arXiv:2601.09289v1 [cs.DS] 14 Jan 2026

Swish 的计算复杂性

Takashi Horiyama
北海道大学信息科学与技术学院，日本
Takehiro Ito
东北大学信息科学研究生院，日本

Jun Kawahara
京都大学信息学研究生院，日本
Shin-ichi Minato
京都大学信息学研究生院，
日本
Akira Suzuki
东北大学数据驱动科学与人工智能中心，日本
Ryuhei Uehara
日本先进科学技术研究所信息科学学院，日本
Yutaro Yamaguchi
大阪大学信息科学与技术研究生院，日本

摘要

Swish 是一种纸牌游戏，玩家拿到带有符号（圆环和球）的卡片，并寻找卡片的有效叠放方式，称为一次“Swish”。Dailly、Lafourcade 和 Marcadet（FUN 2024）研究了广义 Swish，并证明当每张卡片只有一个符号时，该问题可在多项式时间内求解；而每张卡片有三个或更多符号时，该问题为 NP 完全。本文解决了先前尚未解决的每张卡片有两个符号的情形，这对应于原始游戏。我们证明在这种情形下，Swish 是 NP 完全的。具体而言，我们证明，当卡片允许的变换仅限于单次（水平或垂直）翻转或 180 度旋转时，该问题为 NP 难；并将结果推广至允许全部三种变换的原始设定。相比之下，当不允许任何变换时，我们给出一种多项式时间算法。结合已有结果和本文结果，我们完整刻画了 Swish 的计算复杂性与每张卡片的符号数量及允许的变换之间的关系。

2012 年 ACM 学科分类 计算理论 →问题、归约与完全性；计算理论 → 算法设计与分析；计算理论 → 图算法分析

关键词与短语 Swish、计算复杂性、匹配、奇偶约束环

资助 一般：日本学术振兴会科研费项目编号 JP20H00605、JP24H00690、JP25H01114。*Takashi Horiyama*：日本学术振兴会科研费项目编号 JP23K24806。*Takehiro Ito*：日本学术振兴会科研费项目编号 JP24H00686。*Akira Suzuki*：日本学术振兴会科研费项目编号 JP25K14980。*Yutaro Yamaguchi*：JST CRONOS 日本项目编号 JPMJCS24K2。

1 引言

Swish [15] 是一种纸牌游戏，玩家要尝试找出有效的卡片叠放方式。游戏中，卡片相互叠放，使每个圆环（圆圈）◯ 都与一个圆点（点）●相遇，每个圆点都与一个圆环相遇，并且任意两个圆环或任意两个圆点都不重叠；这样的配置称为*Swish*。商业版本的游戏包含60张透明卡片，每张卡片上恰好有一个圆环和一个圆点，它们排列在一个高4、宽3的网格中（示例见图1）。卡片可以水平或垂直翻转，也可以旋转180度；计入对称后，共有36种不同的卡片，因此有些（等价的）卡片会在牌组中出现两次

2 Swish 的计算复杂性

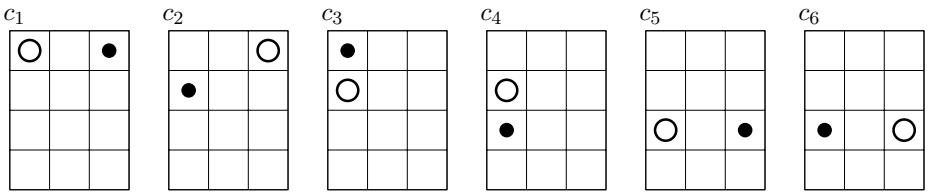


图1 Swish 中卡片的示例。采用第 2.1 节引入的记号，$c_1 = ((1, 1), (1, 3))$、$c_2 = ((1, 3), (2, 1))$ 等。该实例在不进行旋转或翻转的情况下，包含以下 Swish：$\{c_1, c_2, c_3\}$、$\{c_5, c_6\}$ 和 $\{c_1, c_2, c_3, c_5, c_6\}$。如果允许翻转和旋转，那么 $\{c_2, c_3, c_6\}$ 也构成一个 Swish，其中卡片分别为 $c_2 = ((1, 3), (2, 1))$、$c_3^{\mathrm{F}} = ((2, 3), (1, 3))$ 和 $c_6^{\mathrm{R}} = ((2, 1), (2, 3))$；上标 F 和 R 分别表示水平翻转和 180 度旋转。

。商业游戏中，圆环和圆点分别着色为蓝色、橙色、紫色和绿色，但这些颜色仅用于直观地区分位置。从组合学视角来看，这个游戏可以视为单色的，也就是说，符号本身没有颜色。每个回合中，玩家会从牌组中抽出16张公共卡片，并竞相比对手更快找出一个 S wish，然后喊出“Swish！”。尽管规则简单，Swish 却具有丰富的组合结构，因此成为数学和算法分析中一个引人入胜的研究对象。

Rowland [12]开创了对 Swish 的数学研究，重点关注商业版牌组的结构和概率方面。她运用伯恩赛德引理证明，60 张卡片的牌组在对称意义下包含 36 张不同的卡片。她还分析了符号位置的分布，指出这些位置并非均匀出现，并研究了用两张卡片组成一次 S wish 的概率。此外，她还通过矩阵表示，从线性代数视角探讨了 Swish。

从计算复杂性的角度来看，Dailly、Lafourcade 和 Marcadet [3] 于 FUN 2024 首次对 Swish 进行了系统研究。他们引入了广义 Swish，并分析了其复杂性与每张卡片上的符号数量之间的关系。他们证明，当每张卡片只包含一个符号时，Swish 可以在多项式时间内求解；而当卡片上可能包含三个或更多符号时，该问题则变为 NP 完全。为证明 NP 完全性，他们还研究了一种受限变体：卡片不能翻转或旋转，这种变体通常称为 Simple-Swish。

本文通过解决两个尚未解决的问题，完成了 Swish 的复杂性分类。第一个问题涉及每张卡片上的符号数量：当每张卡片恰好包含两个符号（如原版游戏中那样）时，Swish 的复杂性如何？第二个问题涉及几何变换的作用：选择性地允许或禁止翻转和旋转，会如何影响复杂性？

我们证明了一个明确的分界。当不允许翻转或旋转时，含有每张卡片两个符号的 Swish 存在多项式时间算法（定理 1）。相比之下，只要允许一种变换（即水平翻转、垂直翻转和 180 度旋转中的任意一种），即使每张卡片只有一个圆环和一个球，该问题也会变为 NP 完全（定理 2）。因此，即使卡片上可以包含更多符号，在这些设置下 Swish 仍然是 NP 完全的。结合先前的研究结果，这就完整刻画了 Swish 关于每张卡片上的符号数量及允许的对称变换的计算复杂性。

本文其余部分的组织如下。在第2节中，我们给出定义并将问题形式化。在第3节中，我们陈述研究结果，并分别在

各小节中加以证明。在第4节中，我们通过对未来工作的展望总结全文。

2 预备知识

2.1 Swish

我们根据卡片的大小，对 Swish 的一种推广进行形式化定义。固定一个 $h \times w$ 整数网格，其中 (i, j) $(1 \leq i \leq h,\ 1 \leq j \leq w)$ 表示从上方数第 i 个、从左侧数第 j 个的点；在原始 Swish 中，$h = 4$ 和 $w = 3$。由于我们主要关注每张卡片恰好有一个圆环和一个球的情形，因此每张卡片 c 都由网格上的两个点构成的有序对 $((i_1, j_1), (i_2, j_2))$ 指定，这表示 c 在 (i_1, j_1) 处有一个圆环Ⓞ，在 (i_2, j_2) 处有一个球 ●。如果允许旋转卡片 $c = ((i_1, j_1), (i_2, j_2))$，则可以将$c$用作卡片 $c^{\mathrm{R}} := ((h - i_1 + 1, w - j_1 + 1), (h - i_2 + 1, w - j_2 + 1))$；此处即使 $h = w$，也仅允许 180 度旋转。如果允许翻转 c，则可以将 c 用作卡片$c^{\mathrm{F}} := ((i_1, w - j_1 + 1), (i_2, w - j_2 + 1))$；此处基于对称性，仅允许水平翻转（因为同时允许水平和垂直翻转时，两者组合会产生 180 度旋转）。如果允许同时旋转和翻转 c，除了 c^{R} 和 c^{F}之外，还可以将 c 用作卡片 $c^{\mathrm{RF}} = ((h - i_1 + 1, j_1), (h - i_2 + 1, j_2))$；该卡片与将 c垂直翻转后得到的卡片相同。

假设给定一组卡片。如果允许翻转或/和旋转$\mathcal{S}$ 中的某些卡片后，子集$\mathcal{S} \subseteq \mathcal{C}$ 中的卡片满足以下条件，则称为一个 *Swish*：对于网格上的每个点 p，要么

- $\mathcal{S}$ 中没有卡片带有圆环，且$\mathcal{S}$ 中没有卡片在 p 处带有球；要么
- $\mathcal{S}$ 中恰有一张卡片带有圆环，且$\mathcal{S}$ 中恰有一张卡片在 p 处带有球。

示例请再次参见图1。

原游戏的目标是找到一个规模较大的 Swish。于是，我们可以考虑如下自然的判定问题：

Swish
输入： 网格上有一组卡片$\mathcal{C}$ 以及一个整数 k。**目标：** 判断是否存在一个 Swish $\mathcal{S} \subseteq \mathcal{C}$，满足 $|\mathcal{S}| \geq k$，其中允许翻转和旋转任意卡片。

不翻转的 Swish
输入： 网格上有一组卡片$\mathcal{C}$ 以及一个整数 k。**目标：** 判断是否存在一个 Swish $\mathcal{S} \subseteq \mathcal{C}$，满足 $|\mathcal{S}| \geq k$，其中允许旋转任意卡片，但不允许翻转卡片。

不旋转的 Swish
输入： 网格上有一组卡片$\mathcal{C}$ 以及一个整数 k。**目标：** 判断是否存在一个 Swish $\mathcal{S} \subseteq \mathcal{C}$，满足 $|\mathcal{S}| \geq k$，其中允许翻转任意卡片，但不允许旋转卡片。

不翻转或旋转的 Swish**输入：** 网格上的一组卡片 $\mathcal{C}$ 以及一个整数 k。**目标：** 判断是否存在一个 Swish $\mathcal{S} \subseteq \mathcal{C}$ 使得 $|\mathcal{S}| \geq k$，其中不允许翻转或旋转任何卡片。

2.2 图

我们使用图论中的标准术语和记号。两个顶点u 和 v 之间的无向边记作无序对 $\{u,\ v\}$，从 u 到 v 的有向边记作有序对 $(u,\ v)$。

在图（无向图或有向图）中，匹配是指满足以下条件的边子集：每个顶点至多有一条关联边属于该子集。如果所有顶点都恰好有一条关联边，则称该匹配为完美匹配。为每条边赋予一个权重值后，边子集的权重定义为其中各边权重之和。

在有向图中，顶点 v 的入度是指指向 v的边数，而出度是指从 v出发的边数。有向环是一个连通的有向图，其中每个顶点的入度和出度都恰好为 1。有向环的长度是其中顶点数（或等价地，边数）。

3 算法与复杂性

本文的主要结果如下，解决了 [3], 中的一个开放问题：

▶**定理 1.** 不翻转或旋转的 *Swish* 可在多项式时间内求解。

▶**定理 2.** *Swish*、不翻转的 *Swish* 和不旋转的 *Swish* 均为 *NP* 完全问题。

请注意，所有这些问题显然都属于 NP，因为如果给定一个 Swish $\mathcal{S}$，并附上如何使用 $\mathcal{S}$ 中每张卡片的额外信息（是否翻转和/或旋转），我们就可以在多项式时间内轻松验证其有效性。

3.1 不翻转或旋转的 Swish 的多项式时间算法

本节中，我们证明定理 1。

设 $(\mathcal{C}\ k)$ 是不翻转或旋转的 Swish 的一个实例，即，$\mathcal{C}$ 是一个卡片集合，且 k 是一个整数。我们根据 $\mathcal{C}$ 构造一个带边权重 w 的二分图 $G = (V^+, V^-; E)$，如下所示（见图 2）。

首先，令 V为网格上的点集。对于每个点 $p \in V$，我们创建两个副本p^+和 p^-。令 $V^+ := \{\, p^+ \mid p \in V\}$和 $V^- := \{\ p^- \mid p \in V\}$。

接下来，对于每张卡片 $c = (p, q) \in \mathcal{C}$，我们创建一条边 $e_c = \{p^+, q^-\}$，其权重为 $w(e_c) = 1$。此外，对于每个点 $p \in V$，我们创建一条边 $f_p = \{p^+, p^-\}$，其权重为 $w(f_p) = 0$。令 E为创建的边集。

众所周知，可以在多项式时间内计算二分图中的最大权完美匹配 [8], ，因此以下断言即可完成证明。

▷ 断言 3. 存在一个 Swish $\mathcal{S} \subseteq \mathcal{C}$，使得 $|S| = k$ 无需翻转或旋转任何卡片，当且仅当（G, w）具有权重为 k的完美匹配。

证明。设 S ⊆C 是一个不翻转或旋转任何卡片的 swish。我们在 G 中构造一个完美匹配 $M \subseteq E$，其权重为 $w(M) = k$。对于每张卡片 $c = (p, q) \in$ S，我们向 M 添加权重为 1 的边 $e_c = \{p^+, q^-\}$。于是，由于 S 是一个不翻转或旋转的 swish，对于每个点 $p \in V$，以下情况之一成立：

- 要么 p^+ 和 p^- 都没有当前 M的关联边，要么 p^+ 和 p^- 都恰
- 好有一条当前 M的关联边。

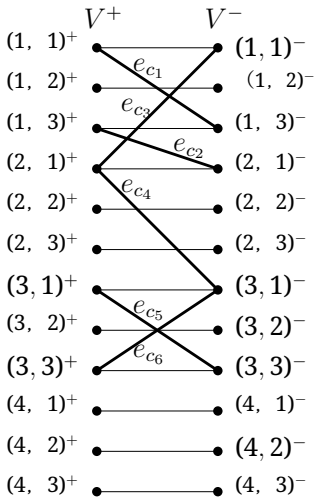


图2 由图1中的 Swish 实例构造的二分图 $G = (V^+, V^-; E)$。$\{e_{c_1}, e_{c_2}, e_{c_3}\} \cup \{f_p = \{p^+, p^-\} \mid p \in V \setminus \{(1,1), (1,3), (2,1)\}\}$ 是一个完美匹配，它对应于不旋转或翻转的 Swish $\{c_1, c_2, c_3\}$。该完美匹配的权重为 3，等于对应 Swish 的大小。

因此，对于前一种情况中的每个点 p，我们都可以通过添加权重为 0 的边 f_p，使 M 成为完美匹配。所得完美匹配的权重显然为 $|S| = k$。

反过来，设 $M \subseteq E$ 是 G 中具有 $w(M) = k$的一个完美匹配。我们构造一个 Swish $S \subseteq C$，使得无需翻面或旋转任何卡片即可得到 $|S| = k$。令 $S := \{c \in C \mid e_c \in M\}$。根据边权重 w的定义，有 $|S| = w(M) = k$。由于 M 是一个完美匹配，且每条边 $f_p \in M$ 连接来自同一点 $p \in V$的两个顶点 p^+ 和 p^-，如果其中一个 p^+ 被某条边 $e_c \in M$覆盖，则另一个 p^- 也被某条边$e_{c'} \in M$覆盖。这意味着对于每个点 $p \in V$，要么 S 中没有卡片在 p 处带有圆环，且没有卡片在 p处带有球；要么 S 中恰有一张卡片在 p处带有圆环，且恰有一张卡片在 p处带有球。因此，S 是一个不翻转或旋转的 Swish，证明完成。

◀

▶ 备注 4。计算时间上界为 $O(n^3)$ 或 $O((m+n)n \log n)$ [5,8]，其中$n := |V| = hw$ 且 m 是输入卡片的数量。这里，我们可以假设 h 和 w 均有 $O(m)$ 的上界，因为不含圆环或球的行或列都可以移除，且不失一般性。因此，该算法的时间复杂度关于输入规模是多项式的。

▶ 备注 5。我们可以考虑更一般的不翻转或旋转的 Swish。假设我们想要找到大小恰好为 k 的 Swish。那么，同样的归约会得到所谓的精确匹配 问题，在该问题中，我们需要找到权重恰好为 k的完美匹配。对于这个问题，众所周知存在一种随机化 多项式时间算法 [10]，但自该问题出现以来的 40 多年里，是否存在确定性 多项式时间算法一直是理论计算机科学中的一个重要开放问题 [11]。

作为另一种推广，假设每张输入卡片要么没有圆环，但恰好有两个位于不同位置的球，要么没有球，但恰好有两个位于不同位置的圆环。在这种情况下，归约和证明也几乎相同（例如，如果卡片 c 在点 p 和 q处有两个圆环，那么我们就创建一条边 $e_c = \{p^+, q^+\}$，其权重为 $w(e_c) = 1$），但所得图不再是二部图。尽管如此，已知可以在多项式时间内计算最大权完美匹配（确定性算法） c，并且可以在多项式时间内计算精确权重完美匹配（随机化算法） p。

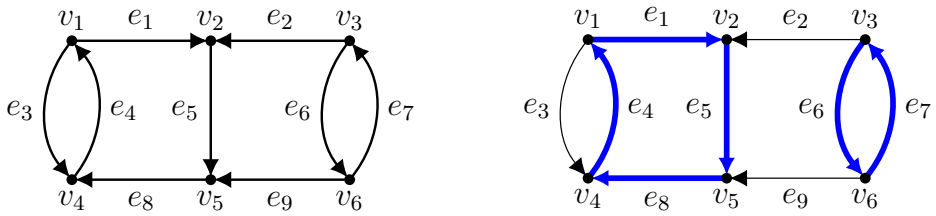


图 3 ∀偶数有向循环因子实例（左）及其解（右）的示例。

已知可以在多项式时间内计算匹配（确定性算法） [4]，，并且可以在多项式时间内计算精确权重完美匹配（随机化算法） [10]。在这两种情况下，已知判定问题都可以在 O(n^ω) 时间内（随机化地） [7, 13, 14] 解决；而在后一种（精确权重）情况下，要找到解本身还需要额外的 n 因子，其中$\omega \leq 2.371339$ 是矩阵乘法指数 [1]。

3.2 不旋转的 Swish 的 NP 困难性

本节证明不旋转的 Swish 是 NP 难的。下一节将基于此证明完成定理 2 的证明。

我们将归约以下已知为 NP 完全的问题。对于一个有向图 $G = (V,\ E)$，若边子集 $F \subseteq E$ 覆盖 V 的所有顶点，且 F 的每个连通分量都是长度为偶数的有向环（见图 3），则称其为偶数有向循环因子。

∀偶数有向循环因子**输入：** 一个有向图 $G = (V,\ E)$。**目标：** 判定是否存在偶数有向循环因子 $F \subseteq E$。

▶ **定理 6** ([2, 6])。 ∀偶数有向循环因子是 *NP* 完全问题。

设 $G = (V,\ E)$ 是偶数有向循环因子的一个实例，即一个有向图。我们按如下方式构造一个等价的不旋转的 Swish 实例（$\mathcal{C}$ k）。

设 $h := |V|$ 和 $w := 4$。我们用 V 为行编索引，用 {1，2，3，4} 为列编索引，并将网格上的每个点 $(v,\ j)$ 简记为 v^j。

对于每条边 $e = (u,\ v) \in E$，我们创建一张卡片 $c_e = (u^1,\ v^1)$。此外，对于每个顶点$v \in V$，我们创建三张卡片 $f_{v,\ j} = (v^j,\ v^{j+1})$ $(j = 1,\ 2,\ 3)$。令 $\mathcal{C}$ 为所创建卡片的集合（见图 4）。

下面的断言完成了该归约，其中我们令 $k = 4|V|$ （见图 5）。

▷ 断言 7。 G 存在偶数有向循环因子，当且仅当$\mathcal{C}$ 允许翻转卡片但禁止旋转卡片时，存在大小为 $4|V|$ 的 Swish。

证明。 令 $F \subseteq E$ 为 G 的偶数有向循环因子。由于 F 的每个连通分量都是偶数长度的有向环，我们可以用两种颜色（例如红色 和蓝色）为 F 的边着色，使任意一对相邻边颜色不同。固定这样一种着色，并令$\mathcal{S} := \{\ c_e \mid e \in F\} \cup \{\ f_{v,j} \mid v \in V,\ j \in \{1, 2, 3\}\ \}$。注意，$|\mathcal{S}| = |F| + 3|V| = 4|V|$。我们证明$\mathcal{S}$ 是一个 Swish。

首先，我们确定如何针对每个 $e = (u, v) \in F$ 使用卡片 $c_e = (u^1, v^1) \in \mathcal{S}$。如果 e 是红色，就原样使用，即作为 $c_e = (u^1, v^1)$。相反，如果 e 是蓝色，就将其翻转后使用，即作为$c_e^{\mathrm{F}} = (u^4,\ v^4)$。由于 F 是一个（偶数）有向圈因子，每个顶点恰有一条入边，其

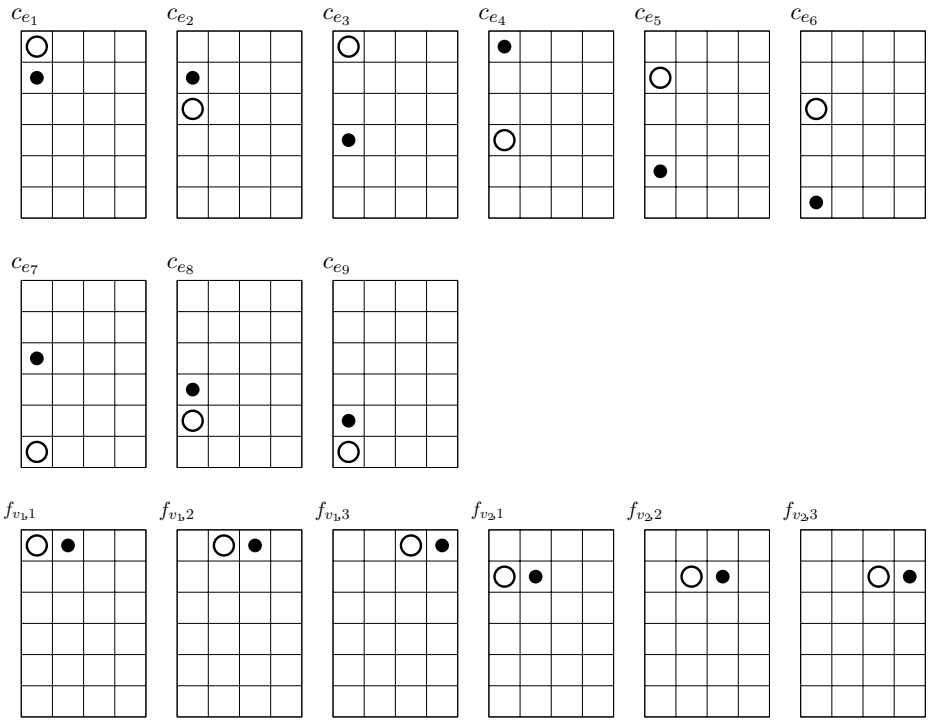


图 4 由图 3 中偶数有向循环因子实例生成的卡片集合，其中省略了 $f_{v_i,\ j}$ 对于 $i = 3$，4、5、6（它们与 $f_{v_1,\ j}$ 和 $f_{v_2,\ j}$ 类似）。

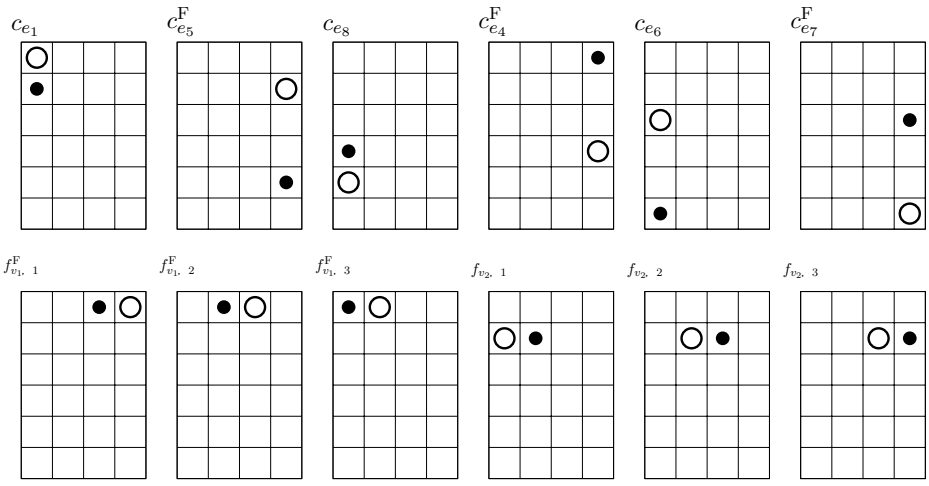


图 5 与图 3 中解相对应的不旋转的 Swish，其中省略了$f_{v_i,\ j}$ 对于 $i = 3$，4、5、6。

F 以及 F的一条出边，且两者着色不同。因此，根据上述定义，对于每个顶点 $v \in V$，两个点 v^1 和 v^4 中，一个恰好接收一个圆环且没有球，另一个恰好接收一个球且没有圆环。如果 v^1 接收一个球（而 v^4 接收一个圆环），那么我们就按原样使用 $f_{v,\ j} = (v^j,\ v^{j+1}) \in \mathcal{S}(j = 1,\ 2,\ 3)$；否则，我们将它们翻转后使用，即用作 $f^{\mathrm{F}}_{v,\ j} = (v^{5-j},\ v^{4-j})$。在这两种情况下，两个中间点 v^2 和 v^3 都会从这三张卡片中恰好接收一个圆环和一个球，而v^1 和 v^4 则各自恰好接收一个圆环或一个球，且与从卡片 $c_e \in \mathcal{S}$接收到的内容一致。因此，$\mathcal{S}$ 确实是不旋转的 Swish。

现在我们证明反方向。设 $\mathcal{S}$ 是一个大小为 $4|V|$ 且不进行旋转的 Swish，这意味着我们只使用翻转，就能在网格的每个点上恰好放置一个圆环和一个球。首先，每个顶点 $v \in V$ 的中间点 v^2 和 v^3 只能由三张卡片 $f_{v,j}$ $(j = 1,2,3)$覆盖，因此必须使用它们。具体而言，由于中间点上的限制，这三张卡片只能全部按原样使用，或全部翻面后使用。接着，对于边 $e = (u, v)$，我们总共恰好使用 $|V|$ 张形如 $c_e = (u^1, v^1)$ 的卡片，使得对于每个顶点 $v \in V$，v^1 和 v^4 中一个恰好接收到一个圆环且没有球，另一个则恰好接收到一个球且没有圆环。

令 $F \coloneqq \{\, e \in E \mid c_e \in \mathcal{S} \,\}$。根据上述论证，对于每个顶点 $v \in V$，恰好有一条入边（对应一个球）和一条出边（对应一个圆环）包含在 F中；这意味着 F 的每个连通分量都是一个有向环，并且覆盖所有顶点。此外，对应的两张卡片中，一张必须按原样使用，另一张必须翻面。根据这一点，我们可以用两种颜色（翻面或未翻面）为 F 中的每个有向环着色，使得任意一对相邻边颜色不同，这意味着环的长度必须为偶数。因此，F 是一个偶数有向循环因子，证明完毕。◀

▶ 注 8。 [2,6] 中并未明确写出，但定理 6 的证明表明，∀即使将输入图限制为同时满足以下条件，偶数有向循环因子问题仍是 NP 难的：

- 每个顶点的入度和出度都至多为 3（根据 [6] 可立即得到；对于 2-连通问题，也可在归约前对输入图进行预处理，再结合 [2]得到），且该图是平面的（由 [6] 与 [9]结合可得）。
- 此外，很容易看出，我们可以进一步加强对输入图的第一项限制$G = (V,\ E)$，如下所示：G 是三部的，其中 G 的顶点集可划分为 $(V^-,\ V^\circ \cup V^+)$，V中的每条边 E 要么从 V^- 指向 V°，要么从 V° 指向 V^+，要么从 V^+ 指向 V^-；V^- 中的每个顶点出度为 1、入度至多为 3，V° 中的每个顶点的入度和出度均为 1，而 V^+ 中的每个顶点入度为 1、出度至多为 3。
- 这可以通过有向图的一种标准变换来观察：将每个顶点 v拆分为三个副本 v^-（用于入边）、v°（用于保持长度奇偶性）和 v^+（用于出边），并添加两条额外的边（$v^-,\ v^\circ$）和（$v^\circ,\ v^+$），再将每条原始边（$u,\ v$）替换为（$u^+,\ v^-$）。该变换不会改变问题。

基于对 ∀偶数有向循环因子的输入图 $G = (V,\ E)$施加的这一限制，我们可以对不旋转的 Swish 的卡片输入集合 $\mathcal{C}$ 施加以下限制：对于网格上的每个点 p，要么（∗）

- 在 p 恰有一张卡片带圆环且至多两张卡片带球，或者在 p 恰有一张卡片带球且至多两张卡片带圆环。通过从证明中构造的卡片集合 $\mathcal{C}$ 构造一个等价的卡片集合 $\mathcal{C}'$，即可看出这一点。

由三划分和度条件可知，从 V^- 到 V° 的边以及从 V° 到 V^+ 的边分别构成匹配。对于这样的边 $e=(v^-,v^\circ)$ 和 $e'=(v^\circ,v^+)$ ($v^-\in V^-$，$v^\circ\in V^\circ$，$v^+\in V^+$)，我们将卡片 $c_e=((v^-)^1,(v^\circ)^1)$ 和 $c_{e'}=((v^\circ)^1,(v^+)^1)$ 按照 C 中的方式加入 C′。此外，我们可以假设在 V^+ 和 V^- 之间存在一个由从 V^+ 到 V^- 的边构成的完美匹配；否则，G 显然是 $\forall$偶数有向循环因子（G 即使不施加奇偶性约束也不包含任何圈因子）的否实例，而这可以在多项式时间内判定。固定任意一个这样的完美匹配 M，对于每条边 $e=(u^+,v^-)\in M$，我们按照 C 中的方式将卡片 $c_e=((u^+)^1,(v^-)^1)$ 加入 C′。对于从 V^+ 到 V^- 的其余边 $e=(u^+,v^-)\in E\setminus M$，我们不加入卡片 $c_e\in$ C，而是将翻转卡片$c_e^{\mathrm{F}}=((u^+)^4,(v^-)^4)$ 加入 C′。按构造，此时 C′ 满足以下条件：

- 对于每个顶点 $v\in V=V^-\cup V^\circ\cup V^+$，恰有一张卡片在 v^1处有一个圆环，且恰有一张卡片在 v^1处有一个球；对于每个顶点 $v^-\in V^-$，至多两张卡片在 $(v^-)^4$ 处有球，且没有
- 卡片在此处有圆环；对于每个顶点 $v^\circ\in V^\circ$，没有任何卡片在 $(v^\circ)^4$ 处有圆环或球；
- 对于每个顶点 $v^+\in V^+$，至多两张卡片在 $(v^+)^4$ 处有圆环，且没有卡片在此处有球。

换言之，若再分别满足以下条件，则所需条件 ($*$) 将成立：每个 v^1 至多接收一个圆环或一个球，每个 $(v^-)^4$ 恰好接收一个圆环，每个 $(v^\circ)^4$ 恰好接收一个圆环或一个球，并且每个 $(v^+)^4$ 恰好接收一个球。实际上，可以如下实现：对于每个 $v\in V^-\cup V^\circ$，我们在 C′ 中加入翻转卡片$f_{v,j}^{\mathrm{F}}=(v^{5-j},v^{4-j})$ ($j=1,2,3$)，而非 $f_{v,j}\in$C；对于每个 $v^+\in V^+$，我们则像在 C 中一样，在 C′ 中加入卡片 $f_{v^+,j}=((v^+)^j,(v^+)^{j+1})$ ($j=1,2,3$)。注意，在任一情况下，每个中间点 v^2,v^3 都恰好接收一个圆环和一个球。因此，我们可以构造一个与 C 等价且满足条件 ($*$) 的卡片集合 C′。

3.3 Swish 的 NP 困难性

本节将证明，和上一节所示的不旋转的 Swish 一样，Swish 和不翻转的 Swish 也是 NP 难的。证明几乎完全相同，唯一的区别在于网格上各点的坐标安排。

在前面的证明中，我们使用了 $|V|\times 4$ 网格，其中每个点 $(v,\ j)$ ($v\in V,\ 1\le j\le 4$) 都简记为 v^j。这里我们考虑将这些点排列在同一条垂直线上。具体来说，令 $h:=4|V|$ 和 $w:=1$，并使用 $v^j(v\in V,\ 1\le j\le 4)$ 为 $h\times w$ 网格上的 $4|V|$ 个点命名，使得点 $(i,$ 1) 被命名为 v^j，当且仅当点 $(h-i+1,$ 1) 被命名为 v^{5-j}。这样一来，水平翻转便失去意义，而旋转则发挥与上一节中的水平翻转相同的作用。因此，同样的归约适用于这种情形（允许旋转，且水平翻转可以允许，也可以不允许），由此完成定理 2 的证明。

4 总结

本文通过解决此前悬而未决的每张卡片两个符号这一情形，完成了 Swish 的复杂性分类。我们的结果表明，Swish 的计算复杂性取决于每张卡片上的符号数量与卡片所允许的几何变换之间的相互作用。具体而言，我们确定了一条明确的分界线：既不允许翻转也不允许旋转时，每张卡片两个符号的 Swish 可以在多项式时间内求解；而即便只允许一种翻转方式或 180 度旋转，问题就已是 NP 完全。结合先前的结果，这些发现完整呈现了 Swish 在所有自然游戏变体中的复杂性。

10 Swish 的计算复杂性

除判定复杂性外，另一个引人关注的研究方向是探究无法组成 swish 的卡牌集合的结构。Dailly、Lafourcade 和 Marcadet [3] 率先系统研究了这类无法组成 *swish* 的配置。在他们的广义情形中，他们构造了大量不含 swish 的卡牌集合，其密度渐近地达到较高水平。此外，对于由 60 张卡片组成的原始商业版牌组，他们找到了一个包含 28 张卡片的无 swish 子集。

一个自然且尚未解决的问题是：这个 28 是否为最优值？能否在原始 Swish 牌组中找到更大的无 Swish 子集，还是说任意 29 张卡片都必然可以组成一个 Swish？更广泛地说，确定固定牌组或受限对称性模型下无 Swish 集合大小的紧确界，似乎是一个很有前景的未来研究方向。这将加深我们对游戏背后组合结构的理解，也可能需要采用不同于广义情形下所用的技术。

参考文献

1 Josh Alman、Ran Duan、Virginia Vassilevska Williams、Yinzhan Xu、Zixuan Xu 和 Renfei Zhou。更多的不对称性带来更快的矩阵乘法。发表于 2025 年 ACM-SIAM 离散算法年度研讨会（SODA）论文集，第 2005–2039 页，2025 年。2 Jørgen Bang-Jensen 和 Stéphane Bessy。（弧）不相交流网络。理论计算机科学，526：28–40，2014 年。3 Antoine Dailly、Pascal Lafourcade 和 Gael Marcadet。他们是如何设计这款游戏的？Swish：计算复杂性与不可玩局面。发表于第 12 届国际趣味算法会议（FUN 2024），第 291 卷，第 10:1–10:19 页，2024 年。4 Jack Edmonds。最大匹配与一个具有 0、1 顶点的多面体。美国国家标准局研究杂志 B，69(125-130)：55–56，1965 年。5 Jack Edmonds 和 Richard M. Karp。网络流问题算法效率的理论改进。ACM 学报，19(2)：248–264，1972 年。6 Florian Hörsch、Csaba Király、Mirabel Mendoza-Cadena、Gyula Pap、Eszter Szabó 和 Yutaro Yamaguchi。关于圈因子的奇数和偶数更难问题。arXiv 预印本 arXiv:2510.18393，2025 年。7 Chien-Chung Huang 和 Telikepalli Kavitha。边权较小的一般图最大权匹配的高效算法。发表于第 23 届 ACM-SIAM 离散算法年度研讨会（SODA）论文集，第 1400–1412 页。SIAM，2012 年。8 Harold W. Kuhn。指派问题的匈牙利算法。《海军研究后勤季刊》，2(1-2)：83–97，1955 年。9 Dániel Marx。平面图边上的列表着色与预着色扩展问题的 NP 完全性。《图论杂志》，49(4)：313–324，2005 年。10 Ketan Mulmuley、Umesh V Vazirani 和 Vijay V Vazirani。匹配问题与矩阵求逆一样容易。《组合数学》，7(1)：105–113，1987 年。11 Christos H Papadimitriou 和 Mihalis Yannakakis。受限生成树问题的复杂性。ACM 学报，29(2)：285–309，1982 年。12 Dana Rowland。Swish 数学探索。发表于 MAA MathFest（美国数学协会），2017 年。13 Piotr Sankowski。矩阵乘法时间内求解最大权二分图匹配。理论计算机科学，410(44)：4480–4488，2009 年。14 Ryotaro Sato 和 Yutaro Yamaguchi。矩阵乘法时间内求解精确匹配。arXiv 预印本 arXiv:2508.04081，2025 年。15 Gali Shimoni 和 Zvi Shalem。Swish。ThinkFun 桌面游戏。 `https://www.thinkfun.com/products/swish/`（截至 2026年1月12日已不再提供），规则手册： `https://legacy.thinkfun.com/wp-content/uploads/2015/09/Swish-1512-IN03.pdf`，视频： `https://www.youtube.com/watch?v=Hzi28pbeWWw` （截至 2026年1月12日可观看），2011 年。