4 × 3盤面の2048の完全解析

山下修平[1,a)]　金子知適[1]

概要：1 人用ゲーム 2048 はルールは単純だが高得点を獲得することは難しいゲームである．ゲームを完全解析することでプレイヤはあらゆる場面で最善手を選ぶことができるが， 2048 を完全解析することは困難である．そこで先行研究では 2048 を 3×3 盤面に縮小したゲームが完全解析された．本研究ではこれを発展させて， 4×3 盤面の 2048 の完全解析を行う．よりオリジナルの 2048 に近いゲームを完全解析することで， 2048 の持つゲームとしての性質をより詳細に明かすことができる．さらに様々な大きさの盤面の 2048 を完全解析を行い， 盤面の大きさがゲームの複雑性に指数関数的に影響を与えることを示した．

Strongly Solving 2048 on 4 × 3 Board

Shuhei Yamashita[1,a)]　Tomoyuki Kaneko[1]

Abstract: The single-player game 2048 has simple rules but is challenging to achieve a high score. Strongly solving the game would allow players to choose the best move in every situation. However, strongly solving 2048 is challenging. Therefore,in previous research, a small variant of 2048 with a 3x3 board has been solved. This paper aims to expand upon this by strongly solving another variant of 2048 on a 4x3 board. By solving a game closer to the original 2048, we can reveal the properties of 2048 in greater detail. Furthermore, by solving many variants of 2048 with various sizes, we demonstrated that the size of the board exponentially affects the complexity of the game.

1. はじめに

2048 は， Gabrirle Cirulli によって公開された 1 人用のパズルゲームである [1]．ルールは単純だが高得点を獲得することは難しく， 強化学習を中心とする多くの研究の対象となってきた．山下ら [2] は通常 4×4 の盤面上でプレイされる 2048 を， 3×3 盤面に縮小したゲームを完全解析した．本研究ではこれを発展させて 4×3 盤面の 2048 の完全解析を行う．2048 というゲームの持つ性質を調べることで， 4×4 盤面の 2048 の性質に関する知見を得ることを目指す．

2. 2048

2.1 2048 のルール

2048 は 4×4 の 16 マスの盤面上で遊ばれるゲームである．ゲームは 16 マスの内ランダムに選ばれた 2 マスに 2 か 4 の数字タイルが置かれた盤面から始まる．プレイヤの行動は上下左右いずれかの方向を選択することである．盤面上のすべての数字タイルは選択した方向に存在する数字タイルから順番に， その方向に向かってスライドして移動する．移動する数字タイルは空きマスはそのまま通過し，自分と異なる数字タイルの前か盤面の端で停止する．移動の際に 2 つの同じ数字のタイルが衝突すると， これらは合体してその合計の数字の 1 つのタイルへ変化し， プレイヤはその数値を得点として獲得する．その後， 空白のマスから等確率に選択されたある 1 マスに 90% の確率で 2 のタイルが， 10% の確率で 4 のタイルが置かれる．ゲームはプレイヤの行動と新たな数字タイルの出現を交互に繰り返して進行する．ルール上明らかなようにゲームには 2 の累乗の数字タイルしか現れない．プレイヤの一般的な目標はゲームのタイトルが示す $2^{11} = 2048$ のタイルを完成させることだが， それ以降もゲームを続けることができる．

[1] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo
[a)] yamashita-shuhei@g.ecc.u-tokyo.ac.jp

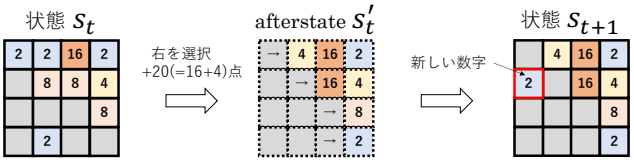

図 1　2048 の状態遷移の例

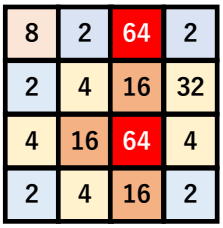

図 2　2048 の終了状態の例

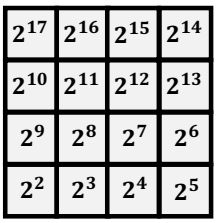

図 3　理論上到達しうる最終盤面

2.2　用語の定義とゲームのランダム性の補足

プレイヤが上下左右のいずれかの方向を選択する盤面を**状態**と呼ぶ．また盤面上の各数字タイルが選択した方向に移動し，新たな数字タイルが現れる直前の盤面を *afterstate* と呼ぶ．図 1 は，状態 s_t から afterstate s'_t を経由して次の状態 s_{t+1} へと遷移する例である．

ここである状態からある行動をとったときの afterstate への遷移の仕方は決定的である．状態 s_t から右を選択すれば必ず afterstate s'_t へと遷移する．一方で afterstate から次の状態への遷移の仕方においては，新しい数字タイルの数 (2 であるか 4 であるか) とその出現場所にランダム性が発生する．よって状態 s_{t+1} は s'_t から遷移しうる状態の 1 つであると言える．

2.3　ゲームの終了状態

プレイヤは行動の結果，状態と afterstate が異なる盤面になるような方向しか選択することができない．選択可能な行動がなくなるとゲームは終了する．図 2 はゲームが終了した盤面の例である．盤面上の数字タイルが市松模様のように配置されているため，上下左右いずれの行動を選択しても盤面は変化しない．このような盤面を終了状態と呼ぶ．またゲームの開始盤面を初期状態と呼ぶ．

一般に 2^n の数字タイルを完成させるには，盤面上に存在する 2 つの 2^{n-1} の数字タイルを合体させる必要がある．2 つの 2^{n-1} の数字タイルを完成させるには，1 つの 2^{n-1} の数字タイルと 2 つの 2^{n-2} の数字タイルを用意し，後者を合体させる必要がある．このように考えると盤面の大きさが 16 マスで 4 の数字タイルが新しく出現しうる 2048 のルールでは，2^{17} の数字タイルが理論上到達可能な最大の数字タイルであることがわかる (図 3 を参照)．またゲームは最善手を取り続けたとしても，必ず終了するゲームであることもわかる．

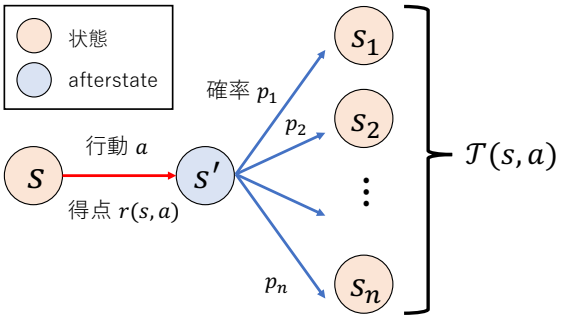

図 4　状態と afterstate の遷移図

3.　2048 の完全解析の定義

2 節で述べたように，2048 は afterstate から次の状態への遷移の際にランダム性を伴うゲームである．そのため同じ状態から毎回同じ手を選んでも最終的に得られる得点の合計は確率的に変動する．本稿ではある状態 s における最善手を「s から獲得できる得点の合計の期待値が最も高くなるような手」と定義する．またある状態から最善手を選び続けて獲得できる得点の期待値をその状態の**価値**と呼ぶ．終了状態はそれ以降得点を獲得することができないため，価値は 0 点である．よってゲーム木の末端である終了状態から後退解析を行うことで，すべての状態の価値を計算することができる．

状態 s の価値を $V(s)$，状態 s で行動 a をとった際に獲得する得点を $r(s,a)$，状態 s から行動 a をとったときに遷移しうる次の状態の集合を $\mathcal{T}(s,a)$ とする (図 4 を参照)．たとえば図 1 において，$r(s_t, \text{右}) = 20$，$s_{t+1} \in \mathcal{T}(s_t, \text{右})$ である．

任意の $s_{\text{next}} \in \mathcal{T}(s,a)$ について $V(s_{\text{next}})$ がすでに計算済みであれば，$V(s)$ は式 1 に従って計算できる．

$$V(s) = \begin{cases} 0 & (s \text{ が終了状態}) \\ \max_a \left(r(s,a) + \mathbb{E}_{s_{\text{next}} \in \mathcal{T}(s,a)} V(s_{\text{next}}) \right) & (\text{otherwise}) \end{cases} \tag{1}$$

式 1 において $\mathbb{E}_{s_{\text{next}} \in \mathcal{T}(s,a)} V(s_{\text{next}})$ は，状態 s から行動 a をとって決定的に遷移する afterstate の価値と解釈できる．またこのとき状態 s における最善手は $\arg\max_a \left(r(s,a) + \mathbb{E}_{s_{\text{next}} \in \mathcal{T}(s,a)} V(s_{\text{next}}) \right)$ である．

ゲーム木のすべての状態を列挙し，それらの価値を式 1 に従って計算することで任意の状態で最善手を選択することができる．これを 2048 における完全解析とする．

4.　先行研究 (3×3 盤面の 2048 の完全解析)

山下ら [2] は一般に 4×4 盤面上で行われる 2048 を，ルールはそのままで盤面を 3×3 に縮小したゲームを提案し，これを完全解析した．以下では文献 [2] で示された完全解析の方法とその解析結果を簡潔に示す．5 節では 4×3

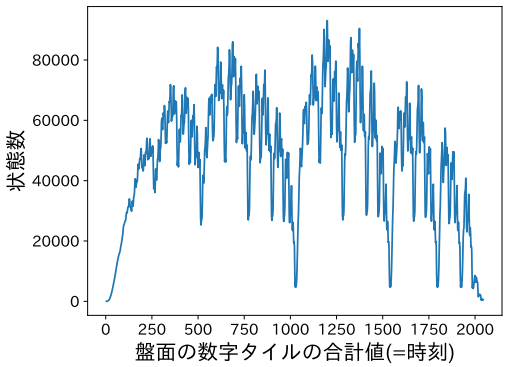

図 5 3×3 盤面の 2048 の時刻と状態数のグラフ

盤面の 2048 を完全解析するにあたって， 本研究が加えて行った工夫を述べる．

4.1 完全解析の方法

3 節で述べたように， まずゲームに現れうるすべての状態を列挙する．これはキューを用いた幅優先探索によって行う．まずキューには初期状態を入れて探索を開始する．キューの先頭から状態 s を取り出し， s から遷移可能な状態の集合 $\mathcal{T}(s,a)$ を列挙する．$s_{\text{next}} \in \mathcal{T}(s,a)$ について，s_{next} が未発見の状態であればキューの末尾に追加する．これをキューが空になるまで繰り返し行うことですべての状態を列挙できる．

次に列挙した状態の価値を式 1 に従って計算する．価値が未計算の状態 s について， 任意の $s_{\text{next}} \in \mathcal{T}(s,a)$ の状態の価値が計算済みであれば $V(s)$ を計算する．ある $s_{\text{next}} \in \mathcal{T}(s,a)$ の価値が未計算であれば， s の価値計算は一旦保留して別の状態の価値を計算する．最終的にすべての状態の価値を計算することができる．

4.2 解析結果と課題

結果としてゲームに現れうる状態は全部で $48,713,519$ 通りあり， ゲームの初期状態の価値は約 $5,469$ 点であることが示された．図 5 に 5.1 節で導入する時刻 (盤面の数字タイルの合計値) と状態数のグラフを示す．5.2 節では 4×3 盤面の結果と比較を行う．

文献 [2] の方法ではゲーム木を整理して管理していなかった．そのため幅優先探索で現れた状態がすでに発見済みかどうかを確認するためには， これまでに発見した状態の集合をすべてメモリにのせておく必要がある．これは 3×3 盤面の場合は状態数が小さいため， 1 ギガバイト程度のメモリがあれば実行可能だった．しかし盤面が大きくなり，状態数が大きくなるとこれは問題となる．

また価値計算では次にどの状態の価値が計算可能であるか予め分からないため， 効率よく計算を行うことができない．そのため列挙の実行は数分で終わるのに対して， 価値計算の実行には約 5 時間を要したと報告されている．

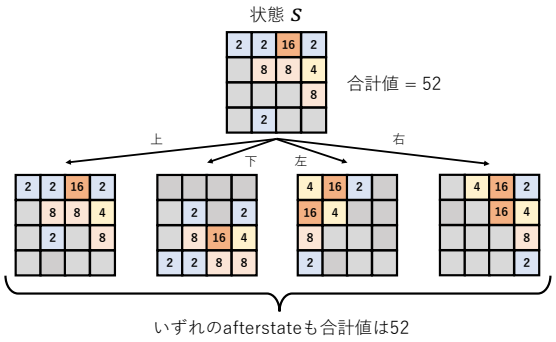

図 6 状態から afterstate への遷移における盤面の合計値の不変性

これらを踏まえると， そのままの方法では盤面サイズが大きな 2048 を完全解析するのは必要なメモリ量と実行速度の観点から難しい．

5. 4×3 盤面の 2048 の完全解析

本研究では 2048 のゲームとしての性質をより詳細に明かすことを目標に 4×3 盤面の 2048 の完全解析を行う．1 つのマスについて 10 通りの数字が現れうると大雑把に仮定すると， 4×3 盤面の 2048 は 3×3 盤面の 2048 に比べて 3 マス多いため状態数は少なく見積もっても $1,000$ 倍以上あると予想される．そのため 4.2 節の議論を踏まえると，文献 [2] の解析方法と全く同じ方針では実行時間とメモリ量の観点から困難である．そこで本研究では新たに盤面上の数字タイルの合計値をゲームの**時刻**とする考え方を導入し， これを解決する．

5.1 盤面の時刻の導入

2048 はゲームの性質上， 状態から afterstate への遷移においては盤面の数字タイルの合計値は変化しない．状態から afterstate への遷移時に数字タイルの合体が起きない場合は， 当然盤面の数字タイルの合計値はそのままである．数字タイルの合体が起きる場合も， 合体した数字タイルは合体前の 2 つの数字タイルの合計値であるから盤面の数字タイルの合計値は変化しない (図 6 を参照)．

盤面の数字タイルの合計値が変化するのは， afterstate から次の状態への遷移において新たにタイルが出現するときである．新たな数字タイルの数字分だけ合計値は増加する．そのため盤面の数字タイルはゲームが 1 ステップ進むごとに必ず 2 か 4 ずつ増加し， 減少することはない．

よって盤面上の数字タイルの合計値を**時刻**と呼ぶことにすると， ゲーム木を時刻に従って整理することができる．時刻 t の状態は必ず時刻 $t+2$ か時刻 $t+4$ の状態にしか遷移しない．そのため時刻 t から遷移しうる状態を列挙する際には， 時刻 $t, t+2, t+4$ の状態以外はディスクに保存することができる．

また時刻 t の状態は時刻 $t+2$ と $t+4$ の状態の価値が計算済みであれば， 必ず式 1 に従って計算できる．よって

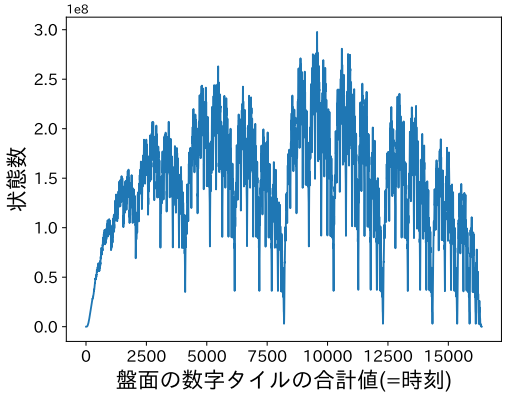

図 7 4 × 3 盤面の 2048 の時刻と状態数のグラフ

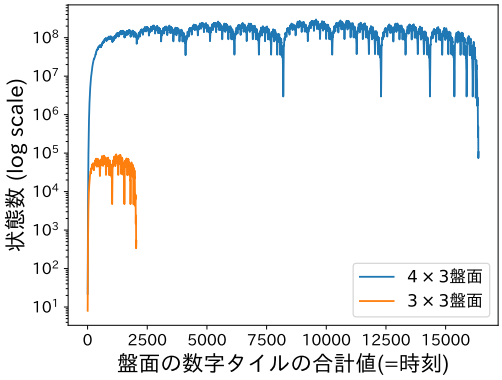

図 8 4 × 3 盤面と 3 × 3 盤面の状態数の比較

時刻が最大の状態から順番に価値を計算することで後退解析を効率的に行える．これによってプログラムに必要なメモリの問題と価値計算にかかる時間の問題が大きく改善された．

5.2 解析結果

5.1 節で述べた時刻を導入した上で， 3 × 3 盤面の 2048 の完全解析と同様にすべての状態の列挙とそれらの価値計算を行う．なお， ある盤面について回転・反転に関して対称な 4 つの盤面をまとめて 1 つの状態として扱う．実装に関する詳細は付録 A.1 節を参照されたい．

4 × 3 盤面の 2048 の状態数は全部で 1, 152, 817, 492, 752 通りあることがわかった．時刻と状態数の推移グラフを図 7 に示す．図 5 の 3 × 3 盤面の 2048 の解析のグラフと同じような増減の繰り返しをすることが見て取れる．これは大きな数字タイルを完成させる直前の盤面は多くの数字タイルで埋まるため状態数が少ないが， 大きな数字タイルを完成させると空白が多くなり状態数も多くなるためだと考えられる．さらに縦軸である状態数を対数スケールにして， 3 × 3 盤面の状態数と比較したグラフを図 8 に示す．各時刻の状態数だけでも 10^3 倍程度の差があり， 盤面を 3 マス追加するだけでゲーム木のサイズが非常に大きくなったことがわかる．4 × 4 盤面の 2048 も図 5 や図 7 のような状態数の推移をたどると考えられるが， その大きさは莫大なものであることも予想される．

また計算の結果， 初期状態の価値は約 50, 724 点であった．これは 6 節で他の大きさの盤面の 2048 の解析結果とともに考察する．

6. 様々な大きさの盤面の 2048 の完全解析

本研究では 4 × 3 盤面の 2048 に加えて， 様々な大きさの盤面の 2048 の完全解析を行った．2 × 2 盤面から 4 × 3 盤面の 2048 の完全解析を行った結果を表 1 に示す．

表 1 盤面の大きさと解析結果の表

盤面サイズ	状態数	初期状態の価値
$2 \times 2 = 4$	176	67.6
$3 \times 2 = 6$	21, 752	480.9
$4 \times 2 = 8$	4, 980, 767	2642.6
$3 \times 3 = 9$	48, 713, 519	5469.2
$4 \times 3 = 12$	1, 152, 817, 492, 752	50, 724.2

表から分かるように， 盤面サイズが増えるに従って指数関数的に状態数は増大する．4 × 3 盤面よりも 4 マス大きい 4 × 4 盤面の 2048 は， 4 × 3 盤面の 10, 000 倍以上もの状態数を持つことが予想される．

また盤面サイズが小さな 2048 では初期状態の価値が $2^{\text{盤面サイズ}+2}$ 程度である．一方で盤面サイズが 4 × 2 以上の 2048 では初期状態の価値が $2^{\text{盤面サイズ}+3}$ 以上であるという傾向が見られる．特に解析したもののうち最も盤面が大きな 4 × 3 盤面の 2048 の価値は， $2^{\text{盤面サイズ}+3} + 2^{\text{盤面サイズ}+2}$ 程度である．この傾向を踏まえると， 4 × 4 盤面の 2048 の初期状態の価値は $2^{16+3} + 2^{16+2} = 786, 432$ 点以上であることが予想される．現状最も有力な強化学習手法の 1 つである Stochastic MuZero [3] も 50〜60 万点程度の性能であることを踏まえると， より高得点を獲得できるエージェントが開発されることが期待できるだろう．

7. まとめと 4 × 4 盤面の 2048 の完全解析に関する議論

本研究では先行研究の 3 × 3 盤面の 2048 の完全解析を発展させて， 4 × 3 盤面の 2048 の完全解析を行った．解析を行うにあたって盤面の数字タイルの合計値をゲームの時刻とし， 時刻に従ってゲーム木を整理することを提案した．これによってプログラムが必要とするメモリ量と価値計算にかかる時間を， 先行研究と比べて大幅に少なくすることに成功した．

また様々な大きさの盤面の 2048 を完全解析することで，2048 というゲームが一般に持つ性質を明らかにすることを試みた．

最終的にはオリジナルの 4 × 4 盤面の 2048 の完全解析を行えることが理想だが， 現状困難な課題であると考えられる．その理由としてまず莫大な記憶容量を必要とする点

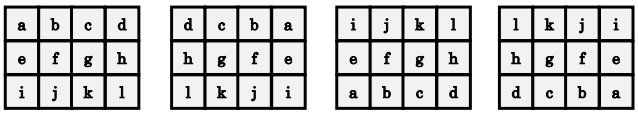

図 A·1 回転・反転に関して対称な盤面

が挙げられる．計算速度の観点からも，4×3 盤面の 2048 の解析に全部で約 2 ヶ月かかったことを踏まえると，現実的な時間に解析を終わらせることは難しいと考えられる．A.1.3 節で述べるようにスケーラビリティの良い手法が見つからない限り，解析を終わらせることは困難である．

一方で，4×3 盤面の 2048 はそれ自体が状態数が 1 兆を超えるゲームである．先行研究 [2] では 3×3 盤面の 2048 を完全解析し，強化学習のベンチマークとして用いることが提案された．4×3 盤面の 2048 は 3×3 盤面の 2048 と比べて $10,000$ 倍以上の状態数を持つ複雑なゲームであるにも関わらず，本研究で完全解析を行うことができた．そのため強化学習のベンチマークとしても，3×3 盤面の 2048 よりも精密に様々な手法の良し悪しを判定することができると考えられる．完全解析に成功した 4×3 盤面の 2048 を題材として，強化学習手法の比較等の研究を行うことを今後の課題とする．

付　　録

A.1 実装の詳細

A.1.1 実験条件

本研究では 256GB のメモリを持ち，プロセッサとして 32 コアで動作する AMD Ryzen Threadripper 3970X を搭載するマシンを用いた．プログラムの実装には C++17 を用いた．状態列挙には約 45 日，価値計算には約 20 日を要したが，これは実装の細かい工夫の仕方で改善することができると考えられる．また列挙した状態とその価値を保持するために合計で 16.8TB の記憶領域を必要とした．

A.1.2 状態の表現

4×3 盤面の 2048 の盤面の各マスは，空白であるか 2^1〜2^{13} のいずれかの数字タイルが置かれているという 14 通りが考えられる．よって各マスは 4 ビットあれば表現可能であり，12 マスある盤面全体は 1 つの 64 ビット整数で表現することができる．またある盤面の回転・反転に関して対称な盤面をまとめて 1 つの状態として同一視している (図 A·1 を参照).

A.1.3 並列化とスケーラビリティ

状態の列挙と価値計算のいずれのステップにおいても並列化を施したプログラムを用いた．

時刻 t の状態から時刻 $t+2$ の状態の列挙を以下の方法で並列化した．まず時刻 t の状態の集合を各スレッドに分配し，それぞれのスレッドが与えられた状態から遷移可能な時刻 $t+2$ の状態を列挙する．列挙された状態にはスレッド同士で重複があるため，この重複をシングルスレッドで取り除く必要がある．スレッドを多くするほど各スレッドの仕事は小さくなるが，重複を取り除くのに時間がかかる．そのため状態列挙において並列化の効果は大きくはなく，本研究で用いたプログラムでは 8 スレッドを用いてシングルスレッドの 2.5 倍程度の高速化にとどまった．

また時刻 t の状態の価値計算は以下の方法で並列化した．まず計算済みである時刻 $t+2$ と $t+4$ の状態の価値は，それぞれが 1 つのハッシュテーブルに記録して用意されているとする．時刻 t の状態を各スレッドに分配し，それぞれのスレッドが与えられた状態の価値を時刻 $t+2$ と $t+4$ の状態価値のハッシュテーブルを参照して計算する．この際に各スレッドは自身が持つ固有のハッシュテーブルに計算した時刻 t の状態の価値を記録する．最後に各スレッドが持つ時刻 t の状態価値を記録したハッシュテーブルを 1 つに統合する．これを時刻 $t-2$ の状態価値計算の際に参照するハッシュテーブルとして用いる．ここではハッシュテーブルの統合に時間を要するという問題があり，30 スレッドを用いてシングルスレッドの 5 倍程度の高速化にとどまった．

4×4 盤面の 2048 を完全解析するにはよりスケーラビリティの良い手法を考案する必要があるだろう．

参考文献

[1] Cirulli, G.: 2048 (2014). http://gabrielecirulli.github.io/2048/.

[2] 山下修平，金子知適，中屋敷太一：3 × 3 盤面の 2048 の完全解析と強化学習の研究，GPW2022 論文集，Vol. 2022, pp. 1–8 (2022).

[3] Antonoglou, I. et al.: Planning in Stochastic Environments with a Learned Model, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=X6D9bAHhBQ1⟩ (2022).