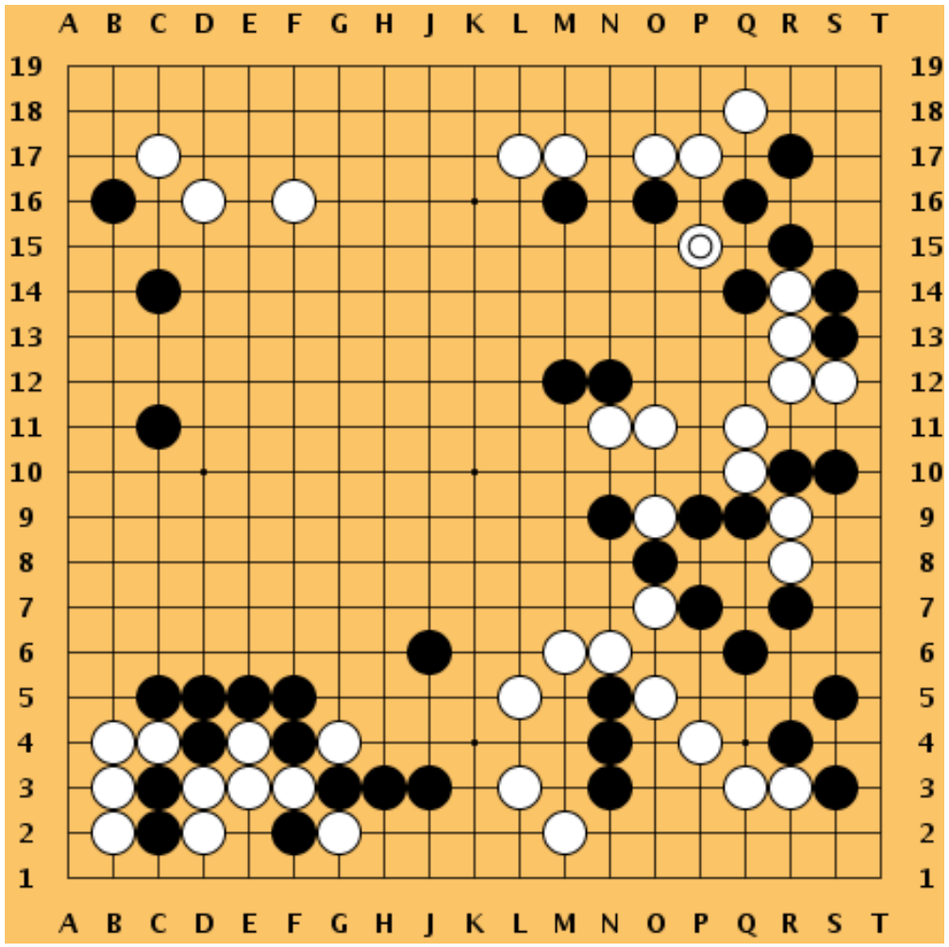
AlphaGo & AlphaGo Zero

Эмиль Каюмов
emil.kayumov@gmail.com
t.me/emilkayumov

Го

- Го — древняя китайская игра
- Поле 19 на 19
- Соперники поочерёдно ставят камни
- Можно окружать (захватывать) чужие камни
- В конце игры подсчитывается захваченная территория
- 10^{170} возможных комбинаций



Го: всё так плохо?

- Полностью **детерминированная** среда
- Полностью **наблюдаемая** среда
- **Дискретное** пространство действий
- Есть **симулятор**
- **Короткие** игры
- **Легко оценить** результат
- Есть **база** игр людей

- То есть всё не так плохо (пост Karpathy «AlphaGo, in context»)

AlphaGo

- Известные техники соединили вместе:
  - Supervised learning
  - REINFORCE
  - Value function
  - Monte Carlo Tree Search
- Статья от 28 января 2016

AlphaGo: шаг 1

- Сеть для предсказания ходов людей (SL-policy)
- Данные — истории игр людей
- Оперирует набором признаков
- 57% точности
- Дополнительно на расширенном наборе признаков обучается быстрая линейная модель (fast rollout policy)

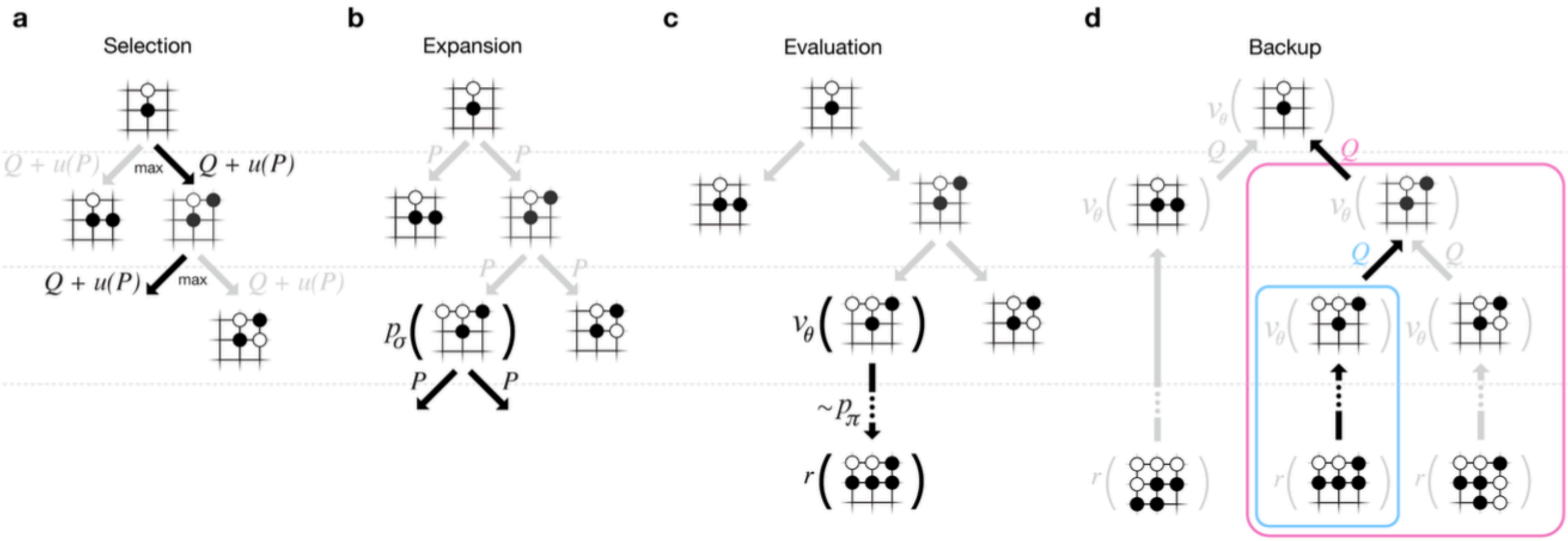
AlphaGo: шаг 2

- Дообучаем SL-policy, играя с собой более старым
- Играем партию, узнаём победителя, обновляем веса
- Играем жадно на каждом ходе

AlphaGo: шаг 3

- Обучаем value network для оценки победителя из произвольного хода
- Учится только на N+2 ходе синтетических данных (SL + random + RL)

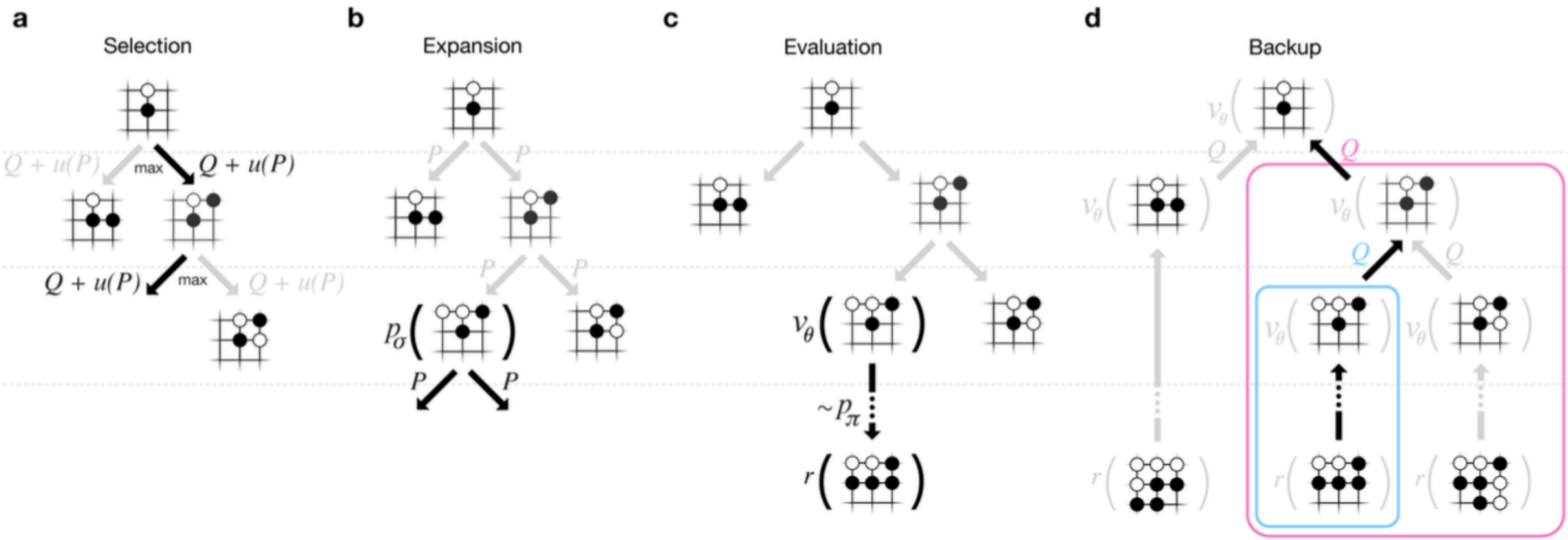
AlphaGo: шаг 4



- Имеем SL-policy, fast rollout policy, RL-policy, value function

- Имеем дерево позиций, находимся в корне, для каждой вершины есть Q — показатель уверенности в победе

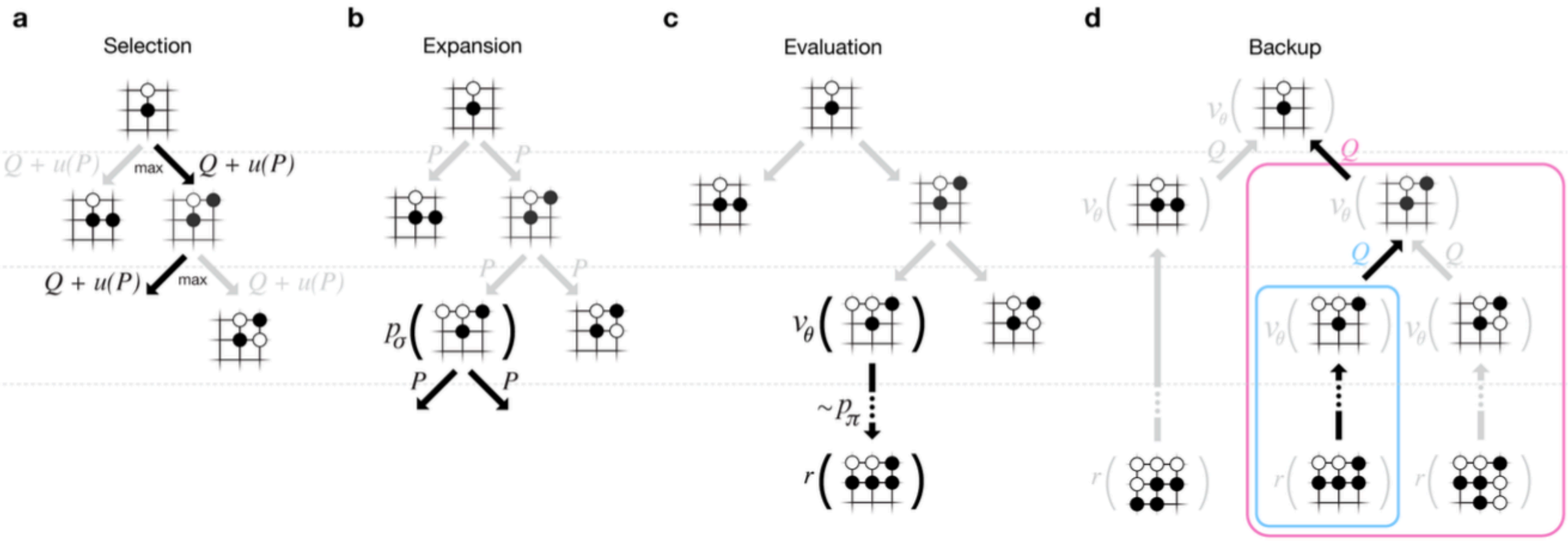
AlphaGo: шаг 4



a. Симуляция идёт по дереву по max(Q+u(P))

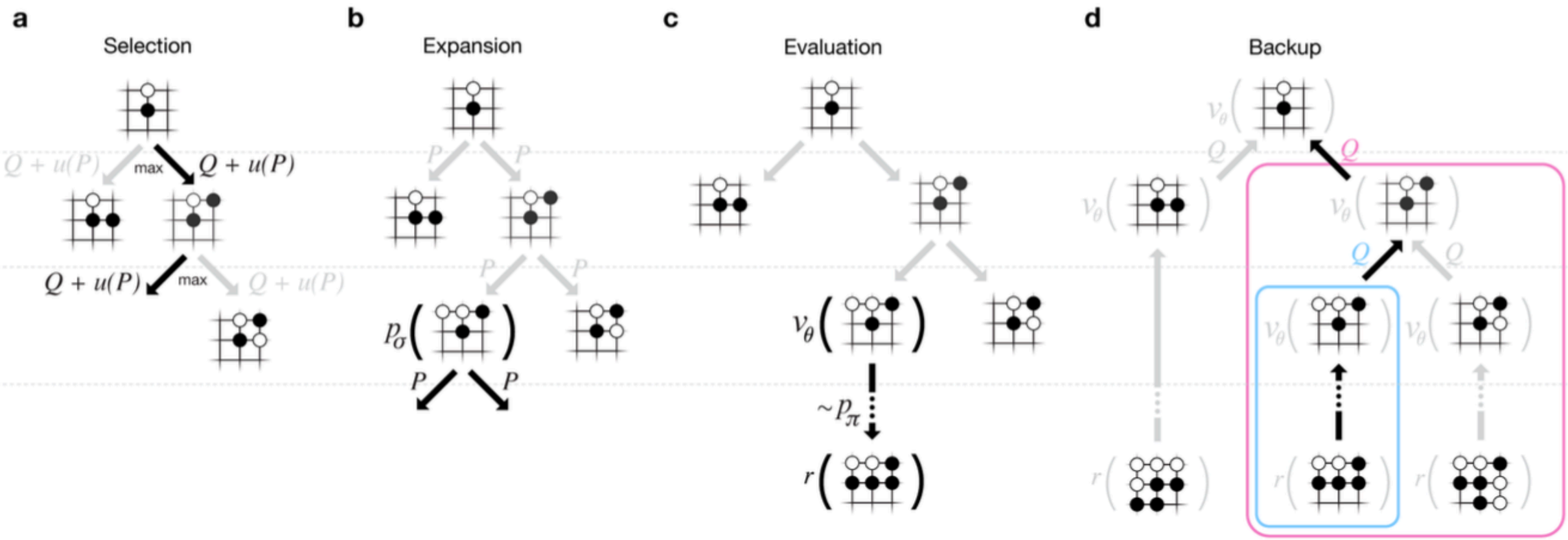
- u(P) — добавка, стимулирующая exploration
- P — априорные вероятности ходов от policy

AlphaGo: шаг 4



b. Когда доходим до листа, добавляем новую вершину с возможными ходами и их вероятностями P через policy

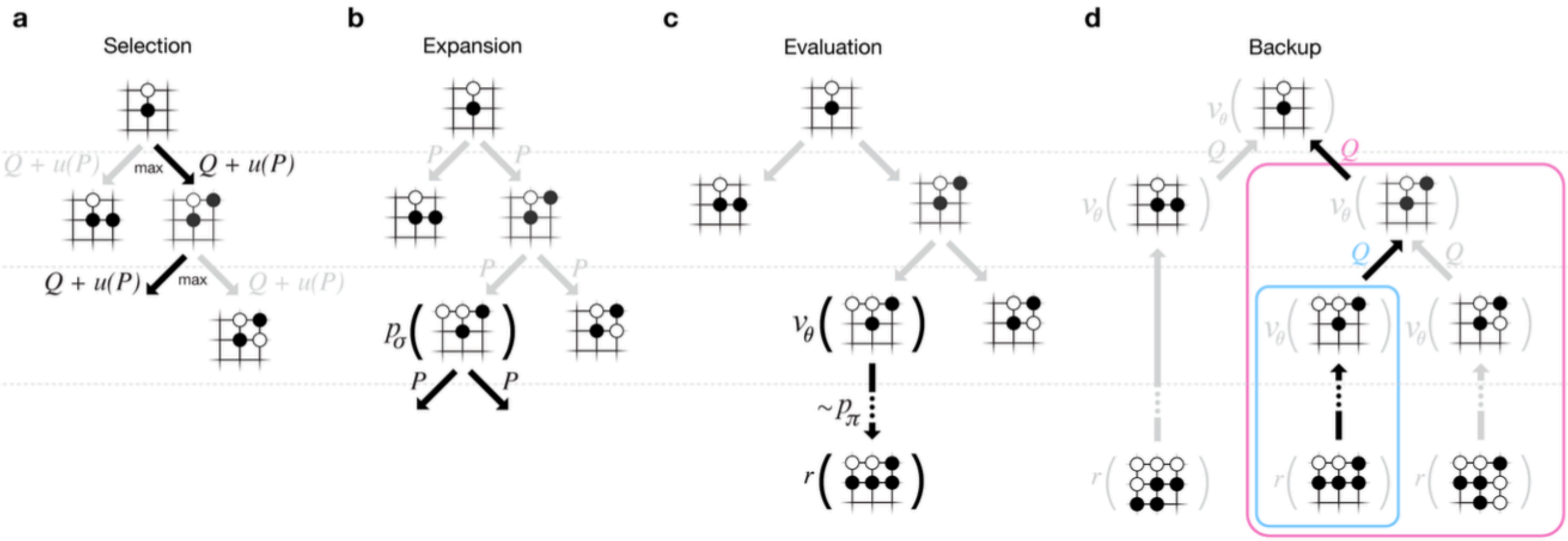
AlphaGo: шаг 4



c. Оцениваем новую вершину:

$$Q_{new} = \frac{value\ function + fast\ rollout\ policy}{2}$$

AlphaGo: шаг 4



d. Обновляем все Q как среднее по всем потомкам

AlphaGo: шаг 5

- Во время игры используем не Q, а «количество хождений в вершину» (стабильнее)
- Для подсчёта априорных P используется не RL-policy, а SL-policy
  - Эмпирически лучше
  - Возможно, играет разнообразнее

AlphaGo: итого

- Октябрь 2015, чемпион Европы Fan Hui — 5:0
- Март 2016, Lee Sedol — 4:1
- Конец 2016 – начало 2017, онлайн с игроками с топовых позиций — 60:0
- Май 2017, топ1 мирового рейтинга Ke Jie — 3:0

AlphaGo: претензии

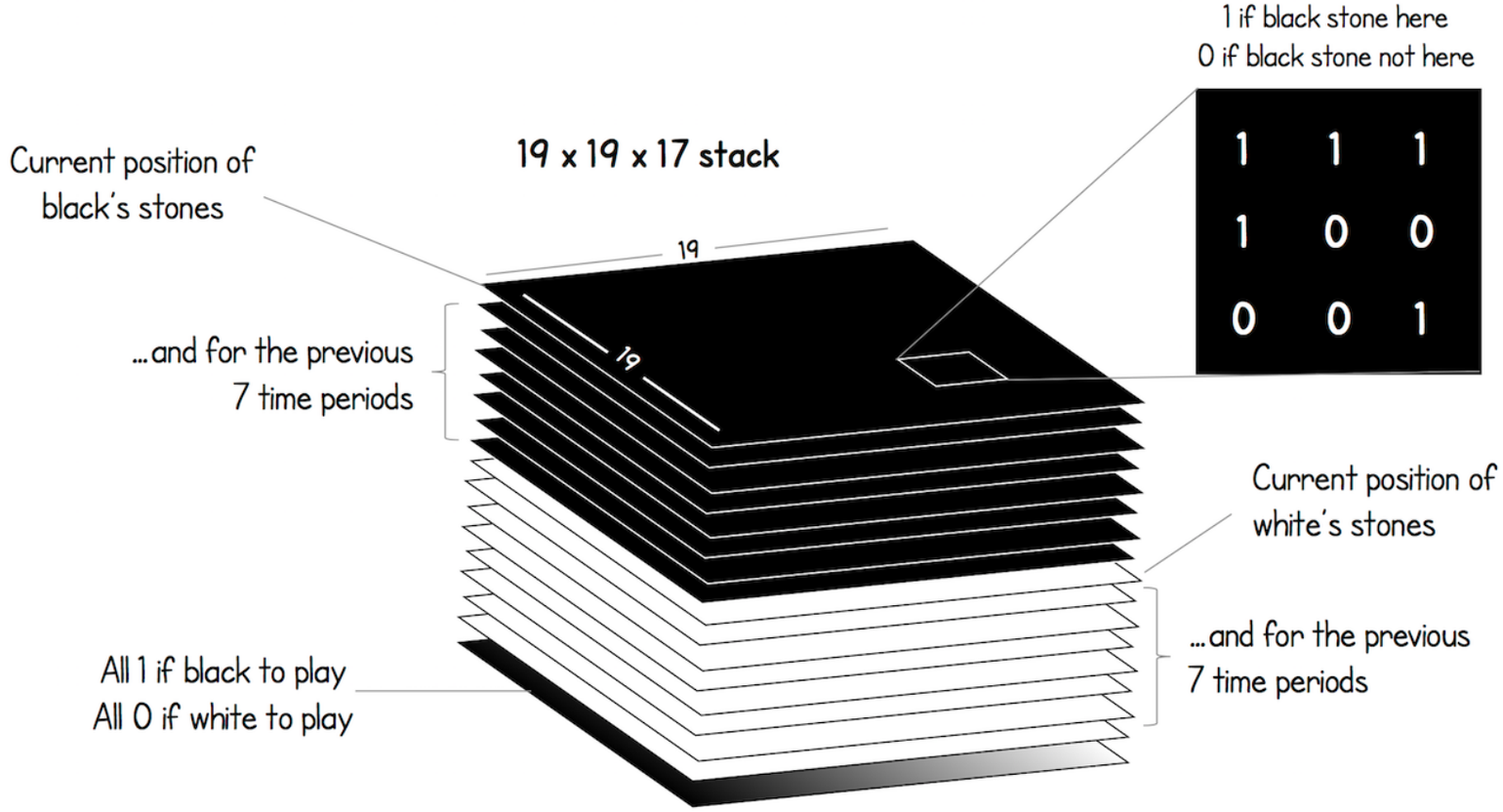
- Используются игры людей для обучения
- Подаются признаки, а не просто текущая позиция
  - «сколько своих камней ты пожертвуешь, если сходишь в данную точку»
  - «поучаствует ли камень, поставленный в эту точку, в лестничном построении»
  - «сколько ходов назад был поставлен камень»
- Нужно много мощностей (176 GPU у AlphaGo Fan)
  - На самом деле дальше взяли TPU

AlphaGo Zero

- Май 2017: «AlphaGo уходит из Го, больше никаких матчей проводить не будем».
- Октябрь 2017: AlphaGo Zero

AlphaGo Zero: общее

- На вход только состояние поля за последние 8 шагов + свой цвет
- История нужна для того, чтобы сеть сама выучила правило Го против повторных ходов



AlphaGo Zero: общее

- Одна сеть с двумя головами: policy и value
- MCTS не только для финальных игр, но и для обучения
- Никаких историй игр от «кожаных мешков»

AlphaGo Zero: MCTS

- Перед **каждым** ходом:
  1. Идём по Q+U (U — добавка для поиска новых путей)
  2. Доходим до конца, создаём новую вершину, вычисляем сетью **v** и **P** (один раз)
  3. Всем потомкам устанавливаем $N = V = Q = 0$
  4. Обновляем все вершины выше текущей

$$N = N + 1;\ V = V + v;\ Q = \frac{V}{N}$$

  5. Повторяем цикл 1600 раз

AlphaGo Zero: ход

- Для реальной игры идём туда, где максимальный N
- Во время обучения выбираем из $\pi_i \sim N_i^{\frac{1}{T}}$

AlphaGo Zero: сеть

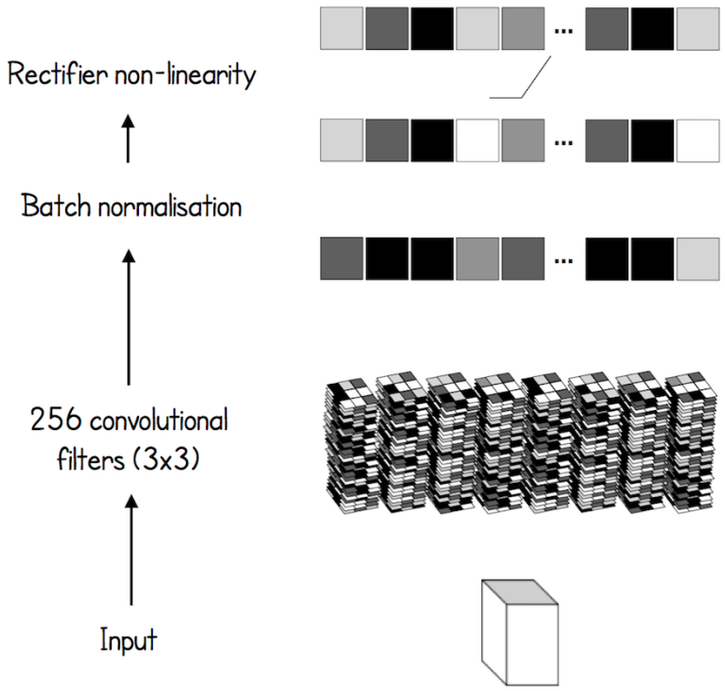
$$L = (z - v)^2 + \pi^\top \log p + c\|\theta\|^2$$

1. Сеть A играет сама с собой 25000 раз
2. Берём 1000 батчей по 2048 позиций из 500000 последних игр
3. Получаем из сети A сеть B
4. Сеть B играет против A 400 раз и если побеждает ≥55% раз, то становится новой сетью A, иначе начинаем заново

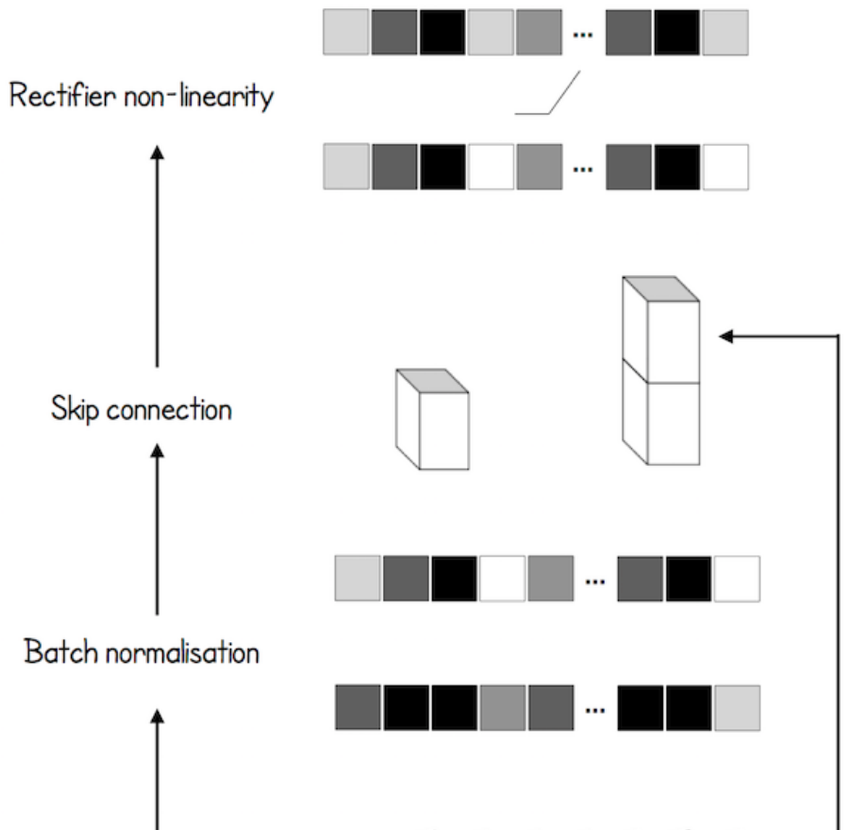
AlphaGo Zero: сеть

1. 1 convolutional-блок
2. 40 residual-блоков (подряд)
3. Policy-голова и value-голова

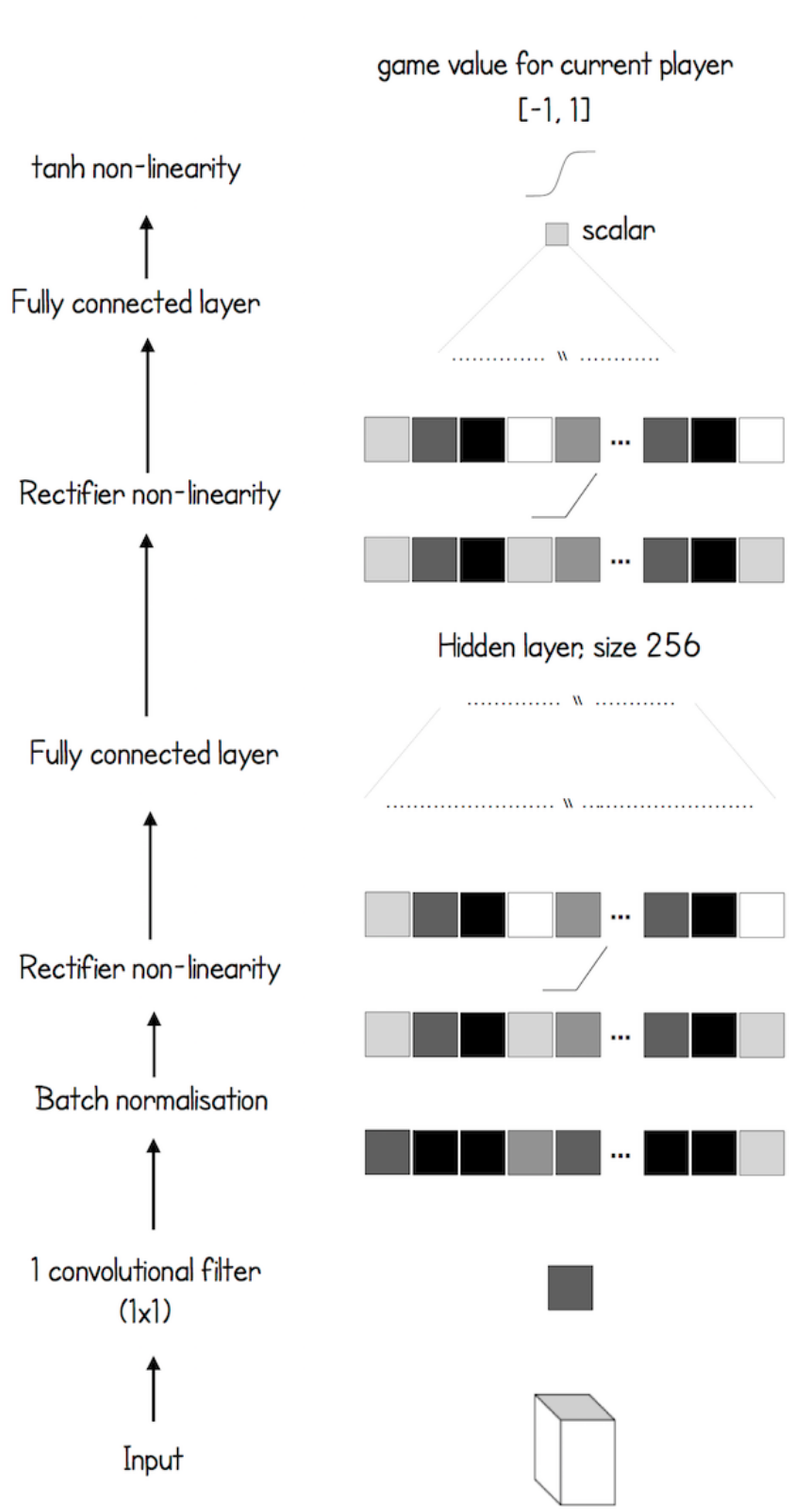
A convolutional layer



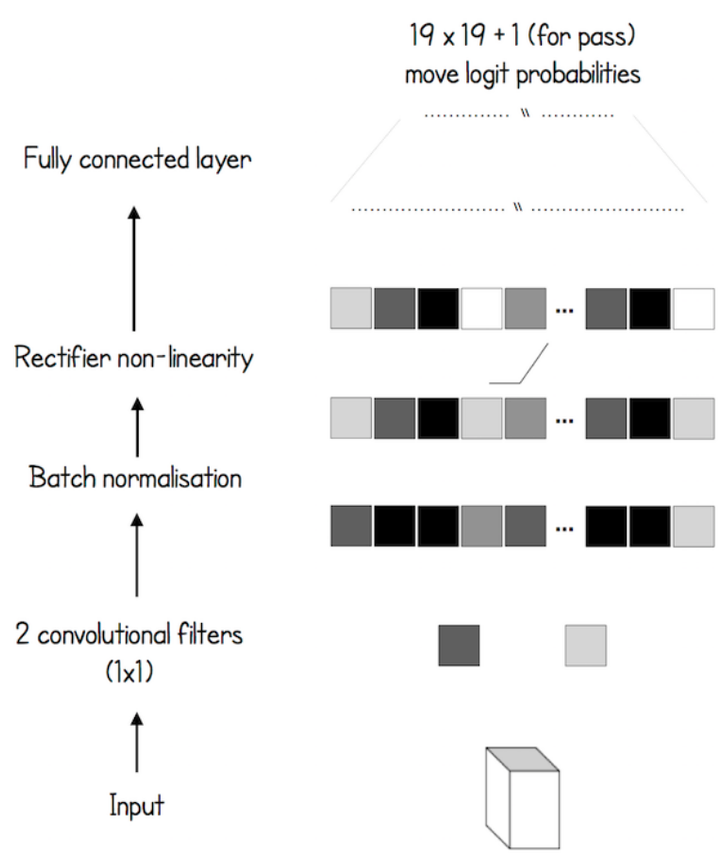
A residual layer



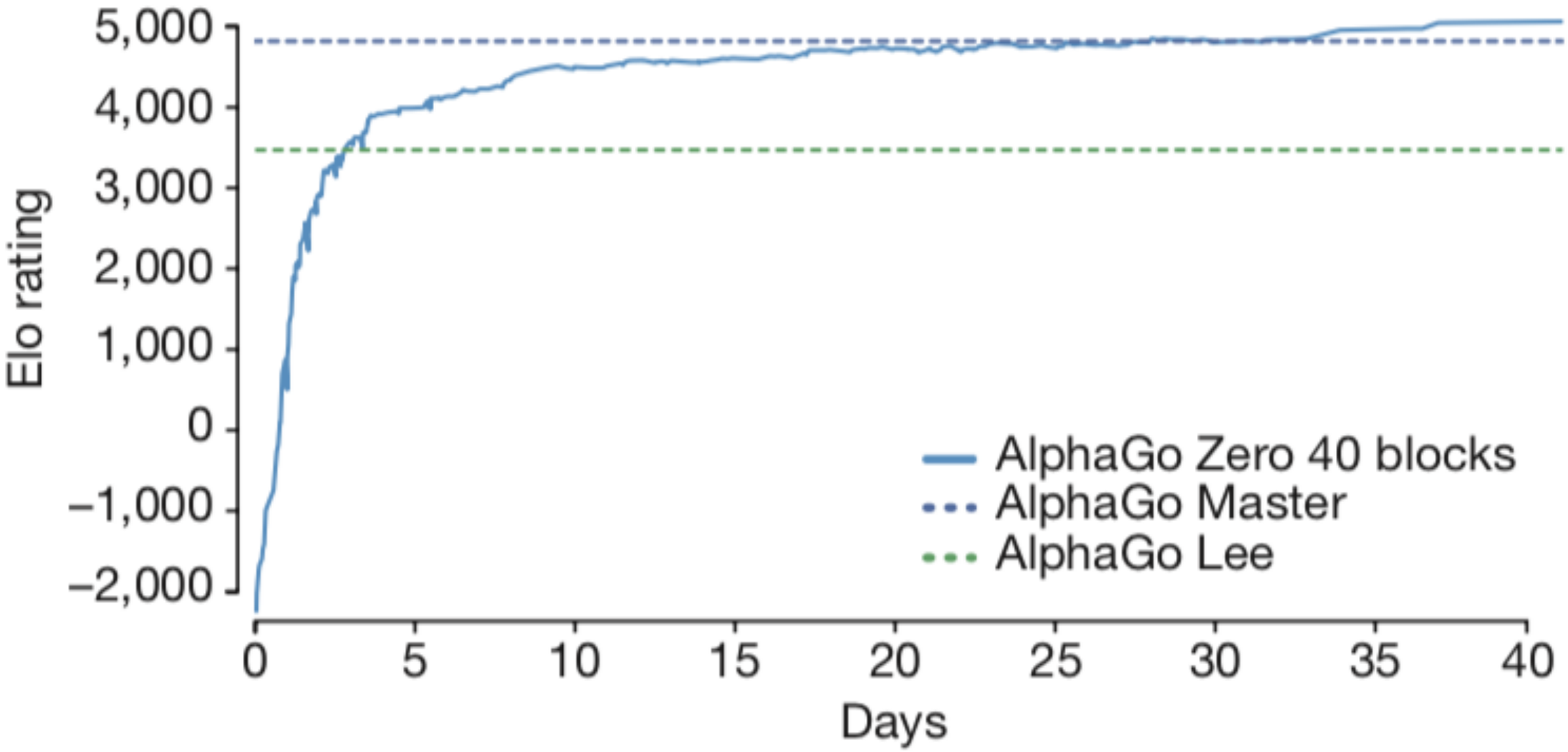
The value head



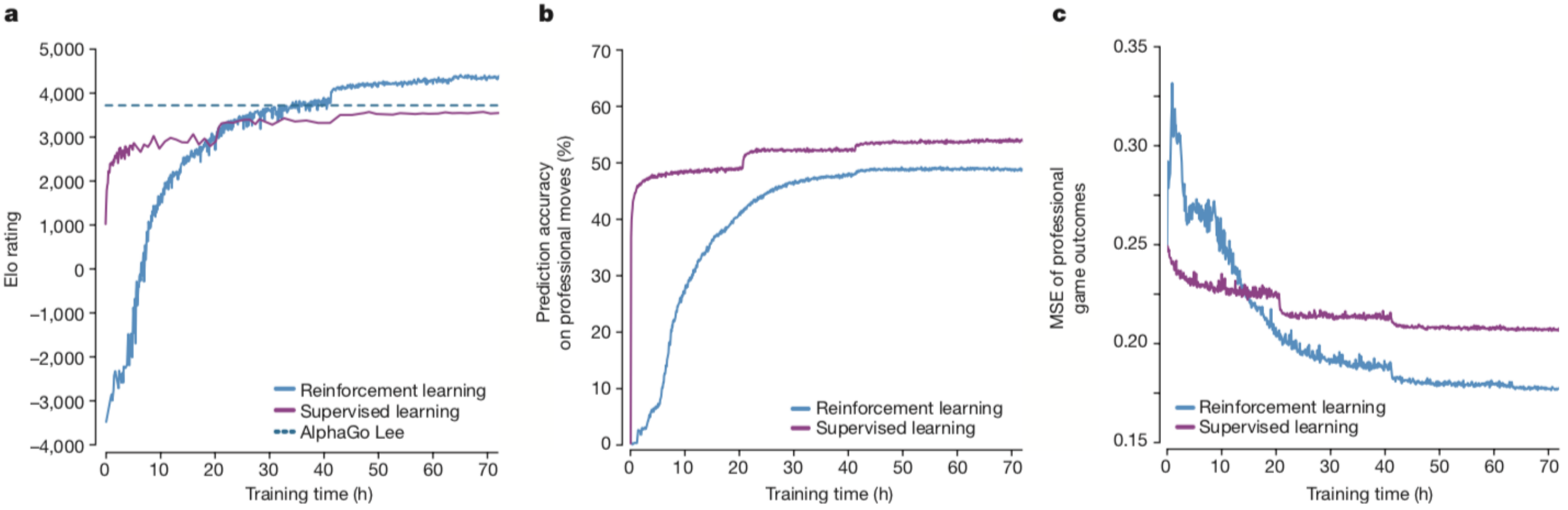
The policy head



AlphaGo Zero: итого



AlphaGo Zero: итого



AlphaGo Zero: итого

- «Humankind has accumulated Go knowledge from millions of games played over thousands of years, collectively distilled into patterns, proverbs and books. In the space of a few days, starting *tabula rasa*, AlphaGo Zero was able to rediscover much of this Go knowledge, as well as novel strategies that provide new insights into the oldest of games.»

AlphaGo Zero: итого

- Что теперь? Ничего, просто люди уступили ещё и в Го.
- Например, для роботов:
    1. Пространство действий континуально
    2. Нет полной информации
    3. Симуляторы не идеальны
    4. Эпизоды длиннее
    5. Сложнее определить успех
    6. Нет исторических примеров

Alpha Zero

- Обобщение AlphaGo Zero:
  - Больше свободы в выборе гиперпараметров (без подбора, одинаковая инициализация для всех игр)
  - Постоянное обновление сети (без сражения двух версий сети)
    - Не использует особенности Го (симметрия, отсутствии ничьи)
- Переиграли программы в шахматы, сёги и сыграли на уровне 3-дневного AlphaGo Zero
  - Не обошлось без критики за некоторую нечестность

Ссылки

- AlphaGo Deepmind blog
- AlphaGo: Mastering the game of Go with deep neural networks and tree search
- AlphaGo Zero: Mastering the game of Go without human knowledge
- AlphaZero: Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm
- Фильм
- AlphaGo на пальцах (на русском языке)
- AlphaGo Zero на пальцах (на русском языке)