Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

«Методы машинного обучения»

Домашнее задание

ИСПОЛНИТЕЛЬ:

Студент группы ИУ5-23М

Бакланов Н.В. _____

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е. _____

Москва 2022

ВЫБОР ЗАДАЧИ

Был проанализирован ресурс “paperswithcode”, который включает описание нескольких тысяч современных задач в области машинного обучения. Для анализа в рамках домашнего задания были выбраны две актуальные работы в области автономного вождения и обнаружения объектов: “TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving” и “ MotionCNN: A Strong Baseline for Motion Prediction in Autonomous Driving”. Далее рассмотрим цели, ключевые подходы и результаты, представленные в выбранных статьях.

1. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving

Введение

Авторы статьи рассматривают методы интегрирования представлений от различных датчиков для автономного вождения. Слияние на основе геометрии показало себя многообещающим для восприятия (например, обнаружение объектов, прогнозирование движения). Однако в контексте непрерывного вождения обучение, основанное на существующих методах объединения датчиков, неэффективно в сложных сценариях вождения с высокой плотностью динамических агентов.

Поэтому авторы предлагают TransFuser, механизм для интеграции изображений и представлений LiDAR с использованием внутреннего внимания. Были проведены эксперименты и проверена его эффективность на новом сложном эталоне с длинными маршрутами и интенсивным движением, а также на симуляторе вождения CARLA.

На момент публикации TransFuser значительно превосходит работы, протестированные на CARLA с точки зрения набранных очков. По сравнению

со слиянием на основе геометрии TransFuser снижает среднее количество столкновений на километр на 48%.

Рассмотрим сценарий (Рис.1), в котором самоуправляемое транспортное средство (показанное зеленым цветом) вот-вот въедет на перекресток. Чтобы безопасно передвигаться по перекрестку, эго-автомобиль должен понимать взаимосвязь между светофором справа (показаны желтым цветом) и транспортными средствами слева (показаны красным). Если сигнал светофора зеленый, то автомобили слева будут двигаться к правой стороне перекрестка. Если транспортное средство эго не может уловить эту связь, это может привести к опасным столкновениям. Следовательно, в плотной городской среде с множеством транспортных средств и светофоров крайне важно, чтобы любое транспортное средство моделировало зависимости между различными объектами во всей 3D-сцене, т. е. эго-транспортное средство должно фиксировать глобальный контекст 3D-сцены.

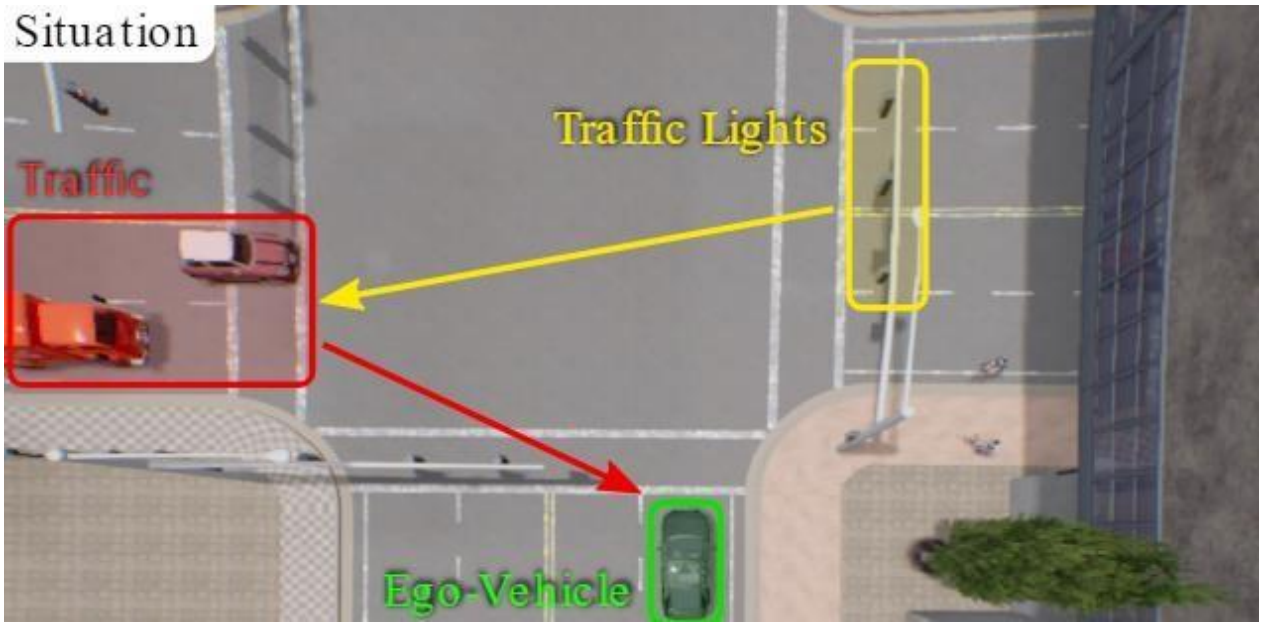

Рис. 1

Автомобиль воспринимает окружающую среду с помощью различных датчиков, наиболее популярными из которых являются камеры и LiDAR. Несмотря на то, что камера обеспечивает плотную перцептивнуюинформацию о сцене, ей не хватает надежной трехмерной информации, и она

очень чувствительна к изменениям погодных условий. С другой стороны, LiDAR состоит из трехмерной информации, но измерения LiDAR обычно очень редки (особенно на расстоянии) и не содержат важной информации, такой как состояние светофора. Следовательно, методы только для изображений и только для LiDAR, вероятно, не сработают в сложных городских сценариях.

Это ограничение можно уменьшить, объединив информацию с камеры и датчиков LiDAR, поскольку их преимущества дополняют друг друга. Это приводит к нескольким исследовательским вопросам:

1. Как интегрировать информацию с нескольких датчиков?
2. В какой степени они должны обрабатываться независимо?
3. Какой механизм слияния использовать для максимального прироста?

В основном, работы по объединению датчиков были сосредоточены на слиянии на основе геометрии между камерой и датчиками LiDAR. В этом методе точки в трехмерном пространстве (облако точек LiDAR) проецируются в пиксели в пространстве изображения (ввод камеры), а информация из проецируемых местоположений агрегируется. В частности, функции (извлеченные с помощью сверточной нейронной сети), соответствующие этим проецируемым местоположениям, объединяются вместе. На приведенном выше рисунке это называется проекцией геометрического элемента. Было показано, что это очень эффективно для задач машинного зрения, таких как обнаружение объектов, прогнозирование движения и оценка глубины, но не было широко изучено в контексте сквозного вождения.

Авторы отмечают, что геометрическое слияние неэффективно в сложных городских сценариях с интенсивным движением, что приводит к столкновению с другими агентами в сцене.

Они предполагают, что это происходит из-за отсутствия глобального контекста, поскольку объекты агрегируются из локальной области в спроецированном 2D- или 3D-пространстве. На иллюстрации, показанной ниже (Рис.2), для области светофора на изображении (показанной желтым)

геометрическое слияние объединяет признаки из синей области в облаке точек LiDAR, поскольку эти точки проецируются на желтую область в пространстве изображения. Тем не менее, чтобы безопасно перемещаться по перекрестку, важно агрегировать признаки из красной области в облаке точек LiDAR, поскольку он перекрывается с транспортными средствами, движущимися слева направо.

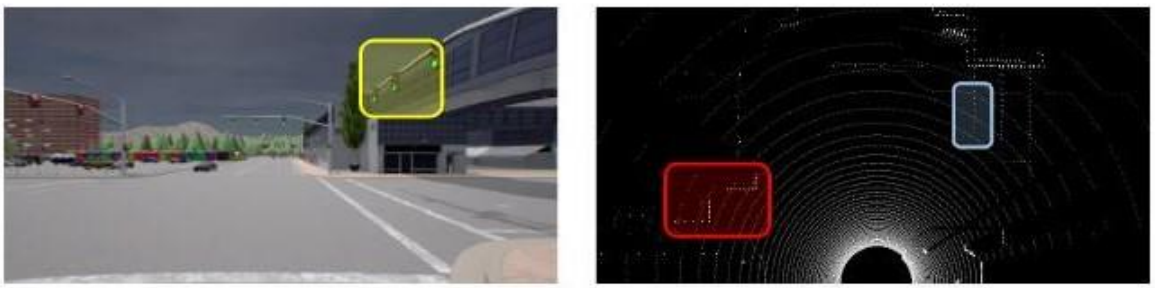

Рис.2

Методы слияния датчиков

Большинство работ по слиянию сенсоров рассматривают задачи обнаружения объектов и прогнозирования движения. Они работают на многоракурсном LiDAR, т.е. Bird’s Eye View (BEV) и RangeView (RV) или дополняют изображение с камеры информацией о глубине от LiDAR. Обычно это достигается путем проецирования функций LiDAR в пространство изображения или проецированием признаков изображения в пространство BEV или RV. Наиболее близким подходом к рассматриваемому в данной работе является ContFuse, который выполняет многомасштабное плотное слияние между изображением и функциями LiDAR BEV. Для каждого пикселя в представлении LiDAR BEV он вычисляет ближайших соседей в локальной окрестности в трехмерном пространстве, проецирует эти соседние точки в пространство изображения, чтобы получить соответствующие признаки изображения, объединяет эти признаки с помощью непрерывных сверток и объединяет их с признаками LiDAR BEV. Работы EPNET++ и CAT-Det также используют мультимасштабное двунаправленное объединение изображений и

облаков точек LiDAR, используя механизмы внимания для получения расширенных представлений признаков для обнаружения 3D-объектов. Другие методы слияния на основе проекций следуют аналогичной тенденции и собирают информацию из локального окружения в 2D или 3D пространстве. Однако представление состояния, полученное с помощью этих методов, недостаточно, поскольку они не охватывают глобальный контекст трехмерной сцены, что важно для безопасного маневрирования в условиях плотного дорожного движения.

Предложенный новый подход

Последние мультимодальные методы сквозного вождения показали, что дополнение RGB-изображений глубиной и семантикой может повысить эффективность вождения. В данной работе исследователи сосредоточились на входных данных изображения и лидара, поскольку они дополняют друг друга с точки зрения представления сцены и легкодоступны в автономных системах вождения.

Ключевая идея состоит в том, чтобы использовать слияние признаков на основе внимания, которое помогает сети объединять признаки из соответствующих областей в обоих датчиках. Эти соответствующие области распределены по всему входному пространству, а не ограничены только проецируемыми местоположениями (как при геометрическом слиянии). Это помогает зафиксировать глобальный контекст всей 3D-сцены. В частности, для этой цели авторы использовали Трансформеры — Multi-Modal Fusion Transformer (TransFuser).

Авторы предложили новую архитектуру сквозного вождения (Рис. 3). Она состоит из двух основных компонентов: (1) мультимодальный трансформер слияния для интеграции информации с нескольких модальностей датчиков (изображение и LiDAR) и (2) авторегрессионная сеть прогнозирования путевых точек.

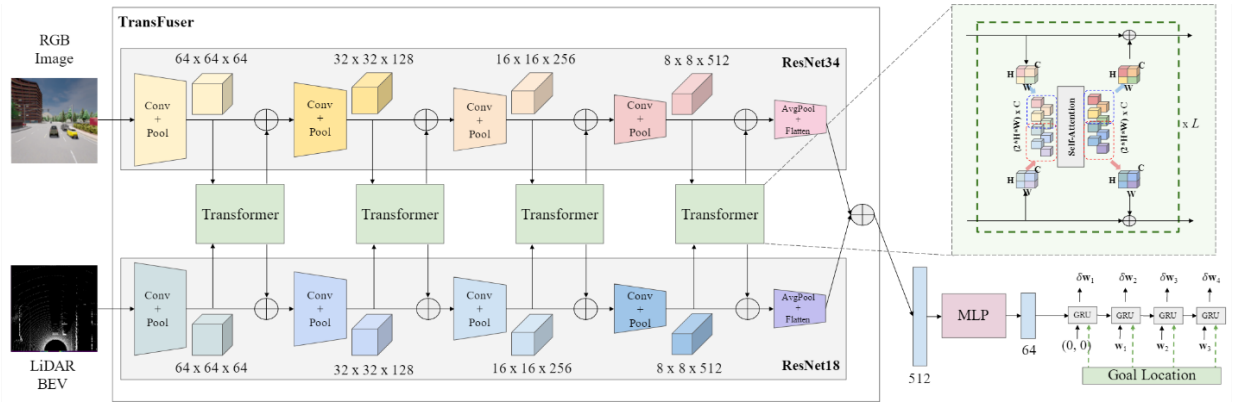

Рис. 3

В качестве входных данных для модели рассматривается RGB-изображение и LiDAR BEV (вид сверху на облако точек LiDAR). Затем они обрабатываются свёрточными нейронными сетями (в частности, модулями ResNet), в результате чего получаются промежуточные карты объектов разного разрешения. Затем мы используем преобразователи для объединения изображения и функций LiDAR с разными разрешениями. TransFuser выводит вектор признаков, который затем передается в сеть авторегрессионного прогнозирования путевых точек на основе GRU. Прогнозируемые путевые точки затем передаются на ПИД-контроллеры, которые выводят управление транспортным средством.

Ключевая идея состоит в том, чтобы использовать механизм внимания трансформеров для включения глобального контекста для модальностей изображения и LiDAR, учитывая их взаимодополняющий характер. Архитектура трансформера принимает в качестве входных данных последовательность, состоящую из дискретных токенов, каждый из которых представлен вектором признаков. Вектор признаков дополняется позиционным кодированием для включения пространственных индуктивных смещений.

Трансформер многократно применяет механизм внимания по всей архитектуре, что приводит к нескольким уровням внимания. Промежуточные карты признаков каждой модальности рассматриваются как набор, а не как пространственную сетку, и рассматриваем каждый элемент набора как токен.

Экстракторы сверточных признаков для изображений и входных данных LiDAR BEV кодируют различные аспекты сцены на разных уровнях. Поэтому эти признаки объединяются в нескольких масштабах.

После выполнения плотного слияния признаков с несколькими разрешениями мы получаем карту признаков размерами 22 × 5 × C из ветви изображения и 8 × 8 × C из ветви BEV. Где C — количество каналов при текущем разрешении в экстракторе признаков. Эти карты признаков уменьшены до размерности 512 за счет слоя усредняющего пулинга, за которым следует полносвязный слой. Затем вектор признаков размерности 512 как из изображения, так и из потоков LiDAR BEV объединяется посредством поэлементного суммирования. Этот 512-мерный вектор признаков представляет собой компактное представление среды, которое кодирует глобальный контекст 3D-сцены. Затем это вектор уже передается в сеть прогнозирования путевых точек.

Результаты

Были проведены эксперименты с использованием CARLA в условиях плотной городской застройки, включая сложные сценарии, такие как пешеходы, выходящие из закрытой области и пересекающие дорогу в случайных местах, другие транспортные средства, проезжающие на красный свет на перекрестках и незащищенные повороты. Результаты показаны на рисунке 4.

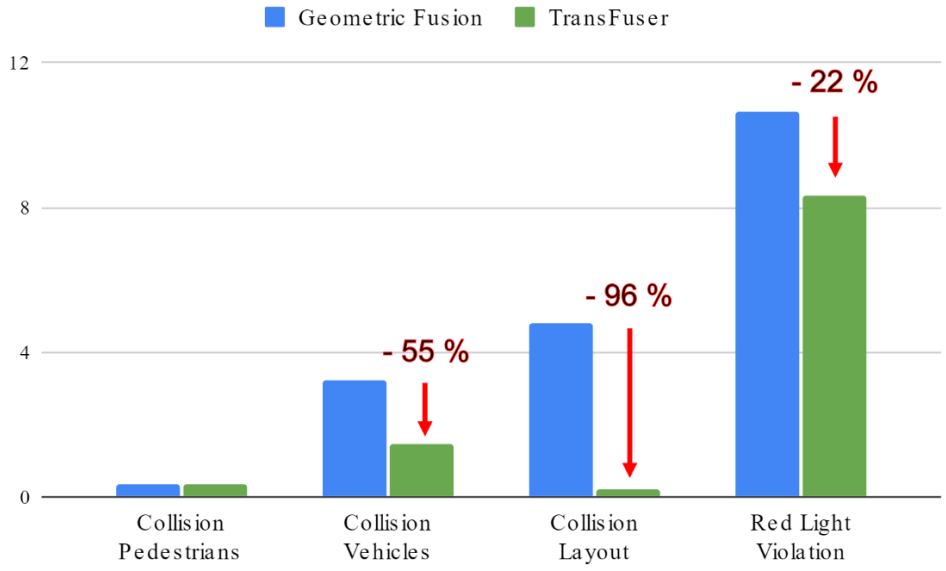

Рис. 4

Результаты различных моделей представлены на рисунке 5. Среди моделей, не использующих вспомогательный контроль, TransFuse достигает наивысшего DS, опережая Geometric Fusion на 17%. CIL-WP имеет архитектуру, аналогичную AIM, и использует 2D-семантику и глубину в качестве вспомогательного контроля. NEAT использует неявную функцию для сопоставления местоположений в пространстве BEV с функциями входного изображения, используя внимание вместе с семантическим прогнозированием BEV в качестве вспомогательной задачи. Включение вспомогательного надзора, как в эти подходы, показало успех в недавней работе по имитационному обучению и, вероятно, также улучшит производительность TransFuser.

Method	Auxiliary Supervision	Sensors	DS	RC	IM
CILRS [5]	Velocity	1 camera	5.37	14.40	0.55
LBC [2]	BEV Semantics	3 cameras	8.94	17.54	0.73
AIM	None	1 camera	12.88	41.52	0.59
Late Fusion	None	1 camera + LiDAR	13.27	42.10	0.54
Geometric Fusion	None	1 camera + LiDAR	14.47	40.99	0.48
TransFuser (Ours)	None	1 camera + LiDAR	16.93	51.82	0.42
CIL-WP	2D Semantics + Depth	1 camera	19.38	67.02	0.39
NEAT	BEV Semantics	3 cameras	21.83	41.71	0.65
MaRLn [18]	2D Semantics + Affordances	1 camera	24.98	46.97	0.52

Рис. 5

Верхняя запись таблицы лидеров, MaRLn, основана на представленном методе обучения с подкреплением (RL). В этом подходе кодировщик сначала обучается прогнозировать как двумерную семантику, так и конкретные возможности, такие как состояние светофора, а также относительное положение и ориентацию между транспортным средством и полосой движения. Затем кодировщик замораживается и используется для обучения метода RL на основе функции значения. Авторы MaRLn указали, что для обучения модели одного города требуется 20 дней моделирования. Для сравнения, представленный в данной работе обучающий набор данных для всех городов CARLA требует всего 21 часа моделирования и имеет потенциал для дальнейшего улучшения, например, с помощью ортогональных методов, таких как активное обучение и DAgger.

2. MotionCNN: A Strong Baseline for Motion Prediction in Autonomous Driving

Введение

Авторы статьи считают, что одним из ключевых компонентов самоуправляемой системы является прогнозирование движения. Для автономного транспортного средства крайне важно надежно прогнозировать будущие траектории других участников дорожного движения, таких как автомобили, велосипедисты и пешеходы. Однако прогнозирование будущего движения и планирование маршрута автономного транспортного по-прежнему является очень сложной задачей, и ее еще предстоит решить. В этой статье автор решает задачу прогнозирования движения. Наиболее известные подходы включают модели на основе изображений, которые используют растровые изображения сцен с высоты птичьего полета и методы, созданные с использованием графовых нейронных сетей.

Автор же предлагает создать простой и в то же время эффективный базовый метод прогнозирования движения, основанный исключительно на сверточных нейронных сетях (CNNs). Данная модель принимает в качестве входных данных растровое изображение, центрированное вокруг целевого агента, и предсказывает набор возможных траекторий вместе с их возможными состояниями. Растровое изображение получается путем растеризации сцены и истории всех агентов.

Метод

Предполагается, что треки объектов сохраняются и обрабатываются некоторой системой и наша задача состоит только в прогнозировании движения. Необходимо предсказать траекторию объекта в следующие T секунд. В этом разделе сначала опишем метод, с помощью которого

растрируем данные и создадим многоканальные изображения. После этого мы опишем архитектуру нашей модели и функцию, используемую для обучения.

Модель

Будущее неоднозначно, поэтому мы стремимся создать K различных гипотез (предложений) для будущей траектории, которые будут оцениваться с истинной траекторией. Мы воплощаем нашу модель в виде регрессии на основе изображений. Модель состоит из CNN, предварительно обученной на ImageNet и одного полносвязного слоя (Рис. 6). Модель принимает многоканальное растровое изображение в качестве входных данных и предсказывает K траекторий вместе с соответствующими значениями уверенности c1, … , c_K, которые нормализуются с помощью оператора softmax таким образом, что $\sum_k Ck = 1$

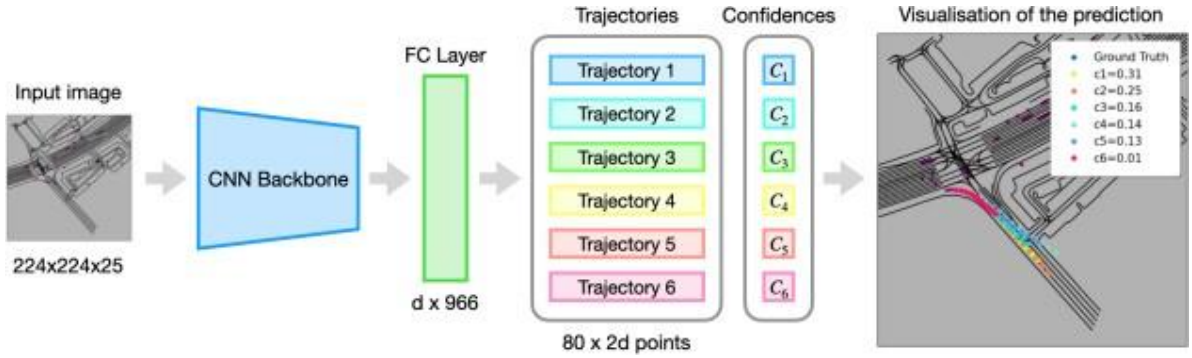

Рис. 6

Функция потерь

Простым решением было бы использовать потери среднеквадратичной ошибки (MSE). Однако эта потеря не позволяет проводить вероятностное моделирование нескольких гипотез и показала плохую производительность в предварительных экспериментах. Вместо этого авторы предложили моделировать возможные будущие траектории как смесь K распределений

Гаусса. В этом случае сеть выводит средние значения гауссианов, в то время как они фиксируют ковариацию каждого гауссиана в смеси равной единичной матрице I. Затем для потери можно использовать отрицательное логарифмическое правдоподобие (NLL) этой смеси гауссианов, определяемое предсказанными предложения с учетом истинных координат. Другими словами, если задана траектория истины

$$X^{gt} = [(x_1, y_1), \ldots, (x_T, y_T)]$$

и K гипотез предсказанной траектории

$$X_k = [(x_{k,1}, y_{k,1}), \ldots, (x_{k,T}, y_{k,T})],\ k = 1, \ldots, K,$$

мы вычисляем отрицательную логарифмическую вероятность траектории наземной истины при предсказанной смеси гауссианов со средними значениями, равными предсказанным траекториям, и матрицей идентичности I как ковариацию:

$$L = -logP(X^{gt}) = -log \sum_k c_k \mathcal{N}(X^{gt}; \mu = X_k, \Sigma = I)$$

где N (.; μ, Σ) — функция плотности вероятности для многомерного гауссовского распределения со средним μ и ковариационной матрицей Σ. Потеря может быть дополнительно разложена на произведение одномерных гауссианов, и мы получим просто логарифм суммы показателей:

$$L = -log \sum_k c_k \prod_{t=1}^{T} \mathcal{N}(x_t^{gt}; x_{k,t}, 1)\mathcal{N}(y_t^{gt}; y_{k,t}, 1),$$
$$= -log \sum_k e^{log(c_k) - \frac{1}{2}\sum_{t=1}^{T}(x_t^{gt} - x_{k,t})^2 + (y_t^{gt} - yk,t)^2}$$

Предлагаемая функция потерь явно не наказывает модель за создание очень близких траекторий. Однако эмпирически мы не наблюдали коллапса моды,

потому что объединение всей вероятностной массы в одну моду приводит к более высокой стратегии риска и более высоким значениям потерь в случае ошибочного прогноза. Таким образом, оптимизация предлагаемых потерь дает достаточную мультимодальность.

Вывод

Мы выбираем количество компонентов в смеси K, равное желаемому количеству прогнозируемых гипотез. Например, при оценке на Waymo Open MotionDataset разрешено предоставлять до 6 гипотез будущей траектории для целевого агента, поэтому мы выберите K = 6. Поскольку мы моделируем возможное пространство решений, используя распределение вероятностей, полезно получить максимально разнообразный набор гипотез из нашего распределения. Один из способов добиться этого состоит в том, чтобы просто выбрать средние значения компонентов, составляющих предсказанную смесь гауссиан, вместе с коэффициентами ck как их достоверности в качестве окончательных гипотез для оценки. Авторы говорят о том, что это может быть неоптимальным решением, но оставляют исследование других способов выборки траекторий из предсказанного распределения для будущей работы.

Датасет

Подход оценивался на датасете Waymo Open MotionDataset. Этот набор данных содержит траектории объектов и соответствующие 3D-карты для 103354 сегментов. Каждый сегмент представляет собой 20-секундную запись траектории объекта с частотой 10 Гц и картографические данные для области, охватываемой этим сегментом. Одна выборка включает 1 секунду истории и 8 секунд будущих данных, полученных путем разбиения сегментов на 9-секундные окна с 5-секундным перекрытием. Каждая такая выборка содержит

до 8 агентов, помеченных как «действительные», для которых модель должна предсказать их положение на 8 секунд в будущем.

Метрики

Предсказывается 6 гипотез для каждого целевого агента, но толькоточки траектории, субдискретизированные с частотой 2 Гц (что приводит к подмножеству 16 двумерных координат из предсказанных 80 точек), используются для вычисления тестовых и проверочных метрик. Средняя ошибка смещения, окончательная ошибка смещения (FDE) — это наиболее часто используемые показатели для оценки:

$$\text{ADE} = \frac{1}{T}\|X^{gt} - X\|_2, \ \text{FDE} = \|x_T^{gt} - x_T\|_2,$$

где

X^{gt} — траектория истинности, а X — предсказанная траектория.

Для оценки нескольких гипотез мы используем minADE и minFDE:

$$\text{minADE} = \min_k \frac{1}{T}\|X^{gt} - X_k\|_2,$$
$$\text{minFDE} = \min_k \|x_T^{gt} - x_{k,T}\|_2.$$

Результаты

Результаты финальной таблицы лидеров испытания прогнозирования движения с открытым набором данных Waymo представлены на рисунке 7. Несмотря на простоту предложенного подхода, работа заняла 3-е место по метрике mAP. Более того, модель превосходит другие конкурирующие методы по показателям Min ADE, Min FDE и Overlap Rate. Простая модель достигает таких впечатляющих результатов без использования передовых методов глубокого обучения или сложных архитектур. На проверочном наборе эта архитектура в 3 раза быстрее обучается, чем архитектура с Xception71backbone, но не достигает такой же высокой производительности,

что свидетельствует о том, что для достижения хороших результатов необходима достаточно глубокая модель.

	Method	mAP	Min ADE	Min FDE	Miss Rate	Overlap Rate
Test	Waymo LSTM baseline [1]	0.1756	1.0065	2.3553	0.3750	0.1898
	ReCoAt (2^{nd} place) [12]	0.2711	0.7703	1.6668	0.2437	0.1642
	DenseTNT (1^{st} place) [9]	**0.3281**	1.0387	1.5514	**0.1573**	0.1779
	MotionCNN-Xception71 (Ours)	0.2136	**0.7400**	**1.4936**	0.2091	**0.1560**
Val	MotionCNN-ResNet18 (Ours)	0.1920	0.8154	1.6396	0.2552	0.1605
	MotionCNN-Xception71 (Ours)	0.2123	0.7383	1.4957	0.2072	0.1576

Рис. 7

Авторы представили простую, но сильную базовую линию — MotionCNN, которая основана на CNN и создает распределение гипотетических траекторий для целевого агента. Предложенную модель легко внедрить и легко обучить. В ней используется растровое представление сцены с высоты птичьего полета, которое кэшируется как многоканальные изображения для более быстрого обучения.

ЗАКЛЮЧЕНИЕ

Был проведен обзор работы, представляющий новую архитектуру интеграции представлений различных модальностей. TransFuser использует внимание для захвата глобального контекста 3D-сцены и сосредотачивается на динамических агентах и светофорах, что приводит к современной производительности CARLA со значительным сокращением нарушений. Учитывая, что метод прост, гибок и универсален, было бы интересно исследовать его дальше с дополнительными датчиками.

Также мы рассмотрели статью, представляющую базовый метод прогнозирования движения, основанный исключительно на сверточных нейронных сетях (CNNs). Модель простая, но показывает достаточно хорошие результаты и может служить отправной точкой для будущих исследований в области прогнозирования движения.

СПИСОК ИСТОЧНИКОВ

1. [Электронный ресурс] - https://paperswithcode.com/sota
2. Chitta K. et al. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving //arXiv preprint arXiv:2205.15997. – 2022.
3. [Электронный ресурс] - https://github.com/autonomousvision/transfuser
4. Aditya P. et al Supplementary Material for Multi-Modal Fusion Transformer for End-to-End Autonomous Driving
5. Konev S., Brodt K., Sanakoyeu A. Motioncnn: A strong baseline for motion prediction in autonomous driving //Workshop on Autonomous Driving, CVPR. – 2021.
6. [Электронный ресурс] - https://github.com/kbrodt/waymo-motion-prediction-2021