I. Исследование влияния параметра гамма на эффективность алгоритма Policy Iteration

1. Описание

В данной работе исследуется влияние гиперпараметра γ на качество обучения алгоритма Policy Iteration для среды Frozen Lake. Этот параметр определяет, как сильно алгоритм учитывает долгосрочные вознаграждения по сравнению с краткосрочными, что влияет на стабильность и эффективность обученной стратегии.

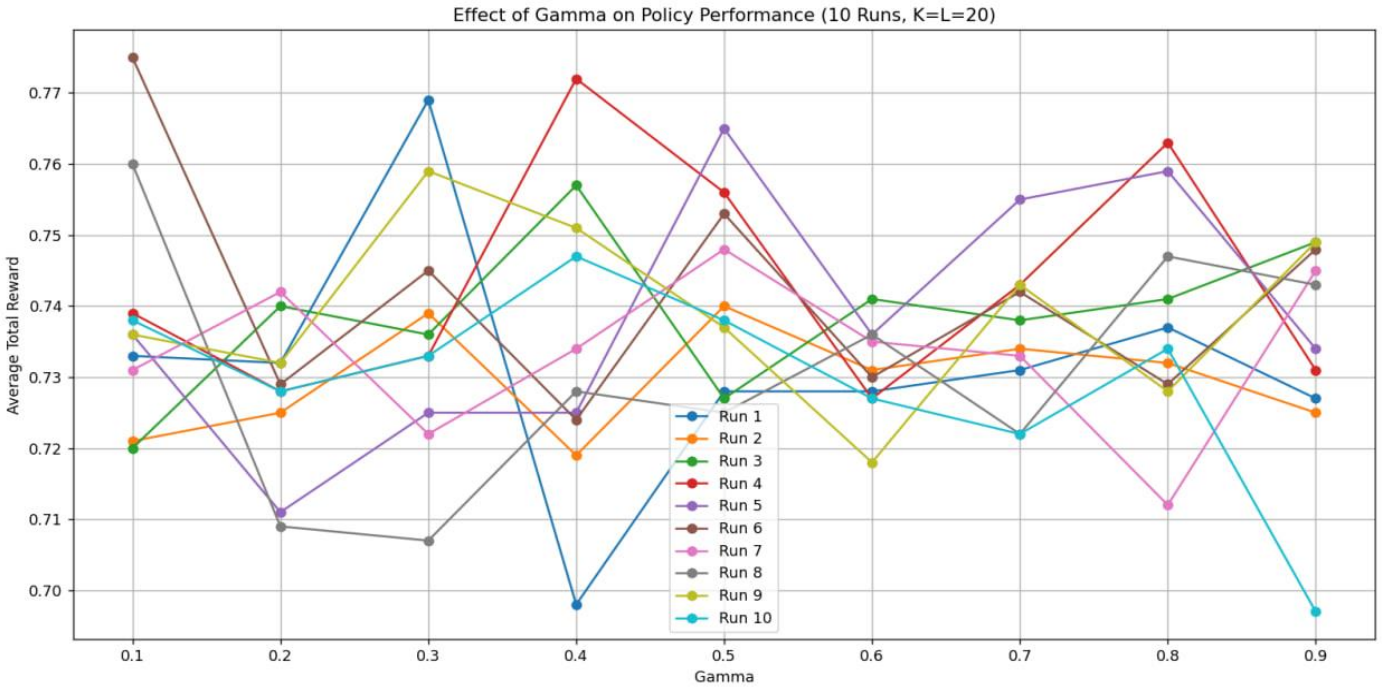
Gamma (γ) — параметр дисконтирования, который управляет весом будущих вознаграждений относительно текущих. При γ=1.0 алгоритм максимально учитывает будущие награды, а при низких значениях (γ близко к 0) алгоритм ориентируется только на краткосрочные вознаграждения.

K — количество итераций улучшения политики. Каждая итерация улучшения приближает политику к оптимальной, но большое количество таких итераций не всегда приводит к значительному улучшению качества стратегии, если текущая политика уже достаточно близка к оптимальной.

L — количество итераций оценки политики. Этот параметр определяет глубину оценки текущей политики перед её улучшением. Слишком низкие значения могут привести к недостаточной точности оценки, однако после определенного количества итераций дополнительное увеличение L не оказывает значительного влияния на итоговую стратегию.

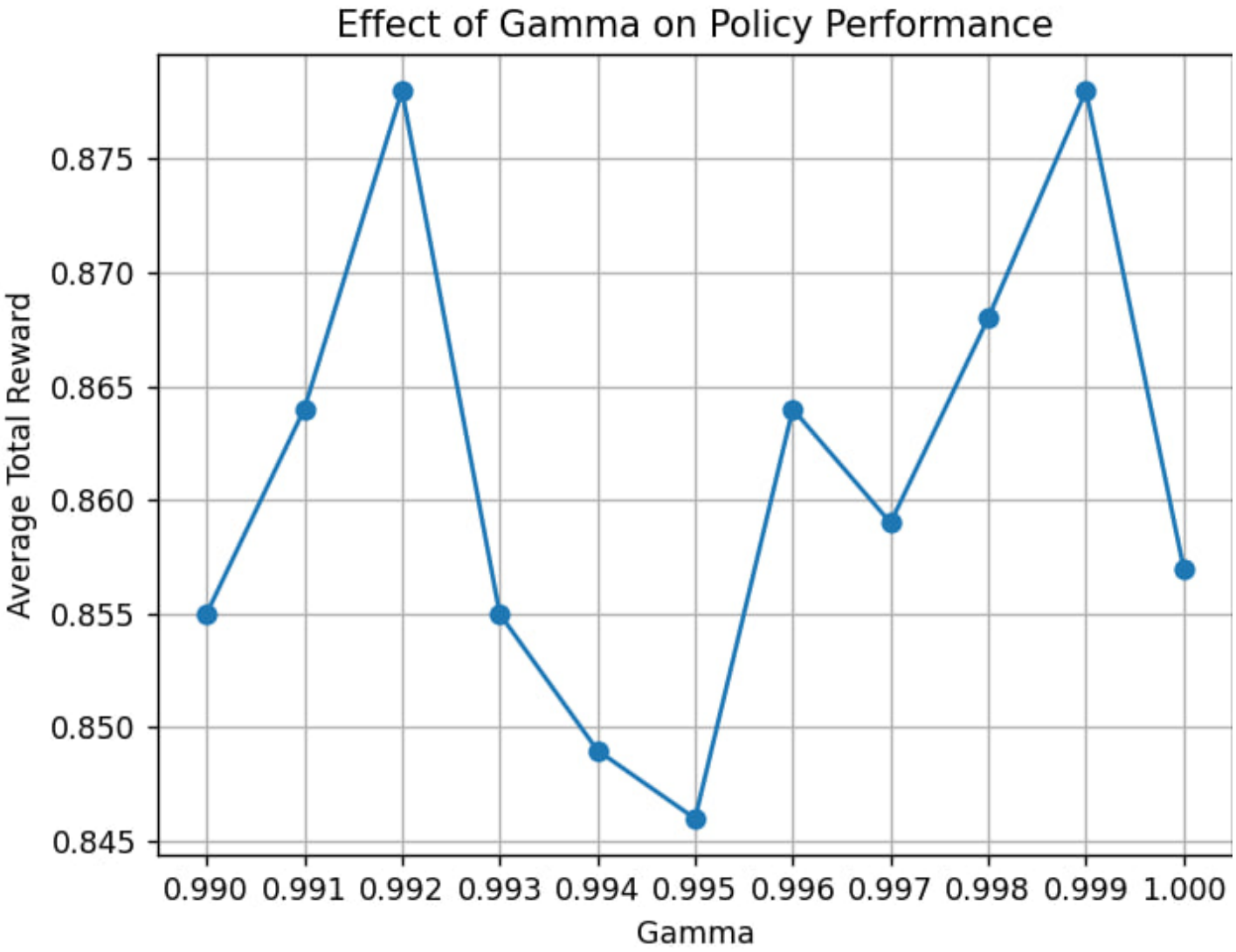
2. Результаты

Тестирование для гамма от 0.1 до 0.9:



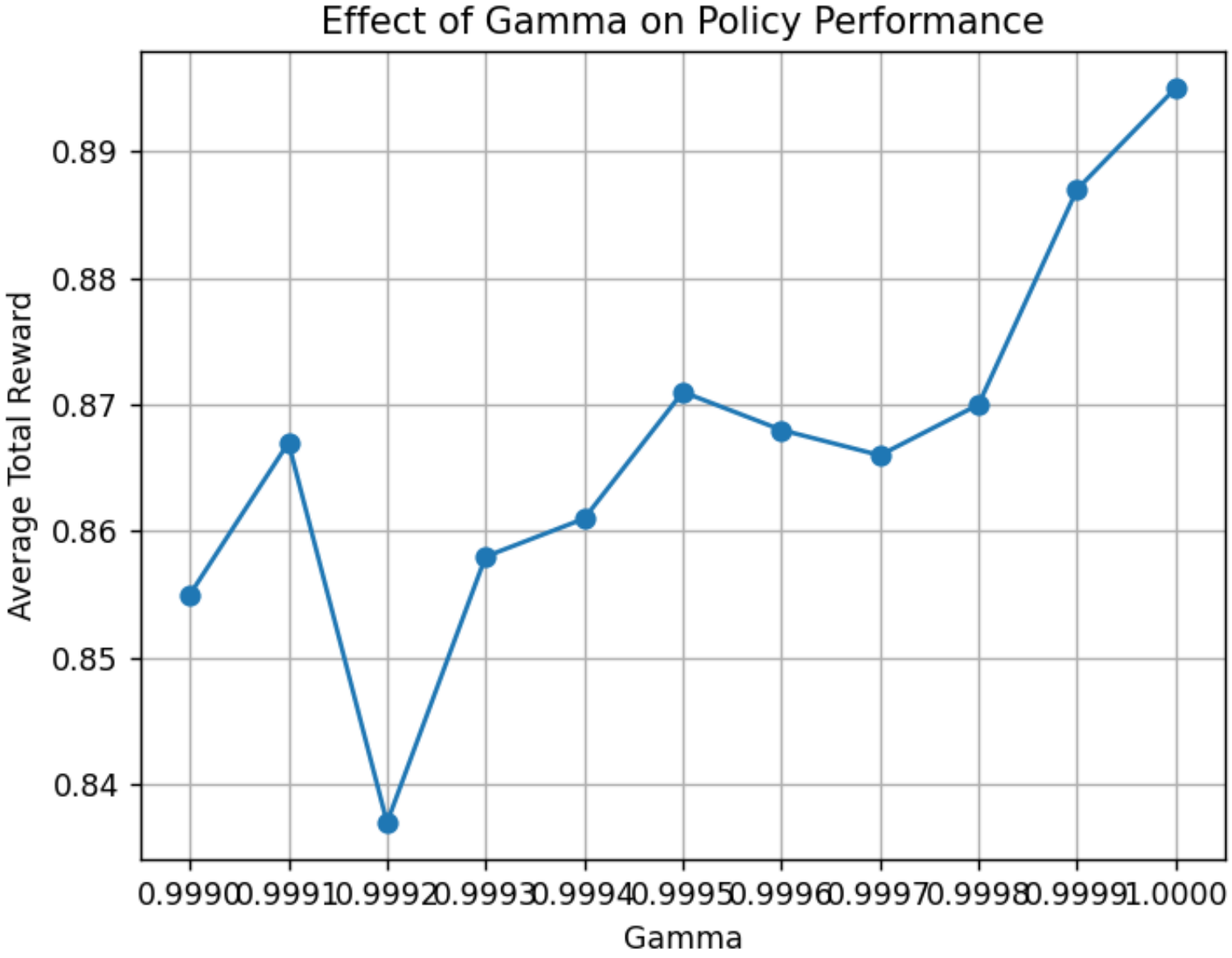
- Средняя награда колебалась на уровне около 0.74, и результаты были нестабильны при каждом запуске.
- Увеличение значений K и L (например, до 50 и выше) не приводило к значительным изменениям средней награды.
- Низкие значения γ (0.1–0.9) приводят к тому, что алгоритм чаще выбирает стратегии, ориентированные на краткосрочные вознаграждения, что негативно влияет на способность эффективно достигать цели.

Тестирование для гамма от 0.9 до 1.0:



- При γ от 0.9 до 1.0 наблюдалось значительное улучшение среднего вознаграждения. Средние награды при γ от 0.99 и выше стабильно превышали 0.85, достигая в некоторых случаях 0.87 и выше.

- Данные результаты показали, что более высокие значения γ позволяют лучше оценивать долгосрочные выгоды и приводят к более качественной стратегии.



3. Вывод

Результаты экспериментов показывают, что выбор значения γ в диапазоне от 0.999 до 1.0 обеспечивает более стабильное и высокое среднее вознаграждение для обученной политики. Это подтверждает, что для задач, где важно учитывать долгосрочные результаты, такие как Frozen Lake, стоит выбирать высокие значения γ, позволяющие алгоритму ориентироваться на достижение конечной цели, а не на промежуточные краткосрочные выгоды. Значения K и L влияют на качество политики в меньшей степени и после определенного порога увеличения этих параметров незначительно влияют на результат.

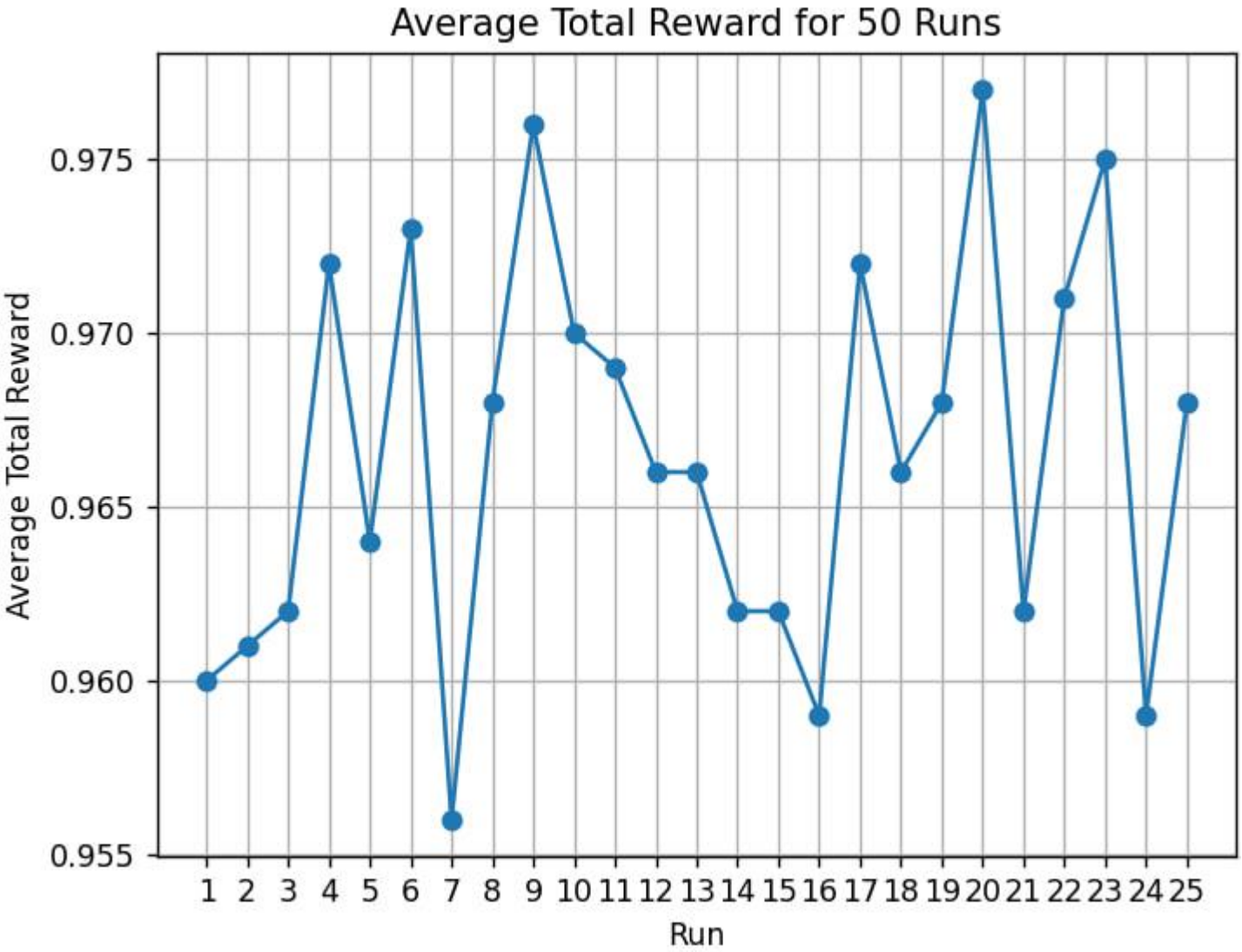
II. Анализ модифицированной оценки политики с инициализацией накопленных значений

1. Введение

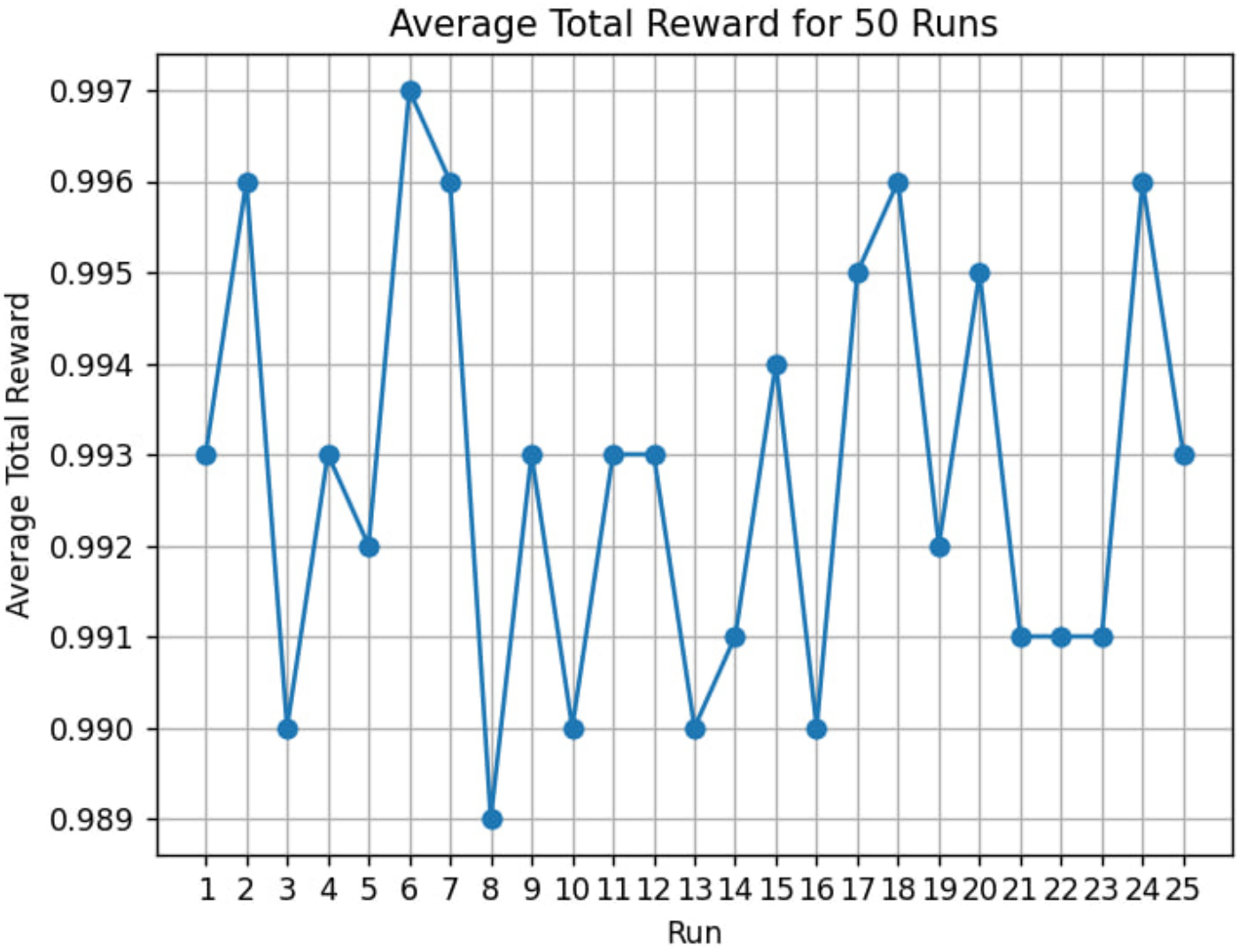
Целью второго задания является анализ изменения алгоритма для задачи Frozen Lake. В частности, мы модифицируем шаг **Policy Evaluation**, чтобы вместо инициализации **values** на каждом шаге нулями использовать значения, полученные на предыдущем шаге. Основной задачей является выяснение того, улучшает ли это подход стабильность и эффективность алгоритма, и если да, то насколько.

Введение сохранения значений **values** между итерациями позволяет алгоритму использовать ранее полученную информацию, что может ускорить сходимость и улучшить стабильность итоговой политики. Это помогает алгоритму лучше учитывать будущие состояния и позволяет быстрее "запоминать" полезные состояния.

2. Результаты



γ = **0.999**: Средняя награда стабильно находится между 0.95 и 0.98



γ = **0.9999**: Средняя награда стабильно достигала 0.99

3. Вывод

Использование предыдущих значений для **values** в шаге оценки политики позволяет алгоритму быстрее сходиться и достигать более стабильной политики. Это связано с тем, что инициализация на основе предыдущих значений ускоряет вычисления и снижает отклонения на каждом этапе.

III. Value Iteriation (первая часть)

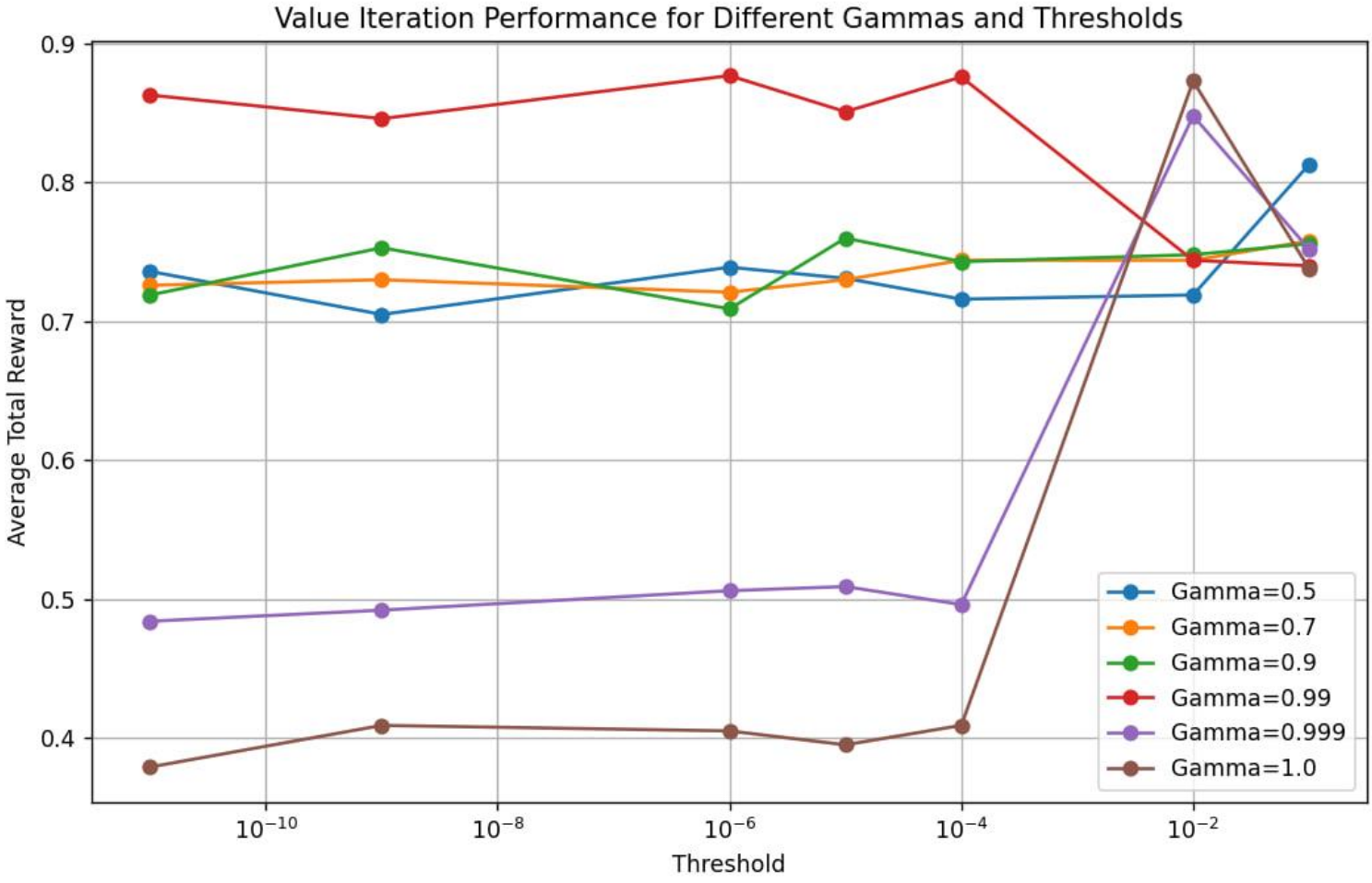
В рамках работы был реализован и протестирован алгоритм **Value Iteration** для решения

Frozen Lake. Главная цель эксперимента заключалась в исследовании эффективности алгоритма при разных значениях гиперпараметров.

Для алгоритма Value Iteration были исследованы два ключевых гиперпараметра:

- **Gamma (γ)**

- **Threshold**: пороговое значение для определения сходимости алгоритма. Threshold задает точность, с которой вычисляется значение состояния V(s) на каждой итерации. Чем меньше значение Threshold, тем точнее будут вычисления, однако потребуется больше итераций, чтобы достичь сходимости.



Исследования показали, что изменение значений gamma и threshold оказывает влияние на эффективность алгоритма Value Iteration. Результаты можно разделить на следующие категории:

- При значениях gamma в диапазоне **от 0.5 до 0.9** и threshold от **1e-11 до 1e-1** средняя награда держится примерно на уровне **0.75**. Это указывает

на стабильное выполнение алгоритма с учетом как ближайших, так и отдаленных наград.

- При значениях gamma **от 0.999 до 1.0** и threshold в диапазоне от **1e-11 до 1e-4** средняя награда держится на **0.4-0.5**. При увеличении threshold свыше **1e-4** начались скачки награды до значений **0.7-0.9**.
- Лучшее значение для gamma оказалось **0.99**. При этом значении алгоритм стабильно достигал награды до **0.9** при порогах, приближенных к нулю (до **1e-2**). При значениях threshold свыше **1e-2** наблюдались просадки в награде до **0.7**.

Таким образом, параметры gamma = 0.99 и threshold < 1e-2 позволили добиться наилучшего результата по стабильности и количеству наград.

IV. Value Iteriation (вторая часть)

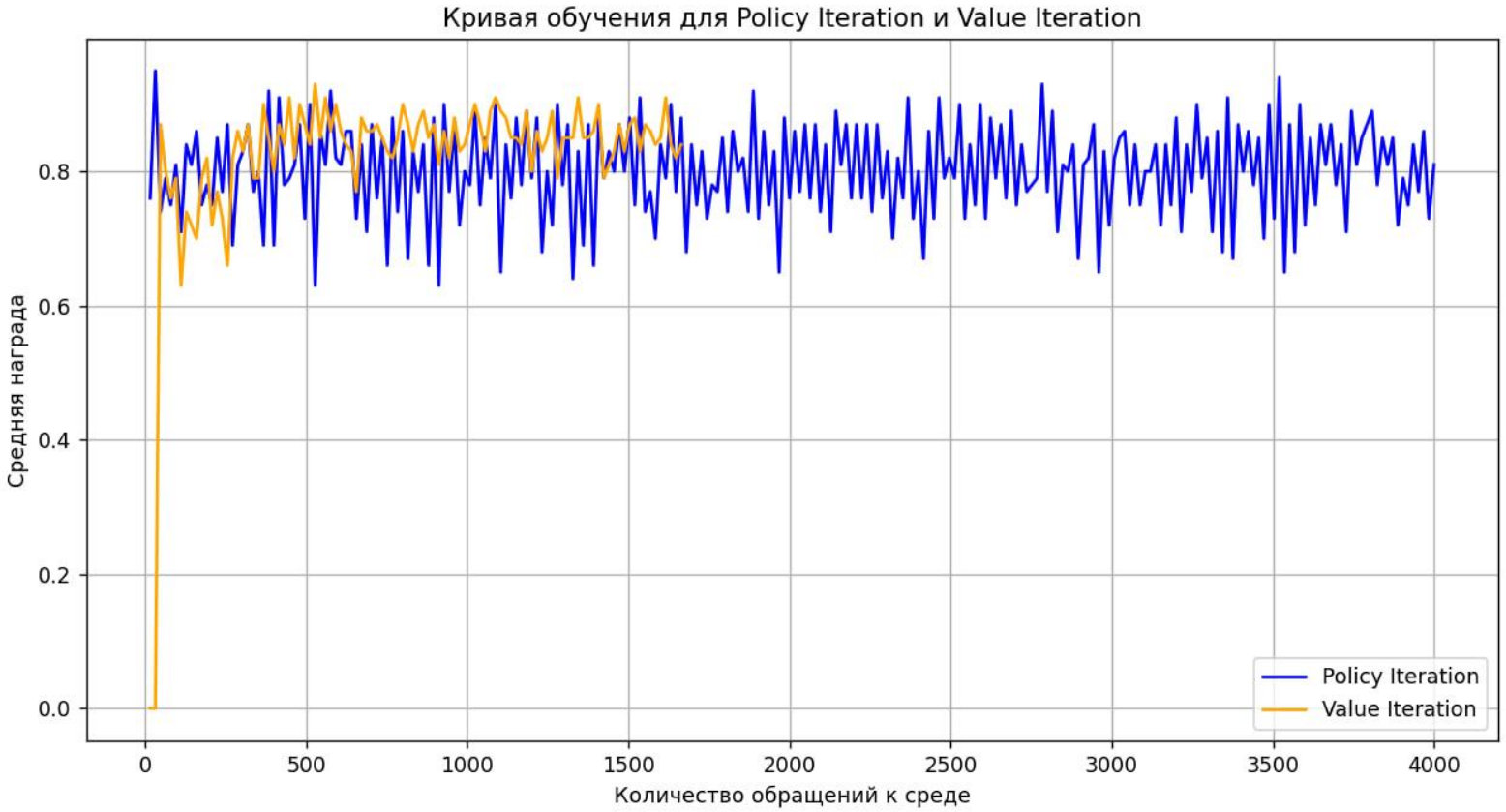
Целью эксперимента было исследовать и сравнить два алгоритма - Policy Iteration и Value Iteration - по их эффективности на среде Frozen Lake.

В Policy Iteration и Value Iteration структура циклов различается, поэтому количество внешних итераций не является адекватной метрикой для сравнения. Каждая итерация в этих алгоритмах **может занимать разное время и требует разного количества вычислений**.

В Policy Iteration два вложенных цикла: внешний цикл выполняет улучшение политики, а внутренний цикл — оценку текущей политики (проводит несколько итераций, пока значение не стабилизируется). В отличие от этого, Value Iteration состоит из одного основного цикла, который пересчитывает значения до тех пор, пока они не сойдутся.

Итак, **одна итерация Policy Iteration может быть “тяжелее”** (так как каждый внешний цикл включает вложенные оценки значений, часто с многократными обращениями к среде) **одной итерации Value Iteration**.

Из-за этой разницы переходим к сравнению через **количество обращений к среде**. Количество обращений к среде в обоих алгоритмах показывает, сколько раз каждый из них взаимодействовал с окружающей средой для получения информации о состоянии, переходе и награде. Это более **универсальная и справедливая метрика** для оценки затрат по времени и ресурсам, так как она не зависит от структуры циклов.



В ходе проведенных экспериментов мы установили максимальное количество итераций для **Policy Iteration** на уровне 250 и для **Value Iteration** на уровне 1000(но **Value Iteration** завершил работу на 104 итерации).

Value Iteration может быстрее достичь сходимости, поскольку он обновляет ценности всех состояний за каждую итерацию, используя текущее значение, чтобы вычислить новое значение. Если состояние достигает стабильного значения (например, когда изменения становятся меньше заданного порога), итерации могут завершиться даже при меньшем количестве шагов.

Policy Iteration показал более высокие максимальные значения вознаграждений (до 0.95), но с большим количеством колебаний. Policy Iteration может дать более высокие вознаграждения на некоторых этапах, но требует более осторожного управления.

Value Iteration стартовал с нуля, но затем стремительно поднимался и демонстрировал более стабильные результаты на более поздних шагах. Value Iteration в целом кажется более стабильным в долгосрочной перспективе, тогда как Policy Iteration имеет резкие колебания.