УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Момир Аџемовић

ПРЕДИКЦИЈА ТРАЈЕКТОРИЈА ВИШЕ ОБЈЕКАТА НА СЦЕНИ

мастер рад

Београд, 2022.

Ментор:

др Младен Николић, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Јована Ковачевић, доцент
Универзитет у Београду, Математички факултет

др Александар Картељ, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 15. септембар 2022.

Посвета... у изради...

Наслов мастер рада: Предикција трајекторија више објеката на сцени

Резиме: У изради...

Кључне речи: машинско учење, аутономна вожња, растеризација, графовске неуронске мреже

Садржај

Глава 1

Увод

TODO: ...

1.1 Поставка проблема

TODO: ...

1.2 Опис података

TODO: HD мапе...

Глава 2

Преглед претходних приступа

TODO: Средити одређене делове да текст буде јаснији...

Техника за предикцију трајекторија више објеката које издвајају посебан објекат као агент, могу да се групишу грубо у четири групе. Постоје и општије технике које не разликују конкретно агента од осталих објеката у процесу предвиђања.

2.1 Опште технике које не издвајају одређени објекат као агента

Неке од првих метода за предикцију трајекторија, јесу рекурентне неуронске мреже (*eng. RNN - Recurrent neural network*) и конволутивне мреже за једну димензију (*CNN - Convolutional Neural Network*). Често је коришћена *LSTM* архитектура рекурентних неуронских мрежа погодна за коридање динамике објеката. Како трајекторија једног објекта зависи од трајекторија осталих објеката, неопходно је да се користи некакав механизам пажње којим се добија веза између различитих објеката на сцени. [1] [2] Мана овог скупа модела је што модели нису погодни за потпуно разумевање веза између различитих елемената сцена (објекти, возне линије, саобраћајни знакови, ...).

Уместо учења модела који дају предикције трајекторија које се у просеку најбоље у тим случајевима, алтернатива су скуп архитектура заснованих на (условљеним) генеративним моделима који омогућавају генерисање произвољог броја трајекторија, као узорковање из условљене расподеле (расподела је условљена историјом трајекторије, као и окружењем у којем се тај објекат налази). Пример је *Social GAN* [11] архитектура која узима у обзир претходне наведене услове и генерише „социјално прихватљиве“ трајекторије пешака. Један од главним мотива ових мрежа је одговор на проблем мултимодалности расподела трајекторија. У случају возила са агентом, генеративни модели захтевају током процеса предикције напредније алгоритме

узорковања ради оптимизације покривености трајекторијама[1].

2.2 Технике засноване на растеризацији

Растеризација подразумева трансформацију *HD* мапа у формат слике. Слике приказују сцене из птичје перспективе (*eng. BEV - Bird's-eye-view*). Предност овог формата је могућност примене више слојева конволутивних неуронских мрежа за извлачење контекста тих мапа. Овакве компоненте архитектуре се углавном називају енкодери. Конволутивне мреже нису ограничене да раде са RGB и сличним форматима слика тј. сваком пикселу може да се додели низ својстава. Нека од очигледнијих својстава су: да ли агент заузима тај пиксел, да ли сусед (неагент) заузима тај пиксел, да ли пиксел припада улици, ... Излаз CNN енкодера се углавном накнадно комбинује са осталим компонентама за генерисање резултата.

Излази модела такође могу да се представљају као слике тј. топлотне мапе *(eng. Heat Map)*, где се сваком пикселу додељује вероватноћа да се агент (или посматрани објекат) налази на тој локацији. Топлотне мапе се добијају декодерима који комплементирају енкодер компоненте. То значи да није неопходно да се експлицитно постави ограничење на одређени број излазних трајекторија модела. Проблем код више излазних трајекторија модела је то што може да изазове колапс моде (eng. *mode collapse*). Број трајекторија се овде не дефинише експлицитно и може да варира од једне сцене до друге. Наравно, не узимају се сви пиксели као кандидати, већ се примењује неки алгоритам узорковања. [3] [4]

Архитектура *HOME* [3] користи овај принцим за паралелно кодирање растеризоване сцене и трајекторија објеката. Резултати компоненти се спајају и прослеђују као улаз у декодер за генерисање топлотне мапе. Узорковањем тачака из топлотних мапа се добија скуп кандидата тачака. Главна претпоставка архитектура заснованих на топлотним мапама је: Ако знамо циљну тачку и историју трајекторије објекта, онда можемо једноставно да одредимо трајекторију до те циљне тачке. За одређивање ових трајекторија могу да се користе једноставнији модели неуронских мрежа.

Могуће је растеризовати потпуно податке тј. растеризовати и трајекторије као низ слика. Свака слика садржи скуп тачака које представљају локације објеката. Архитектура *CASPNet* [5] примењује CNN енкодер на свако стање и користи ConvLSTM [6] за разумевање темпоралних веза. На овај начин се извлаче својства динамичког дела сцена. На сличан начин је могуће и извући својства и за статички део (возне линије, возни површине, ...) користећи класичне конволутивне мреже. Комбинацијом ових података се генеришу трајекторије које су исто у растеризованом облику.

[1]Покривеност се односи на метрике које узимају у обзир више од једне трајекторије, па је битно и да те саме трајекторије буду разноврсне

2.3 Технике засноване на графовским репрезентацијама

Мапе имају комплексну топологију. Технике засноване на растеризацији користе конволуцију која тешко извлачи потпуно семантику тих мапа. Алтернатива је моделовање мапа графовским структурама. Технике засноване на графовском репрезентацијом као улаз добијају стање мапе кодиране као граф и примењују моделе графовских неуронских мрежа. Две архитектуре које представљају основе за већину архитектура ове групе су *LaneGCN* [7] и *VectorNet* [8].

Мапа може да се моделује као скуп повезаних сложених линија (*end. polylines*), где сваком објекту одговара једна усмерена сложена линија. *VectorNet* је хијерархијска графовска неуронска мрежа која као улаз добија мапу која је моделована као скуп сложених линија, а као резултат даје скуп предикција трајекторија. Идеја је да се прво извуку својста из сложених линија појединачних објеката, а онда то пронађу одговарајуће везе између између објеката међусобно и између објеката и возних линија. [8]

Архитектура *LaneGCN* нуди варијанту конволутивних графовских мрежа (*GCN - Graph Convolution Network*) која је специјализована за графове возних линија које имају различите типове веза. Користе се различите матрице повезаности за суседе, претходнике, следбенике (леви и десни). За сваку матрицу повезаности може да се примени класичан GCN, а комбиновањем тих елемената је добија један LaneGCN слој. [7]

2.4 Хибридне технике

Хибридне технике користе комбинацију структура графова и BEV слика. *GOHOME* Модификована верзија *HOME* која уместо CNN енкодера и растеризованих слика мапа, користи *LaneGCN* архитектуру за енкодер. Заправо ансамбл ова два модела (*HOME, GOOME*) даје најбоље резултате по *MR* метрици[2].

2.5 Технике засноване на облацима тачака

Последња група нешто одступа од осталих али и даље даје добре резултате. Подаци се посматрају као облаци тачака (*eng. point cloud*) и примењују се технике намењене за такву структуру података. Основна архитектура је *TPCN* [9] која је заснована на *PointNet* [10], а већина осталих техника су „изведене“.

[2]Објашњење за ову метрике се налази у секцији за евалуацију

Глава 3

Припрема података

Основни скуп података за тренирање и тестирање техника предикције трајекторија је *Argoverse Motion Forecasting* скуп података који се састоји од 324 хиљаде детаљних мапа саобраћаја (*eng. „HD maps“*) које садрже геометријске и семантичке податке сцена. Постоје две *HD* сцене за градове Питсбург и у Мајами. Коришћењем аутономних возила су генерисани сценарији који представљају неколико узастопних слика сцена (у табеларном формату) на деловима мапа. Сви детаљи о овом скупу података се могу пронаћи на адреси www.argoverse.org [1].

Кључне информације које се издвајају из сваког сценарију су:

- Мапа сценарија (Питсбург или Мајами);
- Трајекторије агената;
- Трајекторије осталих објеката на сцени;
- Возне (централне) линије.

3.1 Претпроцесирање података

Подаци сваког сценарија се векторизују и чувају у полу-структуираном формату. За парсирање и обраду улазних података се користи *argoverse API* интерфејс.

Трајекторија агента[1] се дели на два дела: историја (својства) и реализација (будуће вредности). Реализација се састоји од N_r опажања x и y релативних координата[2] тј. облик реализације је $(N_r, 2)$. Историја се аналогно формира да садржи историју N_h опажања x і y релативних координата. Овај део трајекторије иде непосредно пре реализације. Посматрамо следеће случајеве:

[1]Низ (x, y) тачака, где је приближна временска разлика између две тачке око 0.1 секунде

[2]Све координате се нормализују тако да су релативне у односу на последње опажање у трајекторији историје агента

- Постоји више од $N_h + N_r$ опажања: Одбацује се реп трајекторије (првих неколико вредности хронолошки гледано);
- Постоји мање од $N_{hmin} + N_r$ опажања: Сценарио се одбацује (сматра се да је невалидан);
- Постоји између $N_{hmin} + N_r$ и $N_h + N_r$ опажања: реп трајекторије се допуњава до димензије $N_h + N_r$ посматрано као да објекат мирује у тим тренуцима.

Коначно, облик историје је $(N_h, 3)$, где трећа вредност означава да ли је опажање право (1) или допуњено (0).

Трајекторије суседних објеката се деле на два дела аналогно трајекторији агента. Неопходно је да се синхронизују трајекторије суседних објеката по временским ознакама (eng. *timestamp*) са трајекторијом агента, јер не постоји у сваком тренутку исти број објеката на сцени. Након синхронизације се трајекторије деле на историју и реализацију и проверава се да ли дужине тих делова задовољавају критеријуме:

- Уколико је дужина трајекторије историје краћа од N_{homin}, онда се објекат одбацује;
- Уколико је дужина трајекторије реализације краћа од N_{romin}, онда се објекат одбацује.

Као додатна провера, за сваки сусед се провера растојање од агента. Уколико је сусед превише далеко, онда се се он одбацује. Критеријум за одбацивање суседа узима у обзир брзину агента (по x и y оси одвоједно) и растојање њихових последњих опажања у трајекторији историје. Уколико неки од следећих услова није испуњен, сусед се игнорише у сценарију: $\frac{O_n^x}{A_s^x} \leq T_{steps}$, $\frac{O_n^y}{A_s^y} \leq T_{steps}$, где је O_n^x (O_n^y) нормализована x (y) координата суседа, $\overset{x}{s}$ ($\overset{y}{s}$) је наивно апроксимирана[3] брзина агента по x (y) оси и T_{steps} је параметар толеранције. Трајекторије се секу или допуњавају до фиксног облика. Векторизован облик: $(N_n, N_h, 3)$ и $(N_n, N_r, 3)$, где је N_n број судедних објеката.

На основу локације агента се издвајају сегменти централних линија (возне путање) које нису даље од агента за више од D_{lsinit}. Уколико нема пронађених сегмената централних линија, онда се вредност за D_{lsinit} множи K_{ls}[4] пута до највише D_{lsmax} (ако и даље нема сегмената, онда се сценарио одбацује). За сваки сегмент се чува низ од 10 (x, y) координата приширених са метаподацима:

- *is_ intersection* - да ли се сегмент сече са неким сегментом,
- *turn_ right* - да ли је у питању скретање у десно,

[3] Брзина се апроксимира као просек промена координата у трајекторији историје

[4] D_{lsinit} и K_{ls} су фиксне вредности у *argoverse* интерфејсу

Ознака параметра	Објашњење
N_r	Дужина трајекторије реализације (део који се предвиђа)
N_h	Дужина трајекторија историје
N_{hmin}	Минимална дужина трајекторије историје пре допуњавања
N_{hmin}	Минимална дужина трајекторије историје суседа пре допуњавања
N_{rmin}	Минимална дужина трајекторије реализације суседа пре допуњавања
T_{steps}	Умножак максималног растојања до сегмента централне линије
D_{lsmax}	Максимално растојање до централне линије

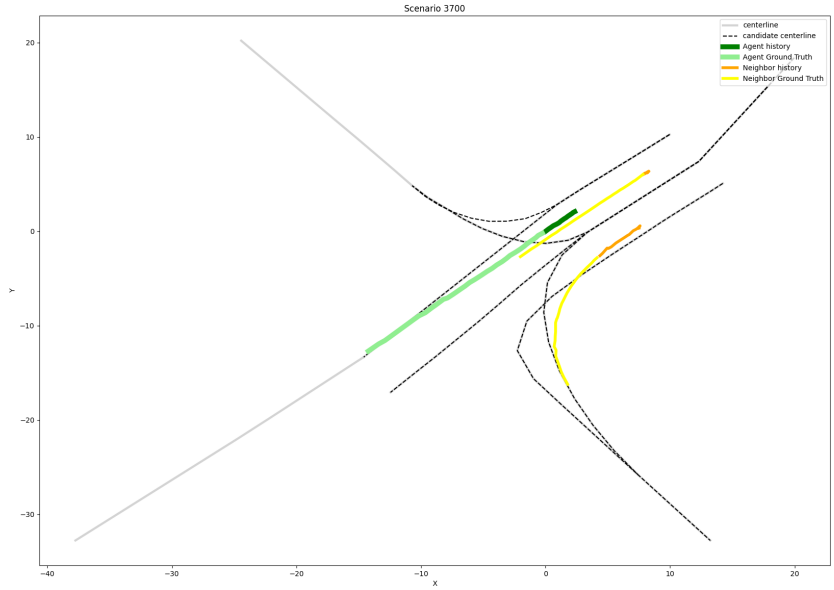
Табела 3.1: Преглед параметара припреме података

- *turn_ left* - да ли је у питању скретање у лево,
- *turn_ none* - да ли нема стретања,
- *is_ traffic_ control* - да ли постоји контрола саобраћаја.

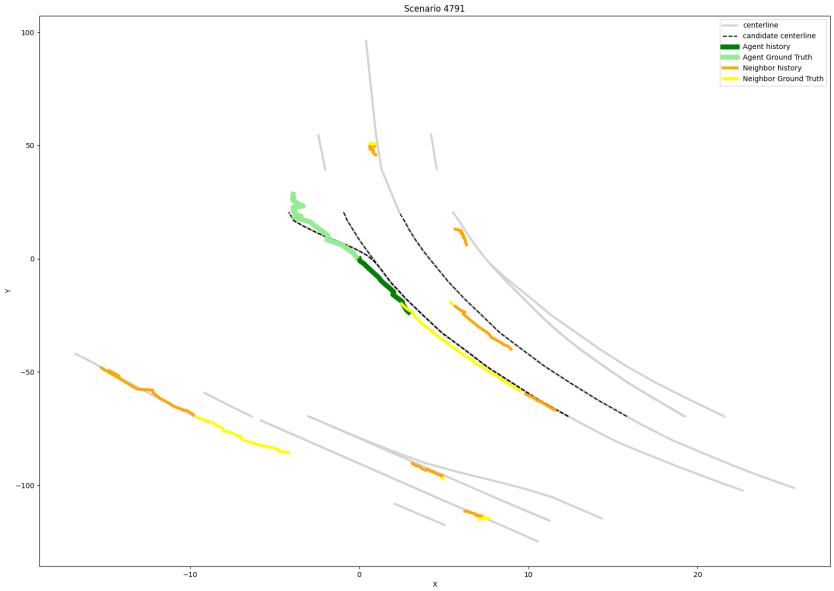
Коначан облик је $(N_{ls}, 10, 7)$.

Скуп кандидата централних сегмената линија за предикције трајекторија: Постоји коначан број централних сегмената линија по којој објекат може да се креће у скоријој будућности, због чега је корисно да се као улаз у модел користе централне линије као кандидати. Основа алгоритма за проналазак ових кандидата се налази у *argoverse* интерфејсу [1]. Кандидати се проналазе коришћењем трајекторије историје агента. Коначан векторизован облик је: $(N_c, N_r, 3)$, где је N_c број пронађених кандидата, N_r дужина трајекторије реализације. Пошто се централне линије допуњавају по потреби до димензије N_r, користи се трећа координата за маску. Погледати табелу 3.1 за преглед свих параметара процеса.

На сликама 3.1 и 3.2 се налазе примери два визуализована сценарија након претходне припреме. У овом формату нису прикази делови сцене на којој је могућа вожња, али постоје (централне линије) тј. путање по којима се возила најчешће крећу. Изузеци су у случају неких скретања, промени линија, ...



Слика 3.1: Визуализација припремљених података - Пример 1



Слика 3.2: Визуализација припремљених података - Пример 2

Глава 4

Преглед метода за евалуацију модела

Неке од стандардних метрика за евалуацију квалитета предикције трајекторија су „просечна грешка одступања“ (*eng. ADE - Average Displacement Error*) и „последња грешка одступања“ (*eng. FDE - Final Displacement Error*). У наставку се користе енглеске скраћенице *ADE* и *FDE*. Метрика *ADE* се добија упросечавањем еуклидског растојања између временски синхронизованих тачака трајекторија предикције и реализације. Метрика *FDE* узима у обзир само последњу тачку. [11] [1] У наставку се налазе формуле у случају да се посматра тачно један објекат (нпр. само агент):

$$ADE = \frac{1}{T} \sum_{k=1}^{T} \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2}$$

$$FDE = \sqrt{(x_{last} - \hat{x}_{last})^2 + (y_{last} - \hat{y}_{last})^2}$$

Метрике се једноставно уопштавају у случајевима где постоји више објеката на сцени:

$$ADE = \frac{1}{T \times N} \sum_{n=1}^{N} \sum_{k=1}^{T} \sqrt{(x_k^n - \hat{x}_k^n)^2 + (y_k^n - \hat{y}_k^n)^2}$$

$$FDE = \frac{1}{N} \sum_{n=1}^{N} \sqrt{(x_{last}^n - \hat{x}_{last}^n)^2 + (y_{last}^n - \hat{y}_{last}^n)^2}$$

Ове једноставне метрике су погодне када претпостављамо да је расподела трајекторија унимодална тј. природа трајекторија је претежно детерминистичке. Неки скуповима података трајекторија имају јачу стохастичку природу због стохастичке природе самих објеката (трајекторија) или непотпуних опажања окружења. Пример таквог скупа података је скуп трајекторија пешака. Пешак који је прешао пешачки прелаз, може у том тренутку да скрене лево или десно. У том случају имају два

вероватна сценарија за исту историју трајекторије (углавном немамо информације о циљевима самог пешака). [11] [12]

Скуп „најбољи из групе“ (*eng. „Best of Many“*) техника узимају у обзир мултимодалну природу расподела трајекторија. Модел може да генерише неколико различитих предикција трајекторија и за сваку трајекторију одговарајућу вероватноћу (поузданост). Као грешка се узима предикција која је најбоља по дефинисаном критеријуму (критеријум не мора да се поклапа са самом мером која се користи). [12] [1] Претходно наведене технике *ADE* и *FDE* се уопштавају у *minADE* и *minFDE*. Због једноставности узимају се у обзир облици са једним објектом: [13] [12]

$$minADE = ADE(\arg\min_{\hat{T}^k_{raj}} FDE(\hat{T}^k_{raj}, T_{raj}), T_{raj}), k \in \{1, ..., K\}$$

$$minFDE = \min FDE(\hat{T}^k_{raj}, T_{raj}), k \in \{1, ..., K\}$$

Уколико модел генерише више од K трајекторија, узима се и обзир првих K по поузданости. У специјалном случају када је $K = 1$, онда *minFDE* постаје *FDE*, а *minADE* се и даље разликује по избору „главне“ трајекторије. Проблем са *minADE* і *minFDE* је у томе што не узимају у обзир остале трајекторије поред најбоље и самим тим се не прави разлика између предикције са свим добрим трајекторијама и предикције са једном добром трајекторијом. [13] Друга замерка овим метрикама је што не узимају у обзир поузданост предикција након филтрирања K трајекторија. Уколико је најбоља трајекторија прецизна, желимо и даље да знамо да ли је модел сигуран или је добар резултат последица „среће“. Модификоване метрике: [3]

$$p\text{–}minADE = \sum_{k=1}^{T} ADE(\hat{T}^k_{traj}, T_{raj}) - \ln P(\hat{T}^k_{traj}|E_{nv})$$

$$p\text{–}minFDE_{prob} = \sum_{k=1}^{T} FDE(\hat{T}^k_{traj}, T_{raj}) - \ln P(\hat{T}^k_{traj}|E_{nv})$$

Овде је $p(T_{traj}|E_{nv})$ условљена вероватноћа те трајекторије у односу на стање окружења. Уколико метрика *ADE* (*FDE*) има малу вредност за одговарајућу трајекторију, али њена одговарајућа вероватноћа има малу вредност, онда негативан логаритам те вероватноће има велику вредност. [1] У импементацији се ова вероватноћа ограничава са доње стране, како не би дошло до прекорачења због велике апсолутне вредности након примене логаритма на веома мале вредности.

Уместо упросечавања *L2* растојања у сваком кораку, могу да се броје кораци у којима трајекторије одступају за више од MR_{thresh}. Мотивација за ову метрику је чињеница да одступање које је 1 или 2 метра од реализације није толико релевантно у односу на велику разлику одступања. [3] Такође постоји верзија метрике која узима

у обзир вероватноћу и кажњава предикцију модела ако је добра, а модел је ипак несигуран.

$$MR = \sum_{k=1}^{T} I(||\hat{T}_{traj}^{k} - T_{raj}||_2 \geq MR_{thresh})$$

$$MR_{prob} = \sum_{k=1}^{T} I(||\hat{T}_{traj}^{k} - T_{raj}||_2 \geq MR_{thresh}) \\ +I(||\hat{T}_{traj}^{k} - T_{raj}||_2 < MR_{thresh}) \cdot (1.0 - P(\hat{T}_{traj}^{k}|E_{nv}))$$

У случају *Argoverse* скупа података, за параметар MR_{thresh} се узима 4 пиксела тј. 2 метра у реалном свету.

Све до сада наведене метрике су опште примене на било које објекте за које се предвиђају трајекторије. Пошто је агент увек возило, може да се анализира да ли предикција трајекторија скреће са пута. Због тога се уводи метрика „сагланост са продучјем вожње“ (*eng. DAC - Drivable Area Compilance*), која одређује учесталост трајекторија које нису скренуле са пута од изабраних K трајекторија: [1]

$$DAC = \frac{DAC_{occurences}}{T}$$

У евалуацији модела се узимају у обзир све метрике. За параметар K се узима вредност 6.

Глава 5

Техника заснована на разумевању контекста обрадом растеризоване сцене

У изради...

Глава 6

Техника заснована на разумевању контекста обрадом сцене представљене графом

У изради...

Глава 7

Евалуација примењених техника

У изради...

Глава 8

Закључак

У изради...

Библиографија

[1] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019. Arxiv preprint: 2103.11624.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces."

[3] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," 2021. Arxiv preprint: 2105.10968.

[4] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points (centernet)," 2019. Arxiv preprint: 1904.07850.

[5] M. Schafer, K. Zhao, M. Buhren, and A. Kummert1, "Context-aware scene prediction network (caspnet)," 222. Arxiv preprint: 2101.06933.

[6] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015. Arxiv preprint: 1506.04214.

[7] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *ECCV*, 2020. Arxiv preprint: 2007.13732.

[8] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *CVPR*, 2020. Arxiv preprint: 2005.04259.

[9] M. Ye, T. Cao, and Q. Chen, "Tpcn: Temporal point cloud networks for motion forecasting," in *CVPR*, 2021. Arxiv preprint: 2103.03067.

[10] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017. Arxiv preprint: 1612.00593.

[11] Gupta, Agrim, Johnson, Justin, Fei-Fei, Li, Savarese, Silvio, Alahi, and Alexandre, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, no. CONF, 2018.

[12] M. F. Apratim Bhattacharyya, Bernt Schiele, "Accurate and diverse sampling of sequences based on a "best of many" sample objective," in *CVPR*, 2018. Arxiv preprint: 1806.07772.

[13] Chen, Guangyi, Li, Junlong, Zhou, Nuoxing, Ren, Liangliang, Lu, and Jiwen, "Personalized trajectory prediction via distribution discrimination," in *ICCV*, 2021.

Биографија аутора

Момир Аџемовић У изради...