Segment Anything 2 Adversarial Robustness
Project Report

Motivation
Модель SAMv1 уже нашла применение более чем в 4000 статьях, что делает вопрос устойчивости новой модели этого семейства особенно актуальным. Кроме того, атака на модель может быть полезной, позволяя понять слабости модели и найти способы для ее улучшения.

Objective
Цель проекта заключается в исследовании устойчивости модели SAMv2 к состязательным атакам.

Segment Anything 2 model overview
SAMv1 (2023) является Foundation Multimodal моделью, решающая promptable task в задаче сегментации и использующая в качестве промптов пользовательские вводы, такие как клики мышью в виде точек и назначенного лейбла, сегментационные маски, а также bounding box-ы.
Недавно, а именно 1-го августа 2024, вышла официальная модель SAMv2, позволяющая пользователю интерактивно работать с видео и открывающая еще больше возможностей в задачах AR/VR, robotics, autonomous vehicles, video editing и многие другие. Это произошло благодаря добавлению новых компонентов в архитектуру модели, а именно memory attention, memory encoder и memory bank.

Adversarial Attacks
В нашем проекте нам были предложены несколько разновидностей состязательных атак:
1. Атака на изображение или видео;
2. Атаки на промпт, то есть на клики мышью в виде точек и на bounding box-ы.

Атака на изображение или видео
Мы хотим понять, насколько текущая модель SAMv2 устойчива к классическим состязательным атакам, таким как Projected Gradient Descent (PGD). Идея атаки состоит в добавлении шума к картинке путем минимизации DiceLoss между маской, которая выдает модель, и маской, которую мы хотим получить. Наш шум является обучаемым параметром, все веса модели заморожены и не могут быть изменены.

Первая стадия – это атака на одно изображение и один промпт (в нашем случае один клик). Для осуществления атаки используем классический PGD, т.е. просто оптимизируем шум по DiceLoss между двумя масками: маской, которую выдает модель, и маской, которую мы хотим видеть от модели. Шум сходится к локальному минимуму быстро.

Вторая стадия – это атака на одно изображение и множество промптов (кликов). Теперь мы хотим сделать шум который был бы инвариантен ко всем или почти всем возможным промптам. Для этого мы делаем следующее: ставим точку, считаем DiceLoss и градиент и обновляем шум, далее ставим точку в другое место и опять считаем DiceLoss, градиент и обновляем шум. Данный алгоритм сходится к оптимальному решению, т.е. получается найти шум, который инвариантен к множеству точек.

Третья стадия – это атака на видео, т.е. на несколько изображений сразу. Сначала мы наложили шум созданный из его оптимизации с первым видео фреймом, далее этот шум накладывается на все остальные фреймы. Описанный алгоритм работает не самым лучшим образом, так что хочется сделать шум инвариантным к множеству изображений. Для этого в будущей работе нужно поэтапно оптимизировать шум на разных фреймов из видео. Таким образом, мы будет делать состязательную атаку на Memory Encoder.

Атака на клики мышью в виде точек

Интересная с практической точки зрения часть, поскольку пользователи сами неявно совершают такие адверсативные взаимодействия при использовании модели. В результате этой атаки мы хотим получить такую сегментационную маску, которая будет идеальной, что возможно путем максимизации IoU между предсказанной и референсной (gt) маской или разрушенной, что достигается при минимизации IoU между предсказанной и gt маской. Для этого мы заморозили всю модель SAM2, но пробросили градиенты через координаты точек промпта. Так, точки в промпте будут изменяться по мере итераций. В случае максимизации IoU, мы минимизируем Dice Loss между предсказанной и референсной маской, в обратном случае – верно обратное. Однако, при такой постановке задачи точка выходит за границы изначальной сегментационной маски, чего мы хотим избежать, ведь такая ситуация нереалистична. Поэтому необходимо применить регуляризацию к лоссу, для этого мы строим поле для нашего изображения, где значения поля в каждом пикселе является расстоянием между данным пикселем и ближайшем пикселем сегментационной маски (distance transform). После чего, мы применяем метод дифференциальной растеризации для преобразования нашей точки в окружность фиксированного радиуса (гиперпараметр), названным далее промптом. Мы суммируем поле в значениях, на которых накладывается наш промпт. Данная сумма является регуляризацией. Она не позволяет нашему пройти выйти за пределы региона интереса.

Атака на bounding box

Атака на bounding box заключается в поиске таких координат, которые нарушают маску, первоначально заданную пользователем. Такая ситуация может возникнуть случайно, если несколько масок перекрываются, и, как следствие, истинные bounding box-ы соседних масок располагаются слишком близко друг к другу.
Для поиска «неправильных» bounding box-ов использовалась функция потерь DiceLoss и метрика IoU (пересечения предсказанной маски и истинной маски). В работе рассматриваются два типа атаки: первый — минимизация метрики IoU, а второй — её максимизация. Чтобы избежать схлопывания рамки в одну точку, была введена регуляризация площади bounding box в виде L2 регуляризации.
В первом случае bounding box перемещался по маске, что могло привести к потере части деталей. Во втором случае целью было продемонстрировать, что даже небольшое увеличение bounding box может значительно изменить маску, добавив к исходной маске лишние детали.

Future work

1. Перенос полученных экспериментов с изображений на видео;
2. Экспериментирование с текущей функцией потерь (Dice Loss). Добавить Cross Entropy, Focal Loss. Также возможно добавление лосса на определенные пиксели;
3. Добавление грамотного подбора гиперпараметров.

Summary

Таким образом, мы рассмотрели различные атаки на новую модель SAMv2, демонстрируя ее недостатки и чувствительность к атакам. Это означает, что данную модель нужно улучшить, чтобы не допустить возможных атак. В данном проекте мы рассмотрели адверсативные атаки к изображениям, однако необходимо провести эксперименты с видео. Все упомянутое ранее дает возможность для дальнейших исследований.