**DOI: 10.15514/ISPRAS-2025-37(4)-23**



# Извлечение знаний в ограниченной области для примеров состязательных атак «черного ящика»

[1, 2, 3] *К.С. Лукьянов, ORCID: 0009-0009-5235-2175 <lukianov@ispras.ru>*
[1] *А.И. Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*
[1, 3] *Д.Ю. Турдаков, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*
[3, 4] *М.А. Паутов, ORCID: 0000−0003−0438−6361 <pautov@airi.net>*

[1] *Институт системного программирования РАН,*
*Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

[2] *Московский физико-технический институт (НИУ),*
*Россия 117303, Москва, ул. Керченская, д.1 А, корп. 1.*

[3] *Исследовательский центр доверенного искусственного интеллекта ИСП РАН,*
*Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

[4] *AIRI, Россия 105064, Москва, р. 19, Нижний Сусальный пер., 5*

**Аннотация.** Устойчивость нейронных сетей к состязательным возмущениям в условиях «чёрного ящика» остаётся сложной проблемой. Большинство существующих методов атак требуют чрезмерного количества запросов к целевой модели, что ограничивает их практическую применимость. В данной работе мы предлагаем подход, в котором суррогатная модель-ученик итеративно обучается на неудачных попытках атак, постепенно изучая локальное поведение модели «чёрного ящика». Эксперименты показывают, что этот метод значительно сокращает количество необходимых запросов, сохраняя при этом высокую вероятность успеха атак.

**Ключевые слова:** состязательная атака черного ящика; извлечение знаний.

**Для цитирования:** Лукьянов К.С., Перминов А.И., Турдаков Д.Ю., Паутов М.А. Извлечение знаний в ограниченной области для примеров состязательных атак «черного ящика» Труды ИСП РАН, том 37, вып. 4, часть 2, 2025 г., стр. 133–146. DOI: 10.15514/ISPRAS–2025–37(4)–23.

**Благодарности:** Работа поддержана грантом, предоставленным Министерством экономического развития Российской Федерации в соответствии с Соглашением о предоставлении из федерального бюджета гранта в форме субсидии федеральному государственному бюджетному учреждению науки Институт системного программирования им. В.П. Иванникова Российской академии наук от 20 июня 2025 г. № 139-15-2025-011. Идентификатор соглашения 000000Ц313925P4G0002.

# Knowledge Distillation in Local-Region for Black-Box Adversarial Examples

[1, 2, 3] *K.S. Lukianov ORCID: 0009-0009-5235-2175 <lukianov@ispras.ru>*
[1] *A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*
[1, 3] *D.Y. Turdakov, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*
[3, 4] *M.A. Pautov, ORCID: 0000−0003−0438−6361 <pautov@airi.net>*

[1] *Institute of System Programming of the Russian Academy of Sciences, 25, A. Solzhenitsyn str., Moscow, 109004, Russia.*

[2] *Moscow Institute of Physics and Technology (National Research University), building 1, 1 A, Kerchenskaya str., Moscow, 117303, Russia.*

[3] *Research Center for Trusted Artificial Intelligence ISP RAS, 25, A. Solzhenitsyn str., Moscow, 109004, Russia.*

[4] *AIRI, 5, Nizhniy Susalny per., p. 19, Moscow, 105064, Russia.*

**Abstract.** The robustness of neural networks to adversarial perturbations in black-box settings remains a challenging problem. Most existing attack methods require an excessive number of queries to the target model, limiting their practical applicability. In this work, we propose an approach in which a surrogate student model is iteratively trained on failed attack attempts, gradually learning the local behavior of the black-box model. Experiments show that this method significantly reduces the number of queries required while maintaining a high attack success rate.

**Keywords:** black-box adversarial attack; knowledge distillation.

**For citation:** Lukianov K.S., Perminov A.I., Turdakov D.Y., Pautov M.A. Knowledge Distillation in Local-Region for Black-Box Adversarial Examples Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 4, part 2, 2025, pp. 133-146 (in Russian). DOI: 10.15514/ISPRAS-2025-37(4)-23.

## *1. Введение*

Устойчивость глубоких нейронных сетей к возмущениям входных данных является ключевым свойством, необходимым для их интеграции в различные области машинного обучения с повышенными требованиями к безопасности. Например, в прикладных задачах: автономного вождение, медицинской диагностики и финансовых технологиях вопросы безопасности имеют очень высокое значение. Хотя от нейронных сетей ожидается, что они будут выдавать схожий результат для близких по содержанию входов, давно известно, что они уязвимы к так называемым состязательным (*adversarial)* возмущениям [1], малым, специально сконструированным изменениям входных данных, которые не меняют семантику исходного объекта, но вынуждают модель выдавать заранее определённый результат. Большинство методов генерации состязательных атак нейронных сетей направлено на построение подобных возмущений, что демонстрирует, что в целом прогнозы нейронных сетей нельзя считать полностью надёжными.

Наиболее эффективные и незаметные атаки, как правило, требуют доступа к градиентам модели и поэтому имеют ограниченную практическую применимость [2-4]. Однако в реальных условиях модели машинного обучения нередко развёртываются в виде сервисов, доступных через прикладной интерфейс (*Application Programming Interface, API*). Такой режим работы, хотя и накладывает определённые ограничения на исследование робастности моделей класса MLaaS (*Machine Learning as a Service*), не исключает возможности генерации *состязательных* возмущений [5-8]. В частности, для «чёрного ящика» (когда доступа есть только к ответам модели, а не ко всей модели, в частности к ее весам) можно построить возмущение, оценивая градиенты в окрестности целевой точки [9-10], применяя методы случайного поиска [6] или используя перенос знаний для обучения вспомогательной модели

и проведения атаки в режиме «белого ящика» (когда у атакующего есть полный доступ к модели, включая ее веса и градиенты) [11-12].

Тем не менее, методы черного ящика зачастую требуют значительного числа запросов к целевой модели. В настоящей работе мы ставим следующий исследовательский вопрос: **как сократить число необходимых запросов для построения *состязательных* примеров для заданной модели черного ящика, решающей задачу классификации изображений с сотен и тысяч до десятков запросов?** Для ответа на него мы предлагаем метод *Mimic* — метод генерации атаки на черный ящик. В основе алгоритма находится идея переноса атаки, полученной на модели-заместителе (далее модель-ученик), построенной на итеративной локализованной дистилляции знаний, на модель черного ящика.

Методы атак, основанные на дистилляции знаний, в последние годы активно изучаются. Они применяются, например, для защиты интеллектуальной собственности: полученная в результате извлечения знаний моделью учеником используется для внедрения цифровых водяных знаков, позволяющих установить источник сгенерированного контента [13-16]. Данный подход также используется при атаках на модели черного ящика [11-12]. В рамках нашей работы мы предлагаем итеративное обучение серии моделей-учеников на расширяющемся наборе данных. Такой процесс позволяет каждой последующей модели ученику точнее воспроизводить поведение исходной модели черного ящика по сравнению с предыдущей и, в конечном счёте, вычислять *состязательное* возмущение для неё.

Наш вклад можно представить следующим образом:

1. Мы предлагаем методику Mimic – атаку с переносом на модель черного ящика, основанную на локальной дистилляции знаний. Алгоритм использует поведение целевой (учительской) модели в окрестности рассматриваемой точки в пространстве признаков и формирует набор моделей-учеников, воспроизводящих предсказания целевой модели на конечном множестве точек. Далее этот набор моделей используется для вычисления *состязательных* возмущений в режиме «белого ящика», которое после конечного числа итераций дистилляции переносится на модель-учителя.
2. Мы экспериментально демонстрируем эффективность предлагаемого подхода по сравнению с другими методами атаками черного ящика в задаче классификации изображений.

## *2. Обзор литературы*

В данном разделе приводится краткий обзор существующих атак черного ящика и применений дистилляции знаний.

### 2.1 Создание состязательных возмущений

В литературе методы построения состязательных возмущений для сетей с ограниченным доступом к архитектуре, весам или обучающему набору данных называются атаками в режиме «чёрного ящика». Такие атаки подразделяются на два типа: запросно-ориентированные (*query-based*) и переносные (*transfer-based*). В данной работе рассматривается переносимость состязательных атак из режима «белого ящика» в режим «чёрного ящика».

В запросно-ориентированных атаках злоумышленник использует выход целевой модели для построения состязательного примера. Один из подходов — оценка градиента модели по входным данным [17-20]. Однако подобные методы, как правило, требуют большого числа запросов к целевой модели, что делает их непрактичными. В противоположность им, при переносных атаках состязательный пример формируется путём атаки на одну или несколько вспомогательных (замещающих) моделей [7, 11, 21]. В литературе описаны подходы, улучшающие переносимость, включая использование аугментаций данных [22], методов

дополненного использованием градиентов [23], агрегации [24] или настройки направлений поиска [25].

Среди известных атак чёрного ящика можно выделить, например, ZOO [18] и NES [9]. Атака ZOO последовательно добавляет небольшие положительные или отрицательные возмущения к каждому пикселю целевого изображения, после чего отправляет измененные изображения на вход черного ящика и, опираясь на ответы модели, производит оценку градиента в окрестности исходного изображения. Атака NES действует аналогично, но вместо последовательного изменения пикселей использует набор случайно сгенерированных изображений для оценки градиента модели.

Существуют два варианта постановки задачи атаки на классификационные модели в режиме «чёрного ящика» в зависимости от выходных данных целевой модели: атака с мягкими метками (*soft-label*, когда на выходе вектор логитов, то есть полный вектор сделанных моделью оценок принадлежности входных данных каждому возможному классу) и атака с жёсткими метками (*hard-label*, когда на выходе индекс наиболее вероятного класса, по оценке модели).

На сегодняшний день среди запросно-ориентированных атак наилучшие показатели по среднему количеству запросов демонстрируют атаки Square Attack [6], NP-Attack [10] и MCG [26] (в случае мягких меток), а также GeoDA [27], Rays [28] и PSBA [29] (в случае жёстких меток). Среди переносных атак одной из наиболее эффективных считается Bayesian attack [11].

**Square Attack** выбирает область изображения для модификации и постепенно изменяет её в ходе работы алгоритма, случайным образом изменяя пиксели внутри этой области. **NP-Attack** использует нейросетевой предсказатель для направления поиска состязательных возмущений, прогнозируя выходные значения модели. **MCG** представляет собой мета-обучаемую атаку чёрного ящика, в которой мета-классификатор обобщает состязательные атаки на различные модели черного ящика.

Метод **GeoDA** рассматривает состязательную атаку как задачу геометрического исследования, фокусируясь на локальных свойствах границы решений. Алгоритм итеративно изменяет вход, получая жёсткие метки от модели, и уточняет границу между двумя классами для построения атаки. Метод **Rays** использует стратегию поиска на основе принципа трассировки лучей, постепенно продвигаясь в пространстве входных данных до пересечения с границей решений. Атака на черный ящик с прогрессивной шкалой границ **PSBA** (*Progressive-Scale Boundary Blackbox Attack*) применяет проективную оценку градиента по жёстким меткам, постепенно корректируя масштаб возмущений.

Атака **Bayesian** повышает переносимость состязательных примеров с помощью замещающей модели с байесовскими свойствами, используя стохастические веса и метод Монте-Карло-дропаута (*Monte Carlo Dropout*) для улучшенной оценки неопределённости. Этот метод относится к переносным атакам и предполагает наличие доступа к части обучающих данных, на которых обучалась модель черного ящика, либо к другому набору данных с аналогичным распределением. В нашей работе предполагается, что злоумышленник не имеет доступа к обучающим данным, поэтому мы не сравниваем наш подход с методами, требующими такого доступа.

## 2.2 Дистилляция знаний и защита от состязательных возмущений

Дистилляция знаний (*Knowledge Distillation*, KD) – это метод передачи качества работы большой модели-учителя на меньшую, облегчённую модель-ученика [30]. Пусть задана модель-учитель **T**. Обучение модели-ученика **S** из параметрического семейства моделей $S'$ формулируется как задача оптимизации:

$$S=\operatorname{argmin}_{S'} E_{(x,y)\sim D}[\alpha\, L(S'(x),y) + (1-\alpha)\, \tau^2\, KL(S'(x),T(x))]$$

где $D$ – набор данных для дистилляции, $L$ – функция потерь для задачи классификации, $KL$ – дивергенция Кульбака–Лейблера, $\alpha$ и $\tau$ – скалярные параметры.

Метод дистилляции знаний применяется в широком спектре задач: сжатие моделей [31-33], обеспечение приватности данных [16, 34-36], адаптация для больших языковых моделей [37-39] и диффузионных моделей [40-42].

В последние годы показано, что дистилляция знаний может повышать состязательную устойчивость к аддитивным возмущениям [43-45]. В отличие от больших моделей-учителей, которые могут достигать высокого уровня робастности, для небольших моделей-учеников сложно одновременно обеспечить высокую устойчивость и сохранение точности [45]. Для решения этой проблемы была предложена состязательно-устойчивая дистилляция, при которой в процессе дистилляции учитываются предсказания на чистых данных [46] или вероятностные векторы [47] робастной модели-учителя.

## *3. Постановка задач*

В этом разделе формально вводится постановка задач, вводятся обозначения, используемые в статье, и формулируются вопросы исследования. Введем формальные определения атак уклонения, отравления, защищенных моделей от соответствующих атак и частично защищенной модели.

### 3.1 Атаки на модель классификации

Атака уклонения (*evasion attack*) для моделей классификации – это процесс, при котором злоумышленник стремится изменить входные данные $x \in R^d$ таким образом, чтобы модифицированные данные $x' = x + \delta$ изменили прогноз модели на другой класс $f: R^d \to Y$, где $Y$ – множество классов.

Целью атаки уклонения является нахождение возмущения $\delta \in R^d$, удовлетворяющего следующим условиям:

1. $argmax_{y\in Y} f(x') \neq argmax_{y\in Y} f(x)$, то есть предсказанный класс изменяется после применения возмущения.
2. $\| \delta \|_p \leq \epsilon$, где $\|\cdot\|_p$ – $L_p$ норма (например, с $p = 2$ или $p = \infty$), а $\epsilon$ – заданное ограничение на величину возмущения, чтобы сохранялась правдоподобность измененного входа.

Таким образом, задача атаки уклонения может быть формализована как задача оптимизации: $argmax_{y\in Y} f(x') \neq argmax_{y\in Y} f(x)$, при условии $\| \delta \|_p \leq \epsilon$

Атаки уклонения делятся на два основных типа:

1. Белый ящик (white-box): злоумышленник обладает полным доступом к модели $f$, включая её параметры и градиенты.
2. Чёрный ящик (black-box): злоумышленник имеет ограниченный доступ к модели $f$, например, может наблюдать только выходы модели в ответ на определенные входы.

## *4. Методология*

В данном разделе описывается предложенный подход для генерации состязательных примеров для модели-учителя (модель черного ящика) с использованием дистилляции знаний.

## 4.1 Mimic: ученик следует за учителем

Для выполнения состязательной атаки на модель-учителя (модель черного ящика) T мы сначала применяем процедуру дистилляции знаний и получаем модель-ученика S в режиме «белого ящика». Обучающая выборка для модели-ученика формируется путём запросов к модели-учителю и сохранения её предсказаний для объектов из независимого (hold-out) набора данных $D_h$, не пересекающегося с обучающей выборкой учителя $D$.

$$(\mathcal{D}(T) : \mathcal{D}_h \cap \mathcal{D}(T) = \varnothing)$$

В нашей постановке в качестве $D_h$ используется тестовое подмножество набора данных учителя. В зависимости от режима дистилляции,

$$\mathcal{D}(S) = \{(x_i, T(x_i))\}_{i=1}^m \text{ or } \mathcal{D}(S) = \{(x_i, h(T, x_i))\}_{i=1}^m$$

Предполагая, что модель-ученик обладает достаточной обучающей способностью, мы обучаем её до совпадения с моделью-учителем на множестве $D(S)$.

**В режиме с мягкими метками** модель-ученик должна удовлетворять условиям:

$$\begin{cases} h(S, x_i) = h(T, x_i) = y_i, \\ \|S(x_i) - T(x_i)\|_\infty < \frac{\varepsilon}{4}, \end{cases}$$

где $\varepsilon > 0$ — заданная константа. Второе условие отражает способность модели-ученика уверенно имитировать поведение модели-учителя на $D(S)$.

**В режиме с жёсткими метками** модель-ученик должна удовлетворять условию:

$$h(S, x_i) = h(T, x_i) = y_i.$$

## 4.2 Mimic: атака ученика

На этом этапе мы описываем процедуру генерации состязательного примера для модели-ученика.

После обучения модели-ученика мы проводим на ней атаку в режиме «белого ящика», используя метод проецируемого градиентного спуска (*Projected Gradient Descent*, PGD) [48].

Пусть дан входной объект $x \in R^d$ класса $y \in [1, \ldots, K]$, который обе модели (учитель и ученик) классифицируют правильно. Метод PGD выполняется итеративно по градиенту для поиска состязательного примера $x'$ в δ-окрестности точки $x$, то есть в множестве $U_\delta(x) = \{x' : \|x - x'\|_2 \leq \delta\}$.

$$\begin{cases} x^{t+1} = \text{Proj}_{U_\delta(x)} \left[ x^t + \alpha \text{sign} \nabla_{x^t} L(S, x^t, y) \right], \\ x^1 = x, \\ x' = x^M, \end{cases}$$

где $\alpha > 0$ – величина шага оптимизации, $M$ – число итераций PGD, $\text{Proj}_{U\delta}(x)$ – оператор проекции в замкнутую δ-окрестность точки $x$, а $L(S, x_t, y)$ – функция потерь модели S на примере ($x_t$, $y$). В нашей постановке $L(S, x_t, y)$ – это функция кросс-энтропии.

**Замечание.** Для повышения вычислительной эффективности атаки мы генерируем не один, а пакет из l атакующих примеров $\{x_1', \ldots, x_l'\}$ для модели ученика. Псевдокод предлагаемого метода представлен в Алгоритме 1. Обратите внимание, что мы используем атаку PGD, благодаря её простоте; однако наш подход не ограничен конкретным типом атаки белого ящика.

## *5. Эксперименты*

В данном разделе приводится техническое описание проведённых экспериментов. В частности, мы описываем используемые наборы данных и архитектуры моделей, методы базового сравнения, а также методологию проведения экспериментов.

### 5.1 Условия обучения

В наших экспериментах в качестве обучающих наборов данных для модели-учителя используются CIFAR-10 и CIFAR-100 [49]. В качестве модели-учителя $T$ используется ResNet50 [50], которая обучалась в течение 250 эпох для достижения высокой точности классификации (92% для CIFAR-10 и 75% для CIFAR-100). Для обучения модели-учителя используется оптимизатор SGD с коэффициентом обучения 0.1, коэффициентом регуляризации (weight decay) $10^{-4}$ и моментумом 0.9.

### 5.2 Настройка Mimic

В качестве архитектуры модели ученика используется SmallCNN. Мы проводим PGD-атаку на модели-ученике со следующими параметрами: количество шагов PGD M = 10, шаг градиента $\alpha$ = 0.005, порог расстояния $\delta$ = 0.05. В случае мягких меток (soft-label) используется функция потерь L1 Loss, для жёстких меток (hard-label) – CrossEntropyLoss. Архитектура модели SmallCNN и описание алгоритма в виде псевдокода представлены в алгоритме 1. Обращаем внимание, что так как в литературе принято опираться на поиск первого атакующего изображения, то в таком случае необходима досрочная остановка в алгоритме. Алгоритм 1 соответствует варианту, который использовался для получения результатов представленных в таблицах 2-5. Однако, если есть необходимость получить большую серию атак при ограниченном числе запросов, то в алгоритме необходимо убрать досрочные выходы из цикла с помощью операторов выхода из цикла **`break`**.

### 5.3 Методы черного ящика для сравнения

В этом подразделе мы кратко перечисляем набор методов, с которыми мы сравниваем наш подход.

В случае мягких меток (soft-label) мы оцениваем MMAttack по сравнению с алгоритмами ZOO [18] и NES [9] как основными конкурентами. Среди методов атак типа «черного ящика», основанных на случайном поиске, мы выбираем Square attack [6] как современный метод с наименьшим средним числом запросов для проведения атаки. В группе методов, использующих оценку градиента, NP-Attack [10] является одним из наиболее эффективных. В категории комбинированных методов мы выбираем метод MCG [26]. Гиперпараметры методов, использованных в экспериментах, приведены в табл. 1. Отметим, что алгоритм MCG изначально предполагает обучение на данных, распределение которых близко к распределению данных учительской модели, что в общем случае может быть неизвестно. В данном исследовании мы подчеркиваем, что наш метод не имеет такого ограничения.

В случае жестких меток (hard-label) мы оцениваем MMAttack по сравнению с GeoDA [27] как современным методом с наименьшим средним числом запросов; Rays [28] является одним из наиболее эффективных методов атаки на основе случайного поиска, а PSBA [29] – одним из наиболее эффективных методов в группе атак, использующих оценку градиента.

### 5.4 Результаты экспериментов

В экспериментах методы ZOO, NES и Square Attack выполнялись 100 раз с различными случайными инициализациями, а методы NP-Attack, MCG и MMA – 30 раз.

В табл. 2 представлен сравнительный анализ существующих методов атак черного ящика атак на наборах данных CIFAR-10 и CIFAR-100 в режиме мягких меток. В таблице приведено

среднее количество запросов (AQN), необходимое для генерации первого атакующего примера для моделей черного ящика и средний успех атаки (ASR), для оценки успешных попыток атаки; символ δ обозначает максимально возможное расстояние до целевой точки в терминах нормы $L_\infty$; лучшие результаты выделены **жирным шрифтом**. Следует отметить, что, если результаты метода, приведённые в литературе, не совпадают с результатами, полученными в нашей реализации, в таблице приводится результат с наименьшим средним количеством запросов. Результаты, взятые из литературы, отмечены как (**Lit**).

Данные для метода MCG на наборе CIFAR-100 были взяты из [26]; данные для методов MCG и NP-Attack на наборе CIFAR-10 – из работы [51].

В табл. 3 приведено сравнение существующих методов атак черного ящика атак на наборах данных CIFAR-10 и CIFAR-100 в режиме жестких меток. Данные для метода Rays на CIFAR-10 были взяты из [28]; данные для методов GeoDA на CIFAR-10 – из работы [51].

**Input:** Черный ящик, который является моделью учителем $T$, входной объект $x$ класса $y$, Порог расстояния $\delta$, шаг градиента $\alpha$, максимальное количество итераций PGD $M$, максимальное количество итераций дистилляции $N$, начальный случайный поднабор данных $\mathcal{D}_h$, количество состязательных примеров $l$ для генерации серии моделей учеников $S_2, \ldots, S_N$, *если не будет найдена успешная атака на модель черного ящика*.

**Ouput:** Набор моделей-учеников $\{S_1, \ldots, S_N\}$, набор $AE(T)$ состязательных примеров для модели $T$.

$z \leftarrow (x, T(x))$ // вычислить логиты модели T в целевой точке

$\mathcal{D}(S) \leftarrow \{(x_i, T(x_i))\}_{i-1}$; $x_i$ \in $\mathcal{D}_h$ // сформировать тренировочный набор $\mathcal{D}(S)$

$\mathcal{D}(S_1) \leftarrow \mathcal{D}(S) \cup z$ // инициализация тренировочного набора для первой модели-ученика $S_1$

$AE(T) \leftarrow \varnothing$ // инициализация набора состязательных примеров для модели $T$

**for** $i$ = 1 **to** $N$ **do**

  $S_i \leftarrow$ train ($\mathcal{D}(S_i)$) // обучение модели-ученика $S_i$ на наборе $\mathcal{D}(S_i)$

  **for** $j$ = 1 **to** $l$ **do**

    $(x'_j, y'_j) \leftarrow PGD(\alpha, \delta, M, S_i, (x, y))$ // вычислить состязательный пример для ученика $S_i$

    **if** $h(S_i, x'_j) = h(T, x'_j)$ **then** // добавить успешную атаку

      $AE(T) \leftarrow AE(T) \cup \{(x'_j, y'_j)\}$ // в набор атакующих примеров для модели $T$

      **break**

  $\mathcal{D}(S_{i+1}) \leftarrow \mathcal{D}(S_i) \cup \{(x'_j, T(x'_j))\}$ // обновить тренировочный набор для модели $S_{i+1}$

  **if** $AE(T)$ != $\varnothing$ **then**

    **break**

**return** $\{S_1, \ldots, S_N\}$, $AE(T)$

```
SmallCNN(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace)
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU(inplace)
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace)
    (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Linear(in_features=4096, out_features=512, bias=True)
    (1): ReLU(inplace)
    (2): Linear(in_features=512, out_features=10 or 100, bias=True)
```

```
    )
)
```

*Алгоритм 1. Атака имитации модели.*
*Algorithm 1. Model Mimic Attack.*

*Табл. 1. Гиперпараметры атак черного ящика.*
*Table 1. Hyperparameters of black-box attacks.*

| **Method** | **Hyperparameters** |
|---|---|
| ZOO | $\epsilon = 0.05$<br>NUM_ITERATIONS = 5000<br>LEARNING_RATE = 0.01 |
| NES | $\epsilon = 0.1$<br>NUM_SAMPLES = 50<br>NUM_ITERATIONS = 300<br>$\sigma$ = 0.01<br>$\alpha$ = 0.03 |
| SQUARE | $\epsilon = 0.1$<br>NUM_QUERIES = 5000<br>P_INIT = 0.8 |
| NP | $\epsilon = 0.05$<br>NUM_ITERATIONS = 1000<br>LEARNING_RATE = 0.01 |
| MCG | DOWN_SAMPLE_X = 1<br>DOWN_SAMPLE_Y = 1<br>FINETUNE_GROW = TRUE<br>FINETUNE_RELOAD = TRUE<br>FINETUNE_PERTURBATION = TRUE |
| GeoDA | $\epsilon = 0.05$<br>TOL = 0.001<br>ALPHA = 0.002<br>MU = 0.6<br>GRAD_ESTIMATOR_BATCH_SIZE = 40<br>SIGMA = 0 |
| Rays | $\epsilon = 0.05$<br>STEP_SIZE = 0.01<br>MAX_ITERATIONS = 5000 |
| PSBA | $\epsilon = 0.05$<br>NUM_SAMPLES = 20<br>NUM_ITERATIONS = 50<br>SCALE_FACTOR = 0.5 |

*Табл. 2. Сравнение методов атаки «черный ящик», настройка «мягких меток». Стрелка ↓ указывает на то, что меньшее значение метрики лучше, стрелка ↑ указывает, что лучше большее значение.*
*Table 2. Comparison of black-box attack methods, soft-label setting. ↓ indicates that a lower metric value is better, while ↑ indicates that a higher value is better.*

| $\mathcal{D}$ | **Attack method** | $\delta$ | AQN (↓) | ASR (%;↑) |
|---|---|---|---|---|
| CIFAR-10 | ZOO | 0.05 | $\geq 3 \times 10^5$ | 87.3 |
| | NES | 0.1 | 3578 | 78.8 |
| | Square | 0.1 | 368 | 83.2 |
| | NP-Attack (**Lit**) | 0.05 | 500 | 96 |
| | MCG (**Lit**) | 0.1 | 130 | **100** |
| | MMAttack (**ours**) | 0.05 | **32.8** | **99** |
| | ZOO | 0.05 | $\geq 3 \times 10^5$ | 85.0 |

| | | | | |
|---|---|---|---|---|
| CIFAR-100 | NES | 0.1 | 4884 | 81.1 |
| | SQUARE | 0.1 | 193 | 72.8 |
| | NP-ATTACK | 0.05 | 325 | 97.5 |
| | MCG (**LIT**) | 0.1 | 48 | **100** |
| | MMATTACK (**OURS**) | 0.05 | **24** | **100** |

*Табл. 3. Сравнение методов атаки «черный ящик», настройка «жесткой меток». Стрелка ↓ указывает на то, что меньшее значение метрики лучше, стрелка ↑ указывает на то, что большее значение лучше.*
*Table 3. Comparison of black-box attack methods, hard-label setting. ↓ indicates that a lower metric value is better, while ↑ indicates that a higher value is better.*

| $\mathcal{D}$ | ATTACK METHOD | $\delta$ | AQN (↓) | ASR (%;↑) |
|---|---|---|---|---|
| CIFAR-10 | GEODA (**LIT**) | 0.05 | 500 | 53 |
| | RAYS (**LIT**) | 0.05 | 574 | **98** |
| | PSBA | 0.05 | 2000 | 89.5 |
| | MMATTACK (**OURS**) | 0.05 | **69.2** | 92.3 |
| CIFAR-100 | GEODA | 0.05 | 317 | 82 |
| | RAYS | 0.05 | 2013 | 92 |
| | PSBA | 0.05 | 2000 | 87.5 |
| | MMATTACK (**OURS**) | 0.05 | **48.1** | **95.4** |

Как видно, метод MMAttack с моделью имитатором SmallCNN превосходит конкурентов по метрике AQN в обоих режимах (soft-label, hard-label). В исследовании влияния гиперпараметров оценивается эффективность метода MMAttack в зависимости от архитектуры модели-ученика, которые являются моделями белого ящика.

## 5.5 Влияние гиперпараметров

Обратите внимание, что успех нашей атаки черного ящика критически зависит от архитектуры модели ученика, который используется как белый ящик. С одной стороны, модель ученика не обязана иметь большое количество обучаемых параметров, так как это подразумевает несколько итераций повторного обучения. С другой стороны, она должна обладать достаточной обучающей способностью, чтобы имитировать поведение модели черного ящика вблизи целевой точки. В таблицах 4 и 5 представлены значения AQN для различных пар моделей учитель и ученик на наборе данных CIFAR-10. (тут таблицы соответствуют разным постановкам мягким и жестких меток, что указано в названиях таблиц, а параметры, которые варьируются одинаковые).

*Табл. 4. Влияние гиперпараметров на производительность Mimic в случае мягкой метки. Стрелка ↓ указывает на то, что меньшее значение метрики лучше.*
*Table 4. Impact of hyperparameters on the performance of the Mimic in soft-label case. ↓ indicates that a lower metric value is better.*

| $T$ | $S$ | $\lvert\mathcal{D}(S_1)\rvert$ | $l$ | AQN (↓) |
|---|---|---|---|---|
| RESNET101 | RESNET34 | 800 | 400 | 4520 |
| RESNET50 | RESNET34 | 600 | 400 | 4160 |
| RESNET101 | RESNET18 | 600 | 300 | 1560 |
| RESNET50 | RESNET18 | 600 | 200 | 530 |
| RESNET34 | RESNET18 | 300 | 30 | 455 |
| RESNET50 | VGG19 | 200 | 30 | 500 |
| RESNET50 | VGG16 | 200 | 30 | 440 |
| RESNET101 | SMALLCNN | 10 | 10 | 37.7 |

| | | | | |
|---|---|---|---|---|
| ResNet101 | SmallCNN | 10 | 10 | 32.8 |
| ResNet34 | SmallCNN | 10 | 10 | 34 |
| VGG19 | SmallCNN | 10 | 10 | 95.6 |
| VGG16 | SmallCNN | 10 | 10 | 40.2 |
| ResNet50 | SmallCNN | 5 | 5 | – |

*Табл. 5. Влияние гиперпараметров на производительность Mimic в случае жестких меток. Стрелка ↓ указывает на то, что меньшее значение метрики лучше.*
*Table 5. Impact of hyperparameters on the performance of the Mimic in hard-label case. ↓ indicates that a lower metric value is better.*

| *T* | *S* | $\lvert\mathcal{D}(S_1)\rvert$ | *l* | AQN (↓) |
|---|---|---|---|---|
| ResNet101 | ResNet34 | 800 | 400 | 4216 |
| ResNet50 | ResNet34 | 600 | 400 | 3347 |
| ResNet101 | ResNet18 | 600 | 300 | 1518 |
| ResNet50 | ResNet18 | 600 | 200 | 1536 |
| ResNet34 | ResNet18 | 300 | 30 | 386 |
| ResNet50 | VGG19 | 200 | 30 | 1055 |
| ResNet50 | VGG16 | 200 | 30 | 690 |
| ResNet101 | SmallCNN | 10 | 10 | 60.7 |
| ResNet101 | SmallCNN | 10 | 10 | 69.2 |
| ResNet34 | SmallCNN | 10 | 10 | 63.9 |
| VGG19 | SmallCNN | 10 | 10 | 184 |
| VGG16 | SmallCNN | 10 | 10 | 30.3 |
| ResNet50 | SmallCNN | 5 | 5 | – |

Помимо среднего числа запросов, мы указываем размер исходного обучающего набора данных $D(S_1)$ модели ученика и количество создаваемых для неё атакующих примеров. В табл. 4 и 5 обозначения $T$ и $S$ соответствуют архитектурам учителя и ученика моделей. Мы обнаружили, что чем проще архитектура модели ученика, тем меньше запросов к модели учителю требуется для успешного проведения атаки.

Начальный размер набора данных, $|D(S_1)|$, представляет собой количество случайных точек данных, включаемых в исходный обучающий набор модели ученика в режиме белого ящика. Как видно из табл. 4 и 5, чем сложнее модель ученика, тем больше должен быть этот параметр. То же самое относится к количеству атакующих примеров для модели ученика, l.

Следует отметить, что отсутствует значение $AQN$, соответствующее $|D(S_1)|=5$. Это связано с тем, что используемый алгоритм не способен найти ни одного атакующего примера для модели учителя черного ящика до достижения максимального числа итераций.

## 6. Заключение и будущая работа

В данной работе мы предлагаем **Mimic** – атаку черного ящика. Для проведения атаки мы применяем **локальную дистилляцию знаний** с целью получения модели ученика, которая фактически является функциональной копией модели черного ящика. Затем мы выполняем модель белого ящика атакующую процедуру на модели ученика.

Экспериментально мы демонстрируем, что успешный атакующий пример может быть найден при небольшом числе запросов к целевой модели, что делает предложенный подход применимым на практике. Возможные направления дальнейших исследований включают адаптацию метода для других областей, в частности для атак на крупные языковые модели или атака на графовые модели.

## Список литературы / References

[1]. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In 2nd International Conference on Learning Representations.
[2]. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. CoRR, abs/1412.6572.
[3]. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.
[4]. Carlini, N., & Wagner, D. A. (2016). Towards Evaluating the Robustness of Neural Networks. In 2017 IEEE Symposium on Security and Privacy (SP), 39–57.
[5]. Chen, J., Jordan, M. I., & Wainwright, M. J. (2020). Hopskipjumpattack: A query-efficient decision-based attack. In 2020 IEEE Symposium on Security and Privacy (SP), 1277–1294. IEEE.
[6]. Andriushchenko, M., Croce, F., Flammarion, N., & Hein, M. (2020). Square attack: a query-efficient black-box adversarial attack via random search. In European Conference on Computer Vision, 484–501. Springer.
[7]. Qin, Y., Xiong, Y., Yi, J., & Hsieh, C.-J. (2023). Training meta-surrogate model for transferable adversarial attack. In Proceedings of the AAAI Conference on Artificial Intelligence, 37(8), 9516–9524.
[8]. Vo, Q. V., Abbasnejad, E., & Ranasinghe, D. (2024). BruSLeAttack: A Query-Efficient Score-Based Black-Box Sparse Adversarial Attack. In The Twelfth International Conference on Learning Representations.
[9]. Ilyas, A., Engstrom, L., Athalye, A., & Lin, J. (2018). Black-box adversarial attacks with limited queries and information. In International Conference on Machine Learning, 2137–2146. PMLR.
[10]. Bai, Y., Zeng, Y., Jiang, Y., Wang, Y., Xia, S.-T., & Guo, W. (2020). Improving query efficiency of black-box adversarial attack. In Computer Vision—ECCV 2020: 16th European Conference, 101–116. Springer.
[11]. Li, Q., Guo, Y., Zuo, W., & Chen, H. (2023). Making substitute models more Bayesian can enhance transferability of adversarial examples. arXiv preprint arXiv:2302.05086.
[12]. Gubri, M., Cordy, M., Papadakis, M., Le Traon, Y., & Sen, K. (2022). LGV: Boosting adversarial example transferability from large geometric vicinity. In European Conference on Computer Vision, 603–618. Springer.
[13]. Yuan, X., Ding, L., Zhang, L., Li, X., & Wu, D. O. (2022). ES attack: Model stealing against deep neural networks without data hurdles. IEEE Transactions on Emerging Topics in Computational Intelligence, 6(5), 1258–1270. IEEE.
[14]. Lukas, N., Zhang, Y., & Kerschbaum, F. (2019). Deep neural network fingerprinting by conferrable adversarial examples. arXiv preprint arXiv:1912.00888.
[15]. Kim, B., Lee, S., Lee, S., Son, S., & Hwang, S. J. (2023). Margin-based neural network watermarking. In International Conference on Machine Learning, 16696–16711. PMLR.
[16]. Pautov, M., Bogdanov, N., Pyatkin, S., Rogov, O., & Oseledets, I. (2024). Probabilistically Robust Watermarking of Neural Networks. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), 4778–4787.
[17]. Bhagoji, A. N., He, W., Li, B., & Song, D. (2018). Practical black-box attacks on deep neural networks using efficient query mechanisms. In Proceedings of the European Conference on Computer Vision (ECCV), 154–169.
[18]. Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C.-J. (2017). ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 15–26.
[19]. Ilyas, A., Engstrom, L., & Madry, A. (2019). Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors. In International Conference on Learning Representations.
[20]. Guo, C., Gardner, J., You, Y., Wilson, A. G., & Weinberger, K. (2019). Simple black-box adversarial attacks. In International Conference on Machine Learning, 2484–2493. PMLR.
[21]. Liu, Y., Chen, X., Liu, C., & Song, D. (2022). Delving into Transferable Adversarial Examples and Black-box Attacks. In International Conference on Learning Representations.
[22]. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., & Yuille, A. L. (2019). Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2730–2739.
[23]. Wu, D., Wang, Y., Xia, S.-T., Bailey, J., & Ma, X. (2020). Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. In International Conference on Learning Representations.

[24]. Liu, X., Zhong, Y., Zhang, Y., Qin, L., & Deng, W. (2023). Enhancing generalization of universal adversarial perturbation through gradient aggregation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 4435–4444.
[25]. Yang, X., Lin, J., Zhang, H., Yang, X., & Zhao, P. (2023). Improving the transferability of adversarial examples via direction tuning.
[26]. Yin, F., Zhang, Y., Wu, B., Feng, Y., Zhang, J., Fan, Y., & Yang, Y. (2023). Generalizable black-box adversarial attack with meta learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(3), 1804–1818. IEEE.
[27]. Rahmati, A., Moosavi-Dezfooli, S.-M., Frossard, P., & Dai, H. (2020). Geoda: a geometric framework for black-box adversarial attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8446–8455.
[28]. Chen, J., & Gu, Q. (2020). Rays: A ray searching method for hard-label adversarial attack. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1739---1747.
[29]. Zhang, J., Li, L., Li, H., Zhang, X., Yang, S., & Li, B. (2021). Progressive-scale boundary blackbox attack via projective gradient estimation. In International Conference on Machine Learning, 12479–12490. PMLR.
[30]. Hinton, G. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.
[31]. Sun, S., Cheng, Y., Gan, Z., & Liu, J. (2019). Patient Knowledge Distillation for BERT Model Compression. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 4323–4332.
[32]. Wang, J., Bao, W., Sun, L., Zhu, X., Cao, B., & Philip, S. Y. (2019). Private model compression via knowledge distillation. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01)_1197.
[33]. Li, T., Li, J., Liu, Z., & Zhang, C. (2020). Few sample knowledge distillation for efficient network compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14639–14647.
[34]. Lyu, L., & Chen, C.-H. (2020). Differentially private knowledge distillation for mobile analytics. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1809–1812.
[35]. Chourasia, R., Enkhtaivan, B., Ito, K., Mori, J., Teranishi, I., & Tsuchida, H. (2022). Knowledge Cross-Distillation for Membership Privacy. Proceedings on Privacy Enhancing Technologies.
[36]. Galichin, A. V., Pautov, M., Zhavoronkin, A., Rogov, O. Y., & Oseledets, I. (2024). GLiRA: Black-Box Membership Inference Attack via Knowledge Distillation. arXiv preprint arXiv:2405.07562.
[37]. McDonald, D., Papadopoulos, R., & Benningfield, L. (2024). Reducing LLM hallucination using knowledge distillation: A case study with Mistral large and MMLU benchmark. Authorea Preprints.
[38]. Gu, Y., Dong, L., Wei, F., & Huang, M. (2024). MiniLLM: Knowledge distillation of large language models. In The Twelfth International Conference on Learning Representations.
[39]. Kang, M., Lee, S., Baek, J., Kawaguchi, K., & Hwang, S. J. (2024). Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. Advances in Neural Information Processing Systems, 36.
[40]. Huang, T., Zhang, Y., Zheng, M., You, S., Wang, F., Qian, C., & Xu, C. (2024). Knowledge diffusion for distillation. Advances in Neural Information Processing Systems, 36.
[41]. Yao, X., Lu, F., Zhang, Y., Zhang, X., Zhao, W., & Yu, B. (2024). Progressively Knowledge Distillation via Re-parameterizing Diffusion Reverse Process. In Proceedings of the AAAI Conference on Artificial Intelligence, 38(15), 16425–16432.
[42]. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., & Park, T. (2024). One-step diffusion with distribution matching distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6613–6623.
[43]. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE Symposium on Security and Privacy (SP), 582-597. IEEE.
[44]. Kuang, H., Liu, H., Wu, Y., Satoh, S., & Ji, R. (2024). Improving adversarial robustness via information bottleneck distillation. Advances in Neural Information Processing Systems, 36.
[45]. Huang, B., Chen, M., Wang, Y., Lu, J., Cheng, M., & Wang, W. (2023). Boosting accuracy and robustness of student models via adaptive adversarial distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 24668–24677.

[46]. Goldblum, M., Fowl, L., Feizi, S., & Goldstein, T. (2020). Adversarially robust distillation. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(04), 3996–4003.
[47]. Zi, B., Zhao, S., Ma, X., & Jiang, Y.-G. (2021). Revisiting adversarial robustness distillation: Robust soft labels make student better. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 16443–16452.
[48]. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. In International Conference on Learning Representations.
[49]. Krizhevsky, A., Hinton, G., & others. (2009). Learning multiple layers of features from tiny images. Toronto, ON, Canada.
[50]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.
[51]. Zheng, M., Yan, X., Zhu, Z., Chen, H., & Wu, B. (2025). BlackboxBench: A comprehensive benchmark of black-box adversarial attacks. IEEE Transactions on Pattern Analysis and Machine Intelligence.

## *Информация об авторах / Information about authors*

Кирилл Сергеевич ЛУКЬЯНОВ – исследователь центра доверенного искусственного интеллекта ИСП РАН; аспирант МФТИ. Научные интересы: методы доверенного ИИ, федеративное обучение, многокритериальная оптимизация, AutoML.

Kirill Sergeevich LUKIANOV – Researcher at the Center for Trusted Artificial Intelligence of the Institute for System Programming of the Russian Academy of Sciences; postgraduate student at Moscow Institute of Physics and Technology. Research interests: trusted AI methods, federated learning, multi-objective optimization, AutoML.

Андрей Игоревич ПЕРМИНОВ является аспирантом института системного программирования РАН. Научные интересы: нейросетевая обработка данных, цифровая обработка изображений, методы доверенного искусственного интеллекта.

Andrey Igorevich PERMINOV is a postgraduate student at the Institute of System Programming of the RAS. Research interests: neural network data processing, digital image processing, trusted artificial intelligence.

Денис Юрьевич ТУРДАКОВ – кандидат физико-математических наук, заведующий отделом ИСП РАН. Научные интересы: анализ социальных сетей, анализ текста, извлечение информации, обработка больших данных, методы доверенного искусственного интеллекта.

Denis Yurievich TURDAKOV – Cand. Sci. (Phys.-Math.), head of department ISP RAS. Research interests: social network analysis, text mining, information extraction, big data, trusted artificial intelligence.

Михаил Александрович ПАУТОВ– кандидат компьютерных наук, научный сотрудник AIRI, Центра доверенного искусственного интеллекта ИСП РАН. Сфера научных интересов: линейная алгебра, теория больших отклонений, доказуемая устойчивость нейронных сетей, цифровые водяные знаки.

Mikhail Aleksandrovich PAUTOV – Cand. Sci. (Phys.-Math,), research scientist at AIRI and ISP RAS Research Center for Trusted Artificial Intelligence. His research interests include linear algebra, large deviations theory, certified robustness and digital watermarking.