings people together to foster strong community and mutual aid? We randomly assigned participants to one of three conditions: control, network rotation, and efficiency. A judge who was an expert in design thinking compared the concepts submitted in each condition, blind to condition. Proposals submitted by teams in the network rotation condition were rated significantly higher than control teams, whereas teams in the efficiency condition were not statistically distinguishable from those by control teams. Analysis of the chat logs suggests that rotation did shift the topic of teams' discussion, supporting the proposed mechanism. Qualitative and survey results indicated that membership change caused helpful disruptions without measurably decreasing members' psychological safety or interrupting social dynamics.

We then sought to understand whether the results from the field experiment generalized to a real-world deployment: Mozilla used Hive to run an open design drive on Firefox accessibility. Over one hundred volunteer disability experts, designers, and programmers collaborated on a week-long design drive, resulting in 60 proposals. Mozilla supported the proposals and plans to move ahead with prototyping and deploying five of them. Participants also enjoyed the diversity of viewpoints and fresh perspectives brought in through membership change.

Our main contribution in this paper is a system for supporting collective design by computationally adapting the collaboration network over time. We contribute network rotation, an algorithm that creates the conditions for people to engage deeply with a small group while benefiting from the scale and diversity of the collective and building social ties across the network. Hive is a step toward social systems that model and center peoples' relationships to one another.

## 2 RELATED WORK

Hive presents a model of collective design that fuses the dynamics of close teamwork with the scale of online collectives, with the goal of engaging a diverse range of viewpoints through membership change. We draw on social psychology and organizational behavior research studying the conditions for open-ended, creative collaboration. We base our approach to collective design on traditions of human centered and participatory design.

Hive is a system that coordinates team members to achieve a shared goal by tracking and shaping the underlying social network. Prior work has designed algorithms for online team formation in social networks that focus on team composition, for example organizing teams based on skills required for the task, communication overhead, workload, and compatibility [1, 33, 40]. Hive builds on this line of research by focusing not on the initial composition of the team, but on the dynamics of membership: how to balance both local team effectiveness and global network structure.

### 2.1 Membership change brings new insight to teams

Brainstorming, the practice of creative idea generation, relies on the cognitive stimulation facilitated by multiple members of a group, rather than individuals in isolation [54, 71]. Interacting with others can spark new ideas that individuals may not achieve alone [6]. By working together in a team, members build common ground, learn to coordinate, and discover how to utilize each person's unique skills [35, 58].

However, without changes in composition, teams risk becoming complacent, isolated, and less innovative [2, 25]. New members bring novel perspectives to stale teams that can stir creativity. In fact, major innovations most often occur when new members join a team [17] and innovation can come from both newcomers and novel combinations of old-timers [55]. Even when the new member's conflicting points and opinions are flawed, the act of raising these points nevertheless leads to new insights in groups [50]. Our model of network rotation relies on this intuition.

There are costs to altering team membership. First, the process of adding a new team member can be disruptive to the working processes of a team, especially at high turnover rates [12, 57]. If team membership is algorithmically determined, the team will wonder why the system made the decisions it did [26]. Additionally, familiar teams develop psychological safety over time [22, 69]. Disrupting this stability could have harmful implications for the team, because psychological safety fosters a team environment where members are comfortable taking risks with each other [15]. Without psychological safety, strong norms around conformity discourage new members from offering alternative perspectives before they are fully assimilated, leading to missed opportunities for generating and obtaining creative insight from an external source [19, 51]. New members too are less likely to impose their ideas and to initially observe the group, with the goals of understanding the existing group dynamics and gaining eventual acceptance by the group [18, 23, 47].

Hive amplifies the concepts behind this literature, introducing the first system to facilitate membership rotation. It aims to support the team processes that this literature has found to be successful. Hive facilitates membership rotation with a global perspective of the network, enabling it to bring new perspectives in contact with each other. We draw on the insights from this literature to design for the interplay between the group's dynamics and the new member's fresh perspectives, avoiding rotating members too quickly and enabling team members to build up familiarity before moving.

## 2.2 Open innovation and crowdsourced design

Large-scale innovation platforms and initiatives—such as IBM's "Innovation Jam" for product ideas, the U.S. government's change.gov site, and Google's 10 to 100th project for charity—promise to gather people to find innovative solutions to challenging problems. This solution has been used by large organizations looking for external sources of innovative ideas. However, the majority of the ideas that have been generated are deemed repetitive, commonplace, and unwieldy in volume [4, 30, 59, 67] — not a collaboration among a diversity of viewpoints.

The cost of sifting through such high volumes of low quality ideas often does not justify the benefit: IBM had 100 senior executives spend several weeks filtering ideas, the government shut down change.gov prior to its intended end date due to participation levels that overwhelmed the system, and 3000 Google employees were put 9 months behind schedule by managing the enormous mass of submissions [4, 30]. To address this, platforms have introduced systems to more efficiently surface high quality submissions; nevertheless, filtering accuracy is still an issue and often the best results are those that are non-obvious or may not be rated most highly [30]. Instead of extracting large volumes of individual ideas and evaluating them post-hoc, we propose creating the conditions for participants to engage with each other and collaborate.

One method for fostering collaboration in generating new ideas is by exposing people to each others' ideas. Crowd-driven idea maps [66] and analogies [78, 81] connect contributors to concepts that they haven't seen before, prompting new ideas [8]. Enabling a traversal through the space of ideas requires extra metadata, but crowds can author this metadata as they work [67] or it can be learned passively by aggregating behavior traces [72]. Once this metadata has been authored, people search through solution spaces at a higher level of abstraction [66, 79] and embed each other's solutions directly into their own [43, 46, 52]. Rather than putting people in contact with external *ideas*, our work presents an approach of putting people in touch with other *people*. Research has shown that people are more receptive to new ideas when they interact closely with those who bring the ideas [11].

Prior research has explored involving external crowds in design work. Crowds can provide feedback on in-progress designs [38, 41, 75, 76], helping designers iterate more quickly. The crowd's feedback can be used to guide the crowd's next phase of ideation [29]. However, the crowd is

not always an expert critic [74]: external expert facilitators can be effective at guiding the crowd in their ideation process [9], or in setting a creative direction [28, 31]. Crowds are also able to identify domains whose experts might be helpful for the problem at hand [80]. Hive expands on this point, combining it with knowledge that the best insights come from distant perspectives, not proximal ones [74], to create an algorithm that rotates distant members onto the team. Relative to this prior work, Hive does not view the crowd as static or external: we contribute a method that uses other members of the collective to provide outside perspectives, and continuously evolves the collaboration network to engage with those perspectives.

Taken together, this prior literature on open innovation and crowdsourced design systems has placed abstract ideas front and center — showcasing them, analyzing them, and remixing them. Hive contributes an alternative perspective of placing relationships front and center, and trusting good ideas to arise from successful collaborative relationships.

### 3 HIVE

Hive forms and evolves teams over time within a large collective to achieve an open-ended goal such as collective design. In this section, we describe a scenario of how Hive can be used to support collective design. Then, we will detail the network rotation algorithm that underpins Hive.

Suppose that a large collective is interested in a design goal, for example improving the web browsing experience to better account for disability and inclusion. All participants sign up on Hive. Hive invites them to Slack and keeps record of their Slack username so that it can communicate with them via the Slack API. An administrator sets up the Hive project, determining for example the team size (e.g., 3–5 people), the number of phases, and instructions to give teams at each phase. In this scenario, the administrator creates phases for each stage of the design thinking process based on the OpenIDEO process: inspire, define, ideate, prototype, and test [56].

Hive enables identification of team leads, who act as facilitators. Facilitators are central to successful and satisfying online collaborations [20, 53]. The team lead's role is to thread discussions, schedule meeting times, onboard new members, sustain commitment, capture ideas, and make sure that the daily deliverables are submitted. To maintain organizational memory, Hive does not rotate team leads across teams.

Hive then splits the large collective into many small teams with one team lead each. Hive notifies each person on Slack of their team, creates a channel on Slack for the team, and invites members to the channel. Each team is assigned a Google Drive folder where the instructions for each phase are in a shared Google document. Hive notifies the team of the instructions for the first phase by sharing a link to the folder with them. The instructions provide resources and specify the process and deliverables. The teams then work collaboratively for the duration of the phase. They collaborate using shared document editors, text chat, and photos of sketches.

Hive is now ready to begin the network rotation process and start the next phase. Hive's network rotation algorithm identifies who to move and where to move them, based on the collaboration network formed by the team groupings in all rounds so far. The administrator triggers Hive to execute the rotation, and Hive sends a message to each rotated member's old team channel and to their new team's channel encouraging a welcome for the new team member. Then, it removes rotating members from their original team channels and adds them to their new teams' channels (Figure 2).

The process continues, with Hive rotating team members in each new phase to diffuse influence and ideas around the network. Next we describe Hive's network rotation algorithm.

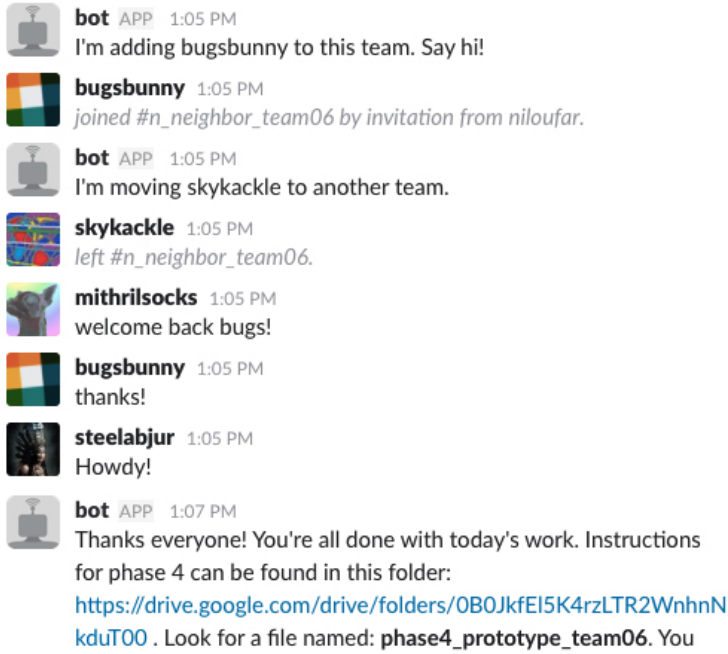


Fig. 2. Hive's Slack bot performs membership changes.

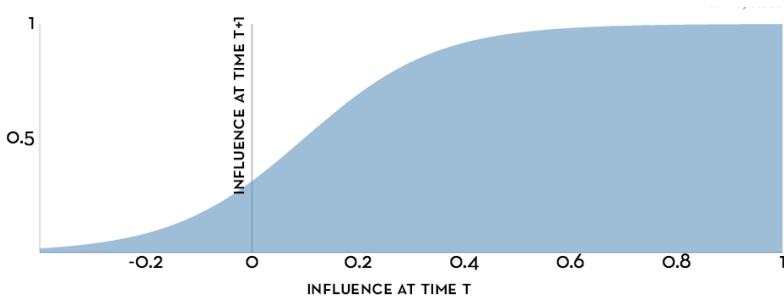


Fig. 3. We apply a logistic function that increases the tie strength between two people who work together in a team. We choose a logistic function as a reasonable proxy for tie strength because it captures relationships that are accelerate rapidly at first then asymptote, similar to familiarity in the real world [27]. For example, if two teammates have never worked together and have tie-strength of 0, one day of working together will increase it to  $\text{logit}(0) = 0.16$ .

#### 4 NETWORK ROTATION

Hive's central goal is to create the conditions for people to engage with a diversity of new perspectives while maintaining a level of within-team tie strength that can foster meaningful collaboration. To do so, Hive gradually changes team membership through a process we call *network rotation* (Figure 4). Network rotation requires an algorithm that computes who to move, and where to move them — an exponential search space. In this section we first describe how Hive models the collaboration network, then we describe an optimization objective that allows us to measure the



effectiveness of a given rotation, and finally we describe a stochastic search algorithm that we use to execute that optimization.

#### 4.1 Constructing a Collaboration Network

To support network rotation, Hive requires a model of the collaboration network as it evolves over time. The collaboration network is encoded as a weighted graph where the weight of each edge between  $i$  and  $j$  captures the tie strength between person  $i$  and  $j$ . We define tie strength in the range  $[0, 1]$ , with 0 indicating a relationship between two strangers and 1 indicating an extremely strong tie, with other relationships falling somewhere in between [21].

Our model must capture how new ties change over time. Network rotation can operate with any model of tie strength evolution over time. We offer one model here as a proof of concept based on prior literature. Tie strength and how it changes over time relies on many different social factors that are difficult to isolate [44]. However, prior literature generally assumes an upper limit, a decrease in tie strength due to lack of communication, and a positive relationship with time spent [7, 44]. Other factors also play a role such as the emotional intensity of a relationship, however due to the limited time spent in our studies and for the sake of simplicity, we rely solely on time spent communicating within our system.

For each  $i$  and  $j$  who were on a team together at time  $t$ , we apply a function that increases the edge weight between  $i$  and  $j$ . To model the growth of tie strength over time, we use a logistic function (Figure 3). Logistic functions are commonly used to model natural and social processes that grow exponentially at first but that reach a saturation point and stop growing, such as diffusion of innovations [44]. A logistic function has also been used to predict the strength of persistent ties over time [48]. Likewise, with collaboration, teammates begin weakly connected, achieve increased performance through familiarity [65], but eventually stop improving [27]. The constants of the logistic function can be tuned to impact how long teammates work together before their tie strength is high.

Second, the model must capture how the strength of past ties dampens over time. We update the collaboration network at every time step  $t$ , based on the collaboration network at time  $t - 1$ . We fade tie strength from prior time steps by first multiplying all edge weights by a constant dampening factor  $\lambda$ :  $0 < \lambda < 1$ . This multiplier retains old familiarity, but weakens it.

In our deployments, we used a logistic function with  $k = 8$  and a dampening factor of  $\lambda = .8$  to achieve a reasonable model of tie strength increasing each day over a week. For deployments with more phases, lowering  $k$  and increasing  $\lambda$  would slow down these dynamics.

#### 4.2 Optimization: Tie Strength and Network Efficiency

Now, we use the collaboration graph to identify who should be moved onto other teams to bring new perspectives. Intuitively, we want to make moves that tighten up the collaboration graph by bringing together people who are far away from each other in the collaboration network, but to avoid making so many moves that it is disruptive to the teams. We frame this as an optimization problem where we maximize a weighted sum of two functions (Figure 4). First is *network efficiency*, which captures how different viewpoints have intersected with each other so far. Network efficiency, or  $\text{Efficiency}(G)$  for a collaboration graph  $G$ , is calculated as the average path length between every two nodes [34]. Second is average *tie strength*, which captures how familiar team members are. Average tie strength, or  $\text{TieStrength}(G)$ , is calculated as the average edge weight between nodes who are in the same team. Moving people between teams increases network efficiency but reduces tie strength, so the algorithm must trade these off. First, we normalize  $\text{TieStrength}(G)$  by a constant value based on network size (in our deployment  $c = 0.005$ ) so that it can be traded off against  $\text{Efficiency}(G)$ . Next, we formalize this tradeoff with a parameter  $\alpha$ . This produces an optimization

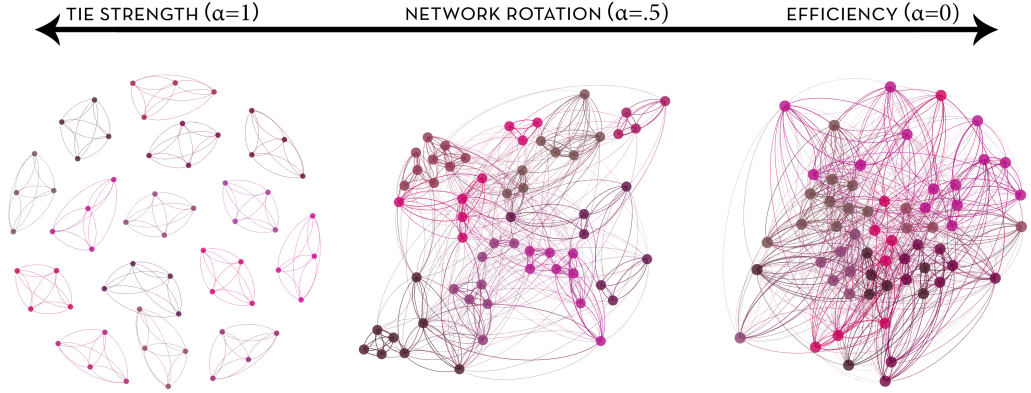


Fig. 4. Network rotation must balance network efficiency against tie strength in the collaboration graph. Left: Caring only about tie strength prompts the algorithm to create static teams. Right: Caring only about network efficiency breaks apart teams each round, which is highly disruptive. Middle: our algorithm balances the two goals, maximizing network efficiency while minimizing disruption.

function  $f$  on the collaboration network  $G$ :

$$f(G) = \alpha * \text{TieStrength}(G) + (1 - \alpha) * \text{Efficiency}(G)$$

In our deployments, we set  $\alpha = 0.5$ , equally weighting both functions. Finding the right subset of all possible moves to maximize this function is the topic of the next section.

### 4.3 Stochastic Search

Finding an effective rotation represents an extremely complex search space —  $O(2^N)$  — where we cannot afford a brute force search. Unlike most exponential search problems that have relatively tractable solutions with integer programming or related techniques, network rotation is particularly challenging because its objective is nonlinear: we must update edge weights after each considered set of moves, which may shift prior network efficiency measures. The main source of complication in this problem is that it is impossible to know the value of a single move independently of all of the other moves that will happen. The value of moving a person to a new team depends on who else moves from or to that team, and to or from other teams across the network. For example, if person  $p$  moves from team  $t_1$  to  $t_2$ , an algorithm can evaluate this single move's impact on tie strength and network efficiency. However, the algorithm may now wish to move someone from  $t_2$  to another team as part of the rotation. When it does so, it may move the very person that made  $t_2$  so attractive a location for  $p$  to join in the first place, removing the basis for putting  $p$  on  $t_2$ . So, if this were an integer program, it would require  $O(2^N)$  variables and have poor structure — not totally unimodular — resulting in modern solvers unable to solve it effectively.

In this section, we introduce a method for finding effective network rotations: a stochastic search algorithm that explores thousands of possible moves and chooses a set of moves that increase the value of  $f(G)$ . Stochastic search is a good fit for problem spaces that do not have a helpful structure affording traditional optimization techniques. Instead, stochastic search explores alternatives and follows a gradient with random restart. We propose this approach not as the only possible algorithmic solution, but to establish the fruitfulness of the approach.

Hive's stochastic search algorithm calculates a network rotation by finding a set of moves  $S$  that create a new graph  $G$  maximizing the function  $f(G)$ . To do so, it considers a set of all possible



**Algorithm 1** Network Rotation with Stochastic Search

---

```

1: procedure STOCHASTICSEARCH( $G$ )
2:    $S_{current} \leftarrow \{\}$ 
3:   while true do
4:      $S_{candidate} \leftarrow \text{AddValidMove}(S_{current})$ 
5:      $G' \leftarrow \text{Transform}(G, S_{candidate})$ 
6:      $G_{current} \leftarrow \text{Transform}(G, S_{current})$ 
7:     if  $f(G') > f(G_{current})$  then
8:        $S_{current} = S_{candidate}$ 
9:     if  $\text{Random}() \leq \epsilon$  then
10:      return  $S_{current}$ 

```

---

moves, picks one probabilistically, evaluates the optimization function after that move, and adds it to the current solution if it increases the optimization function  $f(G)$ . The algorithm repeats these steps with a random restart to ensure that it explores many possible alternatives.

Algorithm 1 describes the stochastic search algorithm. Take  $S_{current}$  to be the best solution so far. At the start,  $S_{current}$  is an empty set. The algorithm adds a valid move to the set at each step. The function  $\text{Transform}(G, S)$  returns a graph  $G'$  that is the result of performing the moves in the set  $S$  on the graph  $G$ . The algorithm can then compare  $f(G')$  with the current best solution and update it. To prevent getting stuck in local maxima, with a probability of  $\epsilon = 2 \times 10^{-6}$  at each step, the algorithm does a random restart. Hive runs this algorithm repeatedly and takes the best solution out of all of these iterations. In our deployment we set the algorithm to run one thousand times for a reasonable run time.

In order to add a valid move to the set  $S$ , the algorithm first creates a list of all moves of one person to another team. We constrain this set to valid moves, which cannot involve a team lead and cannot put a team's size outside the administrator's limits (Hive's default is 3-5 people). We include an additional optimization: the algorithm might move person  $p$  to team  $t_1$ , and then move that same person to another team  $t_2$ , which would be the same as moving  $p$  once to  $t_2$ . To reduce the search space, valid moves can only move each person once per round. Finally, the algorithm picks a move at random from the set of valid moves, creating a new graph  $G'$ , and evaluates the optimization function  $f(G')$  on that graph.

## 5 FIELD EXPERIMENT

Our goal in evaluating Hive is to answer the following questions: (i) Is network rotation more effective at supporting collective design than no rotation (i.e. control) or rotation for efficiency? (ii) Are large online collectives able to use Hive to follow the design process and collaborate on a real-world design goal? (iii) What behavior patterns emerge in small, rotating design teams of this nature? To answer these questions, we first ran a controlled study comparing Hive's network rotation algorithm against two other conditions: control (static teams), and network efficiency (rotation not constrained by tie strength). Then, Mozilla used Hive to host an online design drive on Firefox accessibility. We report on this field deployment to demonstrate how Hive could work in a real-world setting.

### 5.1 Method

We recruited participants on Amazon Mechanical Turk (AMT). To prevent language and technology use barriers, we limited our pool to people from the U.S. or Canada who have completed more than

500 tasks on AMT and have above a 95% approval rate. This process produced  $N = 115$  participants. The gender balance was 55% female, 44% male, and 1% identified as neither or did not respond. Ages: 41% were 35-44, 40% were 25-34, 8% were 45-54, 5% were 18-24, and 3% were 55-64 years old.

We gave participants the following design prompt:

How can we design a neighborhood common space to be a place that brings people together to foster community and mutual aid? If your neighborhood was given a space roughly the size of an elementary school playground, what could you do with it? Our goal is to create proposals for spaces/activities that bring together neighbors and strengthen communities.

We chose this prompt because most people have direct experiences with the topic of neighborhoods and shared spaces.

We paid participants an average of \$99.40 for their participation of around 6 hours ( $SD = 7.4$ ). The levels of commitment and the time participants spent on the project varied across teams. We used a combination of peer-reviews and evaluation of final deliverables to ensure fair payment: In line with the Dynamo guidelines,<sup>1</sup> we initially guaranteed a payment of \$90 to anyone who would complete the project and a \$50 bonus to team leads, who were responsible for creating storyboards/prototypes. Participants could also distribute credit to others who they felt deserved it, which we used along with expert evaluation of final proposals to pay bonuses.

In designing our field experiment we aimed to study three different optimization objectives for Hive's network rotation algorithm. First is tie strength ( $\alpha = 1$ ), which maximizes tie strength within teams. This approach keeps all teams fixed. Second is network efficiency ( $\alpha = 0$ ), which maximizes network efficiency by making moves that diffuse team members around the network, but disregards tie strength. Finally, balanced ( $\alpha = 0.5$ ), which jointly maximizes tie strength and network efficiency (Figure 4).

We randomly assigned participants to one of three experimental conditions (Figure 4). *Control* uses the tie strength algorithm to keep teams static. *Network efficiency* uses the network efficiency algorithm with a limit on the number of moves. *Network rotation* uses the network rotation algorithm to balance the other two conditions. The major difference between the latter two conditions is that network efficiency is not trying to achieve high tie strength within teams.

In simulations on networks the size of our study, network efficiency ( $\alpha = 0$ ) made a large number of moves, 52 moves on average. Our pilots and prior research demonstrated that this level of membership change was very disruptive to teams [65] and failed to have compelling results, making it too weak a control condition. Therefore we decided to limit the number of moves. The network rotation condition made around 27 moves, which was more feasible for the teams to manage, so we limited network efficiency to 27 moves in order to compare the effect of the algorithm objective when given the same budget of number of moves.

When participants signed up for the study, we asked them to perform an initial brainstorm and submit three ideas. We asked an independent reviewer with design experience to evaluate the ideas on a 1-5 rubric. We averaged the three scores and later used this baseline creativity score to control for baseline team member creativity in our analysis.

Hive organized members into teams of four within their condition, resulting in nine teams in control,<sup>2</sup> ten in network rotation, and ten in network efficiency. Hive then created a Slack channel for each team and guided them through a four-phase human-centered design process — empathize, define, ideate, prototype — spending one hour on each of the first three phases and three hours on the final phase. We scheduled the time for the first three phases based on participants'

<sup>1</sup><http://guidelines.wearedynamo.org>

<sup>2</sup>One team lead dropped out of the study at the outset, so the control condition had one fewer team.

availability that they had provided when they signed up. Given the length of time required for the final prototype phase, teams were asked to make their own scheduling and collaboration decisions. Each team had a team lead, who did not participate in membership changes. We provided teams with tutorials, educational videos, examples, and steps to take for each phase.

After each of the empathize, define, and ideate phases, Hive moved participants or kept them static based on the group's condition. After every phase, we asked participants to fill out a survey containing questions sourced from the psychological safety index [14, 16], rating a 1–7 Likert scale for items such as: “Members of this team are able to bring up problems and tough issues” and “Working with members of this team, my unique skills and talents are valued and utilized.” Our goal was to measure how much psychological safety — a measure of perceived ability to take risks in a team and a strong correlate of team performance — suffered when Hive changed team membership. Summing the responses on the seven-question index produced a psychological safety score for their team. We also asked participants to rate how well their team collaborated in each phase on a scale of 1 to 5, and we asked people who had experienced team changes to rate how well they (or the new member) integrated into the team.

We gathered 87 final design submissions from Hive teams, three submissions per team. We asked an independent reviewer with design experience to evaluate the ideas. We provided the reviewer with the evaluation criteria that we had given the teams. The rubric was based on one used by the OpenIDEO collective innovation platform, and included: understanding needs, choosing an important problem, finding a unique solution, demonstrating empathy, and proposing a feasible solution. To provide a more sensitive instrument than Likert-scale ratings, we utilized pairwise comparisons on allourideas.org. allourideas aggregated the pairwise rankings to assign a composite score per project.

For all analyses, submissions were anonymized and raters were blind to condition. The rating method followed a method of alignment on a training set and then independent coding (e.g., [3, 65]). An author who had not seen the teams' projects and the external rater independently reviewed ten proposals blind to team and blind to condition, then clarified the rubric based on discussion. The external rater then proceeded to rate the full dataset.

We also measured the mechanism through which the effect was predicted to occur: specifically, whether the addition of the new member changed what the teams discussed, dislodging the team from their existing design trajectory or fixation. Prior theory predicts that teams fully integrating new members would change their ideas, rather than continue on their existing path [11]. This shift should be visible in what the team discussed. We built a bag-of-words vector space model of the language used by each team in each phase. Text was pre-processed to remove stop words, lemmatized, and stemmed; we then computed the vector sum of the words as a representation of the team's language in the phase. If teams continue talking about the same ideas, cosine similarity between phases would remain high; if teams shift, cosine similarity would lower.

For qualitative analysis, we consolidated survey responses, deliverables, and chat logs according to the following pre-determined themes framed by our research question: following a collaborative design process, team dynamics, and the effect of network rotation.

## 5.2 Results

One hundred and fifteen participants completed the project in 29 teams and submitted 87 final proposals. Network rotation moved members around to diffuse their influence over the phases of the experiment (Figure 5). Cumulatively, participants exchanged 125,000 messages on Slack. In this section, we first present a comparison of the teams' final ideas in each of the three conditions, and a test of the expected mechanism through which the difference would occur. We also detail the

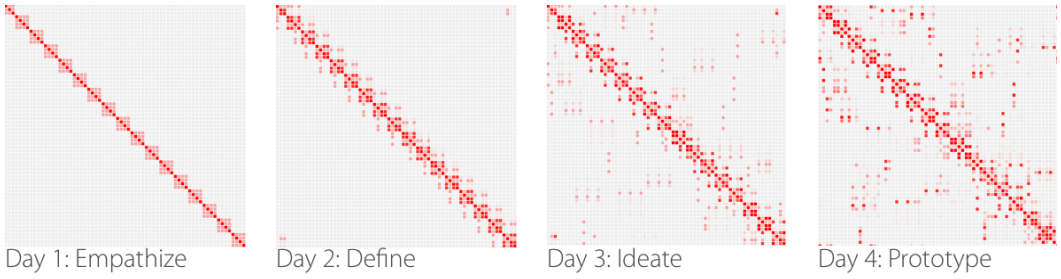


Fig. 5. The network rotation algorithm moved members of the collective into new positions to diffuse their influence. Darker cells in this adjacency matrix indicate higher tie strength.

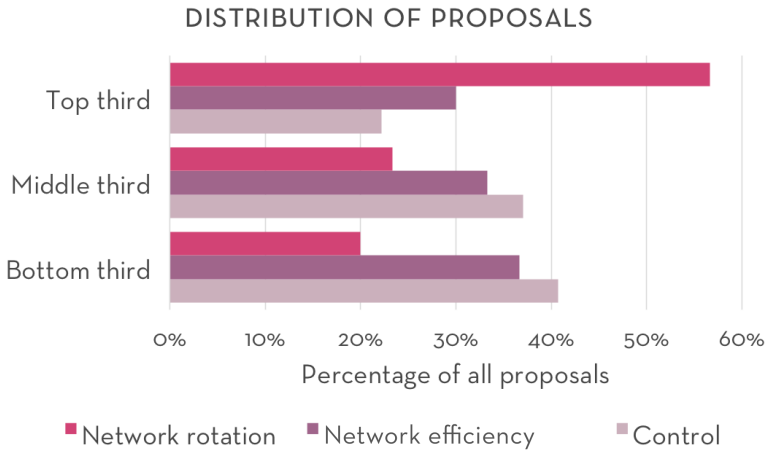


Fig. 6. Over half of the top-rated proposals were from the network rotation condition.

teams' processes and dynamics. The Discussion section will synthesize this study and the next to reflect on the conditions under which network rotation will succeed.

**5.2.1 Network rotation resulted in high-quality proposals.** What effect did network rotation have on the proposals at the top of the score distribution? *The network rotation condition dominated the top-rated proposals.* The distribution of submissions can be seen in Figure 6. In the top third of proposals as rated by the expert, half (17) were from network rotation, 9 were from network efficiency, and 6 were from control. A Chi Square test of independence comparing the number of proposals in each condition found a significant effect:  $\chi^2(2) = 8.7, p < 0.05$ . Inspecting the Pearson residuals confirmed that the network rotation condition contributed most heavily to the residuals in the model.

Considering the whole score distribution, the mean expert score in each condition, ranked descending, were network rotation ( $M = 60, SD = 20$ ), network efficiency ( $M = 47, SD = 23$ ), and control ( $M = 46, SD = 17$ ). Inspecting the overall means, the network efficiency condition was nearly equivalent to the control condition (47 vs. 46), but network rotation was 0.7 standard deviations higher on average (60 vs 46). We analyzed this data using a linear regression. We transformed the dependent variable using a box-cox transformation to ensure that the data met

Variable	$\beta$	SE	t
Network rotation	8.34*	3.86	2.16
Network efficiency	0.72	3.91	0.18
Baseline team creativity	1.62	2.38	0.68
Intercept	29.09***	8.46	3.44

Table 1. Result of linear regression on score. \*\*\*:  $p < 0.001$ , \*:  $p < 0.05$ .  $N = 87$ , Adj.  $R^2$  is 0.04.

regression normality and homoskedasticity assumptions. We used the control condition, i.e. static teams, as the baseline intercept for the categorical condition variable, and expert-rated score as the dependent variable. We also controlled for each team's baseline creativity scores. We did this by calculating a weighted average of each team member's baseline creativity score, weighted by the number of rounds that member had been on the team. Replicating this analysis with untransformed data and no control variable leads to similar results. The regression confirmed the same results (Table 1): the network rotation condition was significantly better than the intercept control condition of static teams ( $p < 0.05$ ), but network efficiency was not. A linear hypothesis comparison between the beta coefficients of network rotation and network efficiency is marginally significant ( $p = 0.05$ ), strongly suggesting that network rotation teams outperformed network efficiency teams via this measure as well.

Qualitatively, surveys and chat logs suggest that the control teams suffered from design fixation, rarely looking past their initial brainstorm:

i didn't think this was a great team overall as our styles were too similar [...] i think new blood is sorely needed all around.

In contrast, network rotation caused teams to re-evaluate their syntheses. One participant, who convinced their new team to consider perspectives formed in their previous group, reported to us:

I came up with a solution to a problem with one of the group's ideas. After that my contributions were taken into consideration [...] I came into a group with VERY different ideas than my original group.

Together, this quantitative and qualitative data suggest that network rotation introduced substantive, helpful changes to teams' ideation.

Membership change could subjectively feel costly for participants, and they sometimes complained to us about having to re-adjust and that they "*liked their previous team's ideas* :("). However, in most cases, teams were welcoming to the new member and they integrated quickly. When on-boarding worked well, the team appreciated the new ideas that the new member brought with them:

- Thanks for your ideas [...] You had some of the best ever. It really gave us a head start.  
+ I'm glad to have been helpful!! You guys had other awesome ideas too.

Membership change is most beneficial when the new member feels comfortable disagreeing with the rest of the team and proposing alternatives. However, newcomers may prefer to go along with the group in order to be accepted in the new team, resulting in groupthink that undermines the membership change. One way to prevent this is to decrease incentives for old-timers in the team to insist on established norms. For instance, when a new member is introduced as a "temporary" addition to a team, they are able to exert more influence than when they are introduced as a permanent addition [62]. In our experiment, we found that the assumption that teams would change

over time made people more receptive to those changes and comfortable in raising conflicting ideas. For instance, in a pilot study on designing for disabilities one new member took experiences from their previous team and explained a new viewpoint to their new team:

- [...] but I'm having a real tough time trying to tell a disabled person how they feel when I've never been in their shoes.
- + I agree that we can't tell someone how they feel. We can only empathize with them.
- I don't feel we're giving them enough credit. I don't think they want sympathy. I think they want a fair shot.
- \* So that's a good place to work from...

New members often made an effort to understand the team's thought process. Since all deliverables were available in the shared Google Drive folder, new members sometimes went back to previous phases and read the team's work. They also read previous chat logs. Most frequently, new members asked questions and engaged in conversations with other team members. These conversations helped the team re-evaluate assumptions.

*5.2.2 Language changed more within groups who experienced membership changes.* Changes in the language are a signal of changing topics, and evidence that new membership shifted the topic of discussion. Teams engaged in an average of over 7000 lines of chat over the deployment.

In the rotation between the first two phases, teams in the network rotation and network efficiency conditions had lower cosine similarity scores in language compared to teams in the control condition (Figure 7). A one-way ANOVA to compare the effect of experimental condition on the cosine similarity of text between the first and second phases had a significant effect of condition ( $F(2, 26) = 3.4, p < .05$ ). In post-hoc planned contrasts comparing network rotation and network efficiency to control, the contrast with network rotation is significant ( $p < .05$ ), and network efficiency is not (*n.s.*).

This result supports the proposed mechanism: that the addition of new members prompted integration of new perspectives and caused the team to shift ideas. Later phases showed no significant difference, suggesting that ideas may have begun to solidify by this point. However, the lack of difference in later phases despite membership change confirms that the language difference between the first two phases was not an artifact of teams welcoming new members.

*5.2.3 Psychological safety remained mostly high as teams rotated.* Despite the concerns that participants raised while rotating, psychological safety scores were relatively high and remained mostly at the same level throughout the project except for people who moved teams (Figure 8). A two-way ANOVA with status (I moved/A team member moved/Static team) and phase as independent variables, and psychological safety score as dependent variable, had a significant effect of status ( $F(2, 943) = 110, p < .001$ ) and a significant interaction effect ( $F(4, 943) = 9.8, p < .001$ ). Inspecting individual cells, the significant interaction was driven by decreasing psychological safety scores for people who were moved on later days — indicating that it felt harder to join a team later on in the process. This result can be incorporated into the algorithm to make more changes early on or to have functionality for reversing changes that are too disruptive.

## 6 FIELD DEPLOYMENT: FIREFOX ACCESSIBILITY

Following the field experiment, we sought to better understand Hive in a real world setting. So, we collaborated with Mozilla as they used Hive for a week long open design drive with accessibility advocates. To complement the prior evaluation, this effort was a field study rather than a field experiment [45]. Mozilla issued an open call via social media and over one hundred disability experts, designers, and programmers joined.



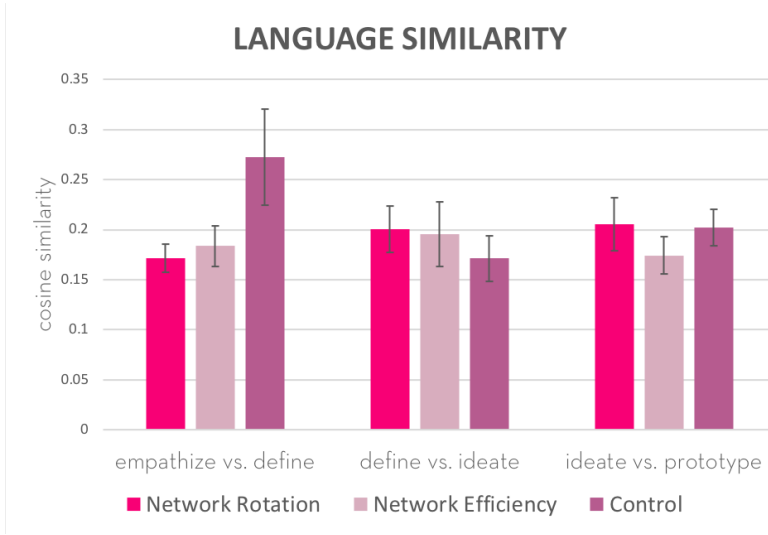


Fig. 7. Cosine similarity of the language used in teams' discussion between phases. Teams in the Network Rotation and Network Efficiency conditions shifted their discussion contents as a result of new membership, whereas teams with stable membership remained focused on the topics represented in their previous discussion.

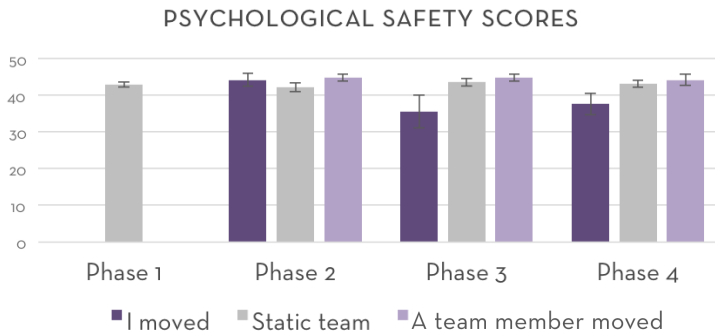


Fig. 8. Psychological safety remained high despite membership changes. In the overall study population we found no significant change in psychological safety scores. When we separated the different groups of participants we found that scores dropped for members who moved to a new team, but raised again for those members one round later, after the initial cost of reintegrating into the new team.

## 6.1 Method

An open effort to design for accessibility needs strong voices from people who have experienced the challenges firsthand and understand points of friction and limitations. Therefore, we reached out to disability communities and to disabled individuals on AMT [83], and invited them to join as team leads. Mozilla offered them a \$150 gift card for lending their expertise to the project. We created a separate channel for team leads in which we shared information and instructions. All other participants volunteered their time.

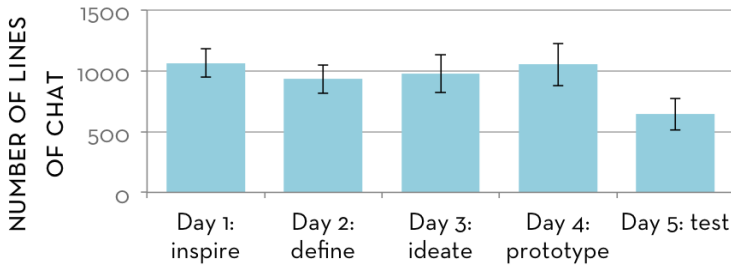


Fig. 9. Ninety-eight Mozilla volunteers completed the project in 23 teams. They exchanged 125,000 messages on Slack over the course of 5 days. Testing required the least discussion in teams.

Every participant filled out an initial form to apply to the project. We asked them for their basic demographic information, areas of expertise, and availability. Afterwards, we invited people to Slack and 113 people joined. Based on their time zone and availability, we divided the pool into five groups and placed people into teams within those groups.

Ninety-eight people completed the project in 23 teams. Of these, 42% had direct experience with disabilities and accessibility needs, 41% had design experience, and 13% had programming experience. Gender: 53% were female, 44% male, 2% preferred not to say, and 1% were gender queer. Location: 58 were from the U.S, 12 Canada, 8 India, 6 Australia, 2 Taiwan, 2 Sweden, 2 Germany, and 1 from Argentina, Brazil, France, Indonesia, Israel, the Netherlands, Nigeria, and the UK. Ages: 33% were 18-24, 26% were 25-34, 31% were 35-44, 7% were 45-54, and 3% were above 55. Over the weeklong project, 15 people (13%) dropped out due to scheduling conflicts or because they lost interest; the remaining 87% of participants completed the weeklong design drive.

The submitted designs were evaluated by an independent Mozilla accessibility panel. In addition, we sent out a final survey to participants asking for feedback on the process: “it is very important for us to hear about your experiences, the good and the bad.”. The survey consisted of the following open-ended questions:

- What did you like about the process?
- What can we do better in future design drives?
- What did you like about the membership changes?
- What do you wish we would do differently about the membership changes?

We consolidated survey responses, teams’ daily deliverables, and chat logs according to the following pre-determined themes framed by our research question: participation in a real world collective design drive, team dynamics, and the effect of network rotation on teams.

## 6.2 Results

The effort resulted in over 60 proposals, which were evaluated by an independent Mozilla accessibility panel. In this section we first detail the process that the teams went through, then we explain our qualitative findings from observing the process and analyzing survey responses.

**6.2.1 Design stages.** The phases of the design drive took place online over five days. *Day 1: Inspire.* Following OpenIDEO, the first phase “inspire” consisted mostly of storytelling and getting to know fellow team members. We asked participants to deliver a list of 10 obvious and 10 non-obvious observations that they had made in this phase. *Day 2: Define.* In this phase, we asked participants to identify patterns from the previous phase and define three different problem statements. *Day 3: Ideate.* In this phase, participants came up with 10 obvious and 10 non-obvious ideas to address their

problem statements. We told them that it is likely that their problem statements would change as they ideate. Often a single submitted idea had been reshaped by multiple members who suggested insights and brought up constraints. Participants largely relied on their own experiences to ideate solutions that would be useful.

Long day but I'm here and my grandparents just went to bed. Boy could my grand father use a lot of the ideas we came up with. I spent most of the night cleaning up his computer from all the pop ups that he's clicked on.

*Day 4: Prototype.* The prototyping phase was the most time-consuming and collaborative phase of the project. Participants chose three ideas from the previous phase. We suggested the guidelines: “the most likely to delight,” “the rational choice,” and “the most unexpected”. They worked together to develop the idea using storyboards. Teams continued to evolve their ideas — often now with a new member — and discussed users and scenarios. Some team members drew rough sketches and photographed them to demonstrate what they meant. *Day 5: Test.* In the final phase, participants found a user and gathered feedback. They submitted observations from their testing, and updated ideas and storyboards to integrate new observations. Participants used creative ways to find users for testing. Some reached out to friends and family or support groups that they were a member of. Some participants who had access to disabled family and friends offered to help others by posting on the collective's #general channel on Slack. This phase required more independent work and less collaboration than previous phases. Testing uncovered aspects of their ideas that teams had not considered before: for example, whether certain features would make browsing the web even more confusing, or what would happen when people with different needs and limitations share a computer. Participants added these observations and open questions to their proposals.

**6.2.2 Mozilla and participants reacted to the proposals positively.** While this second deployment was not focused on comparative evaluation against a control condition, it is still important to examine the quality of the final ideas. The independent Mozilla panel rated the final proposals positively, summarizing on their blog about the experience, “We were impressed with the results of this experiment,” and that Mozilla would be pursuing five of the ideas for further testing and development.

We asked participants about their views on membership changes. Of the 35 participant responses to the final survey, 17 said that they liked the opportunity to engage with new perspectives, 8 said that they enjoyed getting to work with more people, 3 said they did not like it, and 3 had mixed feelings. Participants indicated that they felt network rotation helped facilitate strong ideas, brought “fresh perspectives” and “allowed for more ideas to come into the room”:

Meeting new people was great. Mixing ideas was very helpful, especially as people dropped in and out later in the week.

I liked meeting others with different experiences and backgrounds. Everyone brought something interesting to the table.

Wow, that didn't even feel like an hour! [...] I'm loving this! This is stuff I never talk about, it's helping me realize some things about myself that I was blind to before I started talking about it here. Thank you all.

However, it is also important to capture the challenges that participants faced while participating in network rotation. Participants fretted giving up teammates with whom they had developed stronger ties:

I liked bringing in fresh ideas every day. Gave a different/new perspective [but] sometimes you build good working relationships with others and do not want to lose them.

Engaging with different backgrounds can be engaging and insightful, but also challenging [37]:

I think there was a pretty big rift between the technical thinkers and the more arts and idea-based thinkers such as myself that we could never seem to bridge properly.

Often new team members went to the effort of reading past transcripts to understand the team's status, but this was not always the case:

Sometimes it was difficult to get new members to read our previous discussions, so they didn't seem to understand where we were coming from.

The main functional request was for more control over membership changes: earlier notification, or the ability to prevent disruptive changes.

Overall, the feedback indicated that network rotation introduced positive stressors that were beneficial to the team and their ideas. Taken together, these results and observations reinforced the main conclusions of the field experiment, generalizing them to volunteers rather than paid Mechanical Turk workers.

## 7 DISCUSSION AND FUTURE WORK

Our primary objective in this paper was to create the conditions for collective design participants to understand others' viewpoints and build constructively on others' ideas by centering people and their relationships — rather than their abstract ideas. In doing so we contribute *network rotation* for organizing the collaboration through membership changes. In this section, we reflect on open questions and opportunities for our method. First, we consider the high level goal of network rotation: trading off local effectiveness with global desirability, and how those can be defined differently. Second, we analyze the conditions under which membership changes were effective. Finally, we describe the limitations of our current study and discuss design implications for collective design.

### 7.1 Trading off local effectiveness with global desirability

Our main contribution in this paper is the concept of network rotation, which restructures the social network by balancing two needs: 1) local: an effective team, and 2) global: a desirable collaboration network. Hive's network rotation algorithm provides a lever to trade off these two measures. While we offer initial ideas of what such measurements might look like based on prior literature — logistic growth with time decay for tie strength over time, and network efficiency as a measure of the collaboration network — future work will improve on them. Further, any designer could change these measures to capture their goals for what an effective group looks like, and what a desirable collaboration network looks like, and network rotation would carry them out. Below, we analyze our choices and name a few alternative concepts for these measures.

What constitutes a good team? In our evaluation we used a measure of team member familiarity [65] by averaging pairwise tie strength of all people on the team. Other measures of tie-strength and how it changes over time can also be used. For the sake of simplicity and due to the relative short time frame of our study we based our measure of tie-strength on time spent collaborating. But not all time spent together is of equal significance. Researchers have included criteria such as emotion, communication intensity, overlapping social networks, shared identity, and the sporadic nature of communications within measures of tie strength [44, 48]. Beyond tie strength, other factors for effective teams can include balancing personality types [39] or asking participants for feedback about their experiences with other team members [40, 65]. Another is user input: does the team want to participate in rotation? (An open question: how much self-insight do teams have into when it would be most helpful to introduce new blood?). Linguistic signals such as quantity, qualities, and the temporal dynamics of communications between team members [60] can also be used.

In addition to strong local collaborations, we care about the structure of the network globally. In our evaluation we aimed to encourage participants to engage with diverse viewpoints. To do so we initially randomly spread participants in the network and then aimed to maximize network efficiency, making the assumption that people with the longest path length between them were least likely to have heard each others viewpoints indirectly. This is an approximation of viewpoint diversity that we found worked well for the purpose of our evaluation (and provided a higher bar to show a significant effect). This was because participants did not join with preconceived designs for neighborhood common spaces or web accessibility opportunities. Therefore, even if participants were homogenous at first, they diverged quickly as soon as they started working together because each team came up with different ideas. We can imagine that in some other cases diversity of viewpoint may not be as easily proxied, for instance in cases of political, occupational, or socio-economic similarities among participants. In such cases, initial surveys can be used to prepopulate the network and spread people based on a stronger measure of viewpoint diversity. Beyond network efficiency, other factors may include structural diversity or the existence of internal boundary spanners in the group.

## 7.2 Boundary conditions of the effectiveness of network rotation

Given the results of our studies, what have we learned about when network rotation will produce positive results and when it might fail? We look to Tohidi et al. [73] for a useful frame: they distinguish two phases of design: “getting the right design”, early on when the idea is still malleable, and “getting the design right”, later when the focus is refinement. Qualitative evidence from our deployments suggests network rotation was most effective early on, when getting the right design. Distant ideas caused teams to reflect critically on their assumptions [49]. Participants in the Mozilla deployment reported it let them “gain new perspectives on problems” and engage with “totally new ideas and working style[s]”. Network rotation’s value diminished later, when teams focused on getting the design right. Disruptions then became costly, and participants’ suggestions involved restricting late moves (“Do it at the very beginning because it’s hard to reintroduce again with the new team”) and supporting continuity (sometimes a new member “didn’t seem to understand where we were coming from”).

In practice we found that the network rotation algorithm would sometimes move people back to a team that they had been a part of in the past, so that the algorithm could benefit from both the ties already established with the old members and the new gains in network efficiency. This is in line with prior literature that has found an increase in innovation both through the addition of newcomers and through novel combinations of old-timers [55]. While our current network rotation algorithm does not differentiate between the two, future research can analyze the trade-offs between the addition of newcomers vs. “new” old-timers.

How small can a collective be before network rotation is not helpful? And how many phases are necessary to see a meaningful effect? When the collective is small, network rotation might make similar rotation selections as a random rotation, since the network will quickly become highly efficient and all moves will have the same efficiency value. Likewise, with too few rotations, the algorithm will not be able to weave the network together effectively. To investigate this, we performed simulations to test network rotation on varying group sizes and numbers of phases. Qualitatively, we observed that network rotation required a rough lower bound on group size ( $> 50$ ) and number of phases ( $> 4$ ): otherwise its changes to network structure were not meaningfully different than random rotation.

### 7.3 Limitations

One limitation of our study is that the deployment happened over a relatively short timescale of five days, which is far briefer than regular design efforts last. The result is that our study focused on early-stage concept development and left out iteration and evolution of projects for future work.

Our regression analysis confirmed that teams in the network rotation condition outperformed those in the control (static) condition, whereas those in the network efficiency condition did not. However, our study was limited in its ability to piece out the precise mechanisms of this effect — at what value of  $\alpha$  does the effect begin to appear? A larger N study would allow us to perform a more detailed analysis with additional pairwise comparisons. Additionally, we rely on prior work to justify tie strength as an appropriate metric for local effectiveness of teams [65], but we did not provide evidence. Psychological safety or other linguistic metrics may be used as signals to demonstrate the strength of local collaborations particularly when comparing the network rotation condition with network efficiency.

In the Mozilla deployment, teams were led by people who self-identified as disabled or were care-giver to someone with a disability. However, in practice we found that there were many more decisions about the process that were made by Mozilla, the leaders of the effort who had stakes involved: who can participate? When should someone be asked to leave? How will we share the results? Further research is needed to adapt the design process for collective online design and ensure that the communities involved benefit from the process.

Finally, a note for future work: due to the complexity of the search space for network rotation, all known methods require considering many or all subsets of moves at  $O(2^N)$ , which is computationally intractable. Any solution requires an approximation, and our algorithm is one such approximation via monte carlo search. We do not claim optimality of this approach, and other approaches may exist that are equally or more effective. Future work will continue to examine this space.

### 7.4 Design Implications

Our deployments clarified avenues for improving the design of Hive and similar systems. Many analyses of human performance, from the Yerkes-Dodson Law [77] to the concept of flow [13], suggest that people perform best when facing a moderate amount of discomfort or challenge. Too easy and people zone out; too hard and they collapse under the pressure. Qualitative results made clear that network rotation introduced pressure and challenge to the design process: rotations forced people to get to know new teammates, question their previous assumptions and decisions, and potentially change their minds. However, our results also suggest that this discomfort led to better designs.

This puts Hive in an unusual position of being a design intervention that can actively create discomfort — future systems need to acknowledge this and design for it. User control remains an open question: teams wished for the ability to prevent changes. Future prototyping can answer the question of whether participants would use this ability to keep teams static when they would be better served by exposing themselves to new membership. One option would be to give each team one “token” to use throughout the process, allowing them to prevent changes once, when they deem it important, but set a norm of allowing rotation.

Given that teams were often reluctant to let a team member leave, there may exist alternative models that lessen the stress. One option would be for rotations to be sabbaticals rather than potentially permanent moves: the member would go join another team for one round, then return. Another would be to designate some members of the collective as rotation agents and not assign



them to permanent teams, instead rotating them around between teams as temporary consultants. Both options would avoid breaking up the original team but still bring in new perspectives.

Moving forward, we hope to study the effects of large scale collaborative design teams over longer periods, for instance through multiple iterations of the design process. One question is what kind of long-term effects network rotation might produce. In addition, while moving people across teams, we observed that differences in inclusion varied across teams. Future work can explore how to onboard new members.

## 7.5 Ethical Implications

We argue that the ethical implications of this technology – and team management technologies more broadly – are directly related to the levels of autonomy and power that users have within the system and within the broader social and political context that the system is embedded in. In the best case scenario, these algorithms can provide effective means for collective self-organizing toward a shared goal. In the worst case they can alienate participants. For instance, rapid membership changes may be used deliberately to break social ties among workers and stifle collective action. We propose engaging in participatory design with potential users of these algorithmic systems as one strategy for social computing researchers moving forward.

## 8 CONCLUSION

In this paper we present a system that supports collective design. To do so, we argue that large collectives cannot cross-pollinate ideas simply by placing all ideas into the commons for others to see – people in the collective must instead interact with each other closely to benefit from ideas and perspectives that each bring to the table. We define an approach, network rotation, that supports this goal by gradually weaving together team membership from across the collective to increase the efficiency of the underlying collaboration network while maintaining high tie strength within teams. We embedded this approach in a system called Hive, and showed the value of network rotation in a controlled study. We deployed Hive with a Mozilla open design drive to demonstrate how our approach works in the real world.

When collective intelligence [42] and open innovation [10] gained popularity, the initial concept was to extract value from diverse, specialized innovators who work independently. The resulting successes span from Apple to Lego to NASA [32]. Our work challenges the assumption that collective design should occur through independent work. Instead, we suggest that design at scale should be *collaborative*. Our deployment demonstrated that people were often effective at dissemination, remixing, and engagement of ideas and viewpoints. Small teams have long been a vehicle for design thinking [70]. Drawing on recent work [39, 65], we propose that future systems continue to consider large-scale design as occurring in small-N teams within large-N crowds.

## 8.1 Acknowledgments

Special thanks to Turkers and Mozilla volunteers who participated in, and helped shape this project. This work was supported by a National Science Foundation award IIS-1351131, a Hasso Plattner Design Thinking Research Program grant, and a Stanford Graduate Fellowship.



Fig. 10. Hive proposals submitted to the Mozilla accessibility design drive. a) A browser feature that provides explanations for events without parallels to the non-digital world, e.g., loading times. b) Deaf culture uses sign language call signs to adapt for more complex tasks. Here, a person using ASL can use their friend call sign to compose emails. c) Browsers react when users are having trouble, for example, when a user tries to click on a button and misses, the button increases in size.

## REFERENCES

- [1] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. 2012. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 839–848.
- [2] Karen A Bantel and Susan E Jackson. 1989. Top management and innovations in banking: Does the composition of the top team make a difference? *Strategic management journal* 10, S1 (1989), 107–124.
- [3] Michael S Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, and Eric Horvitz. 2012. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 237–246.
- [4] Osvald M Bjelland and Robert Chapman Wood. 2008. An inside view of IBM's 'Innovation Jam'. *MIT Sloan management review* 50, 1 (2008), 32.
- [5] Kevin J Boudreau and Karim R Lakhani. 2015. "Open" disclosure of innovations, incentives and follow-on reuse: Theory on processes of cumulative innovation and a field experiment in computational biology. *Research Policy* 44, 1 (2015), 4–19.
- [6] Vincent Brown, Michael Tumeo, Timothy S Larey, and Paul B Paulus. 1998. Modeling cognitive interactions during group brainstorming. *Small group research* 29, 4 (1998), 495–526.
- [7] Moira Burke and Robert E Kraut. 2014. Growing closer on facebook: changes in tie strength through social network site use. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 4187–4196.
- [8] Joel Chan, Steven Dang, and Steven P. Dow. 2016. Comparing Different Sensemaking Approaches for Large-Scale Ideation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2717–2728. <https://doi.org/10.1145/2858036.2858178>
- [9] Joel Chan, Steven Dang, and Steven P. Dow. 2016. IdeaGens: Enabling Expert Facilitation of Crowd Brainstorming. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (CSCW '16 Companion)*. ACM, New York, NY, USA, 13–16. <https://doi.org/10.1145/2818052.2874313>
- [10] Henry William Chesbrough. 2006. *Open innovation: The new imperative for creating and profiting from technology*. Harvard Business Press.
- [11] Hoon-Seok Choi and Leigh Thompson. 2005. Old wine in a new bottle: Impact of membership change on group creativity. *Organizational Behavior and human decision processes* 98, 2 (2005), 121–132.
- [12] Marie A Cini. 2001. Group newcomers: From disruption to innovation. *Group Facilitation* 3 (2001), 3.
- [13] Mihaly Csikszentmihalyi. 1996. *Flow and the psychology of discovery and invention*. New Yprk: Harper Collins (1996).
- [14] Amy C Edmondson. 1999. Psychological Safety and Learning Behavior in Work Teams. *Administrative Science Quarterly* 44, 2 (1999), 350. <https://doi.org/10.2307/2666999>
- [15] Amy C Edmondson. 2002. Managing the risk of learning: Psychological safety in work teams. *International Handbook of Organizational Teamwork*, (2002), 1–38.
- [16] Amy C Edmondson. 2013. *Teaming to innovate*. John Wiley & Sons.
- [17] John E Ettlie. 1980. Manpower flows and the innovation process. *Management Science* 26, 11 (1980), 1086–1095.
- [18] Craig L Feldbaum, Terry E Christenson, and Edgar C O'Neal. 1980. An observational study of the assimilation of the newcomer to the preschool. *Child Development* (1980), 497–507.
- [19] Herman Gadon. 1988. The newcomer and the ongoing work group. *Team building: Blueprints for productivity and satisfaction* (1988), 161–175.
- [20] DR Garrison. 2006. Online collaboration principles. *Journal of Asynchronous Learning Networks* 10, 1 (2006), 25–34.
- [21] Eric Gilbert and Karrie Karahalios. 2009. Predicting tie strength with social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 211–220.
- [22] Deborah H Gruenfeld, Elizabeth A Mannix, Katherine Y Williams, and Margaret A Neale. 1996. Group composition and decision making: How member familiarity and information distribution affect process and performance. *Organizational behavior and human decision processes* 67, 1 (1996), 1–15.
- [23] Jerold Heiss and Dennison Nash. 1967. The stranger in laboratory culture revisited. *Human Organization* 26, 1-2 (1967), 47–51.
- [24] Joshua Introne, Robert Laubacher, Gary Olson, and Thomas Malone. 2011. The Climate CoLab: Large scale model-based collaborative planning. In *Collaboration Technologies and Systems (CTS), 2011 International Conference on*. IEEE, 40–47.
- [25] Susan E Jackson. 1996. The consequences of diversity in multidisciplinary work teams. *Handbook of work group psychology* (1996), 53–75.
- [26] Farnaz Jahanbakhsh, Wai-Tat Fu, Karrie Karahalios, Darko Marinov, and Brian Bailey. 2017. You Want Me to Work with Who?: Stakeholder Perceptions of Automated Team Formation in Project-based Courses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3201–3212.
- [27] Ralph Katz and Thomas J. Allen. 1982. Investigating the Not Invented Here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R&D Project Groups. *R&D Management* 12, 1 (1982), 7–20. <https://doi.org/10.1111/j.1467-9310.1982.tb00478.x>

- [28] Joy Kim, Justin Cheng, and Michael S Bernstein. 2014. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 745–755.
- [29] Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S. Bernstein. 2017. Mechanical Novel: Crowdsourcing Complex Work Through Reflection and Revision. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 233–245. <https://doi.org/10.1145/2998181.2998196>
- [30] Mark Klein and Ana Cristina Bicharra Garcia. 2015. High-speed idea filtering with the bag of lemons. *Decision Support Systems* 78 (2015), 39–50.
- [31] Anand Kulkarni, Prayag Narula, David Rolnitzky, and Nathan Kontny. 2014. Wish: Amplifying creative ability with expert crowds. In *Second AAAI Conference on Human Computation and Crowdsourcing*.
- [32] Karim R Lakhani, Hila Lifshitz-Assaf, and Michael Tushman. 2013. Open innovation and organizational boundaries: task decomposition, knowledge distribution and the locus of innovation. *Handbook of economic organization: Integrating economic and organizational theory* (2013), 355–382.
- [33] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 467–476.
- [34] Vito Latora and Massimo Marchiori. 2001. Efficient behavior of small-world networks. *Physical review letters* 87, 19 (2001), 198701.
- [35] Diane Wei Liang, Richard Moreland, and Linda Argote. 1995. Group versus individual training and group performance: The mediating role of transactive memory. *Personality and Social Psychology Bulletin* 21, 4 (1995), 384–393.
- [36] Hila Lifshitz-Assaf. 2014. From problem solvers to solution seekers: dismantling knowledge boundaries at NASA. (2014).
- [37] Xiaodong Lin and Daniel L Schwartz. 2003. Reflection at the crossroads of cultures. *Mind, Culture, and Activity* 10, 1 (2003), 9–25.
- [38] Kurt Luther, Jari-Lee Tolentino, Wei Wu, Amy Pavel, Brian P Bailey, Maneesh Agrawala, Björn Hartmann, and Steven P Dow. 2015. Structuring, aggregating, and evaluating crowdsourced design critique. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 473–485.
- [39] Ioanna Lykourantzou, Angeliki Antoniou, Yannick Naudet, and Steven P Dow. 2016. Personality matters: Balancing for personality types leads to better outcomes for crowd teams. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 260–273.
- [40] Ioanna Lykourantzou, Robert E Kraut, and Steven P Dow. 2017. Team Dating Leads to Better Online Ad Hoc Collaborations.. In *CSCW*. 2330–2343.
- [41] Narges Mahyar, Michael R James, Michelle M Ng, Reginald A Wu, and Steven P Dow. 2018. CommunityCrit: Inviting the Public to Improve and Evaluate Urban Design Ideas through Micro-Activities. (2018).
- [42] Thomas W Malone, Robert Laubacher, and Chrysanthos Dellarocas. 2009. Harnessing crowds: Mapping the genome of collective intelligence. (2009).
- [43] Thomas W Malone, Jeffrey V Nickerson, Robert J Laubacher, Laur Hesse Fisher, Patrick De Boer, Yue Han, and Ben Towne. 2017. Putting the Pieces Back Together Again: Contest Webs for Large-Scale Problem Solving. (2017).
- [44] Peter V Marsden and Karen E Campbell. 1984. Measuring tie strength. *Social forces* 63, 2 (1984), 482–501.
- [45] E McGrath. 1995. Methodology matters: Doing research in the behavioral and social sciences. In *Readings in Human-Computer Interaction: Toward the Year 2000 (2nd ed.*
- [46] Andrés Monroy-Hernández and Mitchel Resnick. 2008. Empowering kids to create and share programmable media. *Interactions* 15, 2 (2008), 50–53.
- [47] Dennison J Nash and Alvin W Wolfe. 1957. The stranger in laboratory culture. *American Sociological Review* 22, 4 (1957), 400–405.
- [48] Henry Navarro, Giovanna Miritello, Arturo Canales, and Esteban Moro. 2017. Temporal patterns behind the strength of persistent ties. *EPJ Data Science* 6, 1 (2017), 31.
- [49] Charlan J Nemeth. 1986. Differential contributions of majority and minority influence. *Psychological review* 93, 1 (1986), 23.
- [50] Charlan Jeanne Nemeth. 1992. Minority dissent as a stimulant to group performance. *Group process and productivity* 95 (1992), 111.
- [51] Charlan Jeanne Nemeth and Barry M Staw. 1989. The tradeoffs of social control and innovation in groups and organizations. *Advances in experimental social psychology* 22 (1989), 175–210.
- [52] Jeffrey V Nickerson. 2015. Collective design: remixing and visibility. In *Design Computing and Cognition*. Springer, 263–276.
- [53] Nicole Lynn Oxley, Mary T Dzindolet, and Paul B Paulus. 1996. The effects of facilitators on the performance of brainstorming groups. *Journal of social behavior and personality* 11, 4 (1996), 633.

- [54] Paul B Paulus and Huei-Chuan Yang. 2000. Idea Generation in Groups: A Basis for Creativity in Organizations. *Organizational Behavior and Human Decision Processes* 82, 1 (2000), 76 – 87. <https://doi.org/10.1006/obhd.2000.2888>
- [55] Fabrizio Perretti and Giacomo Negro. 2007. Mixing genres and matching people: A study in innovation and team composition in Hollywood. *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 28, 5 (2007), 563–586.
- [56] Hasso Plattner. 2010. An introduction to design thinking process guide. *The Institute of Design at Stanford: Stanford* (2010).
- [57] James L Price. 1977. *The study of turnover*. Iowa State Press.
- [58] Ray Reagans, Linda Argote, and Daria Brooks. 2005. Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. *Management science* 51, 6 (2005), 869–881.
- [59] Christoph Riedl, Ivo Blohm, Jan Marco Leimeister, and Helmut Krcmar. 2010. Rating scales for collective intelligence in innovation communities: Why quick and easy decision making does not get it right. (2010).
- [60] Christoph Riedl and Anita Williams Woolley. 2017. Teams vs. Crowds: A Field Test of the Relative Contribution of Incentives, Member Ability, and Emergent Collaboration to Crowd-Based Problem Solving Performance. *Academy of Management Discoveries* 3, 4 (2017), 382–403.
- [61] Floor Rink, Aimée A Kane, Naomi Ellemers, and Gerben Van der Vegt. 2013. Team receptivity to newcomers: Five decades of evidence and future research themes. *The Academy of Management Annals* 7, 1 (2013), 247–293.
- [62] Floor A Rink and Naomi Ellemers. 2009. Temporary versus permanent group membership: How the future prospects of newcomers affect newcomer acceptance and newcomer influence. *Personality and Social Psychology Bulletin* 35, 6 (2009), 764–775.
- [63] Everett M Rogers. 2010. *Diffusion of innovations*. Simon and Schuster.
- [64] Niloufar Salehi, Lilly C Irani, Michael S Bernstein, Ali Alkhatib, Eva Ogbé, Kristy Milland, et al. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1621–1630.
- [65] Niloufar Salehi, Andrew McCabe, Melissa Valentine, and Michael Bernstein. 2017. Huddler: Convening Stable and Familiar Crowd Teams Despite Unpredictable Availability. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1700–1713. <https://doi.org/10.1145/2998181.2998300>
- [66] Pao Siangliulue, Kenneth C Arnold, Krzysztof Z Gajos, and Steven P Dow. 2015. Toward collaborative ideation at scale: Leveraging ideas from others to generate more creative and diverse ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 937–945.
- [67] Pao Siangliulue, Joel Chan, Steven P Dow, and Krzysztof Z Gajos. 2016. IdeaHound: Improving Large-scale Collaborative Ideation with Crowd-powered Real-time Semantic Modeling. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 609–624.
- [68] Pao Siangliulue, Joel Chan, Krzysztof Z Gajos, and Steven P Dow. 2015. Providing timely examples improves the quantity and quality of generated ideas. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*. ACM, 83–92.
- [69] Enno Siemsen, Aleda V Roth, Sridhar Balasubramanian, and Gopesh Anand. 2009. The influence of psychological safety and confidence in knowledge on employee knowledge sharing. *Manufacturing & Service Operations Management* 11, 3 (2009), 429–447.
- [70] Joachim Stempfle and Petra Badke-Schaub. 2002. Thinking in design teams-an analysis of team communication. *Design studies* 23, 5 (2002), 473–496.
- [71] Robert I. Sutton and Andrew Hargadon. 1996. Brainstorming Groups in Context: Effectiveness in a Product Design Firm. *Administrative Science Quarterly* 41, 4 (1996), 685–718. <http://www.jstor.org/stable/2393872>
- [72] Jerry O Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. 2009. Exploratory modeling with collaborative design spaces. *ACM Transactions on Graphics-TOG* 28, 5 (2009), 167.
- [73] Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. 2006. Getting the right design and the design right. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 1243–1252.
- [74] Helen Wauck, Yu-Chun Yen, Wai-Tat Fu, Elizabeth Gerber, Steven P Dow, and Brian P Bailey. 2017. From in the Class or in the Wild? Peers Provide Better Design Feedback Than External Crowds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.
- [75] Anbang Xu, Shih-Wen Huang, and Brian Bailey. 2014. Voyant: generating structured feedback on visual designs using a crowd of non-experts. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 1433–1444.
- [76] Anbang Xu, Huaming Rao, Steven P Dow, and Brian P Bailey. 2015. A classroom study of using crowd feedback in the iterative design process. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social*

- Computing*. ACM, 1637–1648.
- [77] Robert M Yerkes and John D Dodson. 1908. The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology* 18, 5 (1908), 459–482.
  - [78] Lixiu Yu, Aniket Kittur, and Robert E Kraut. 2014. Distributed analogical idea generation: inventing with crowds. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1245–1254.
  - [79] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Searching for Analogical Ideas with Crowds. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1225–1234. <https://doi.org/10.1145/2556288.2557378>
  - [80] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2016. Encouraging “Outside- The- Box” Thinking in Crowd Innovation Through Identifying Domains of Expertise. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1214–1222. <https://doi.org/10.1145/2818048.2820025>
  - [81] Lixiu Yu, Robert E. Kraut, and Aniket Kittur. 2016. Distributed Analogical Idea Generation with Multiple Constraints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1236–1245. <https://doi.org/10.1145/2818048.2835201>
  - [82] Lixiu Yu and Jeffrey V Nickerson. 2011. Cooks or cobblers?: crowd creativity through combination. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1393–1402.
  - [83] Kathryn Zyskowski, Meredith Ringel Morris, Jeffrey P Bigham, Mary L Gray, and Shaun K Kane. 2015. Accessible crowdwork?: Understanding the value in and challenge of microtask employment for people with disabilities. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 1682–1693.

Received April 2018; revised July 2018; accepted September 2018