

Исследование графовой нейросети на простейшей задаче

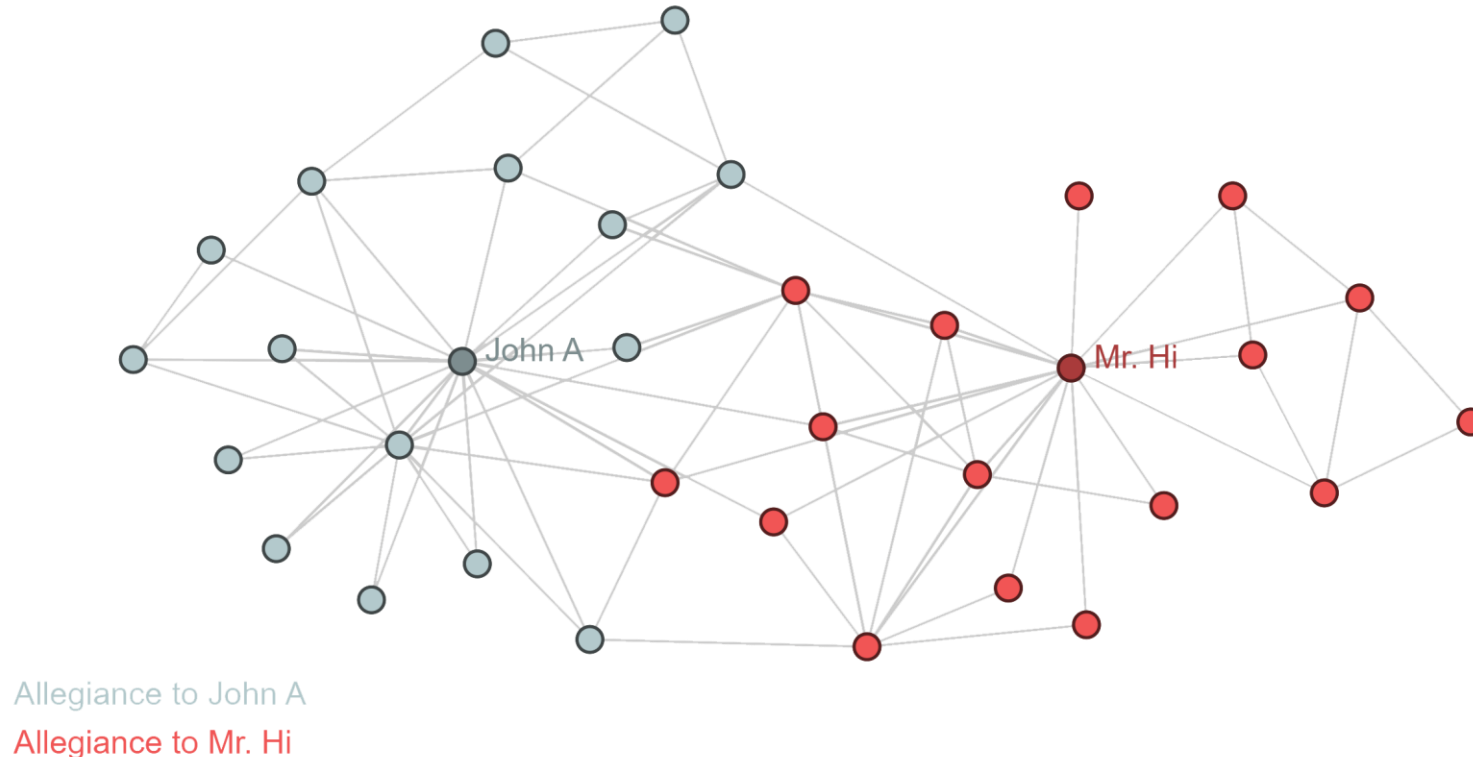
Андрей Куприн 622

Выжимка из прошлых выступлений

Графы

Графом называют совокупность множества вершин и множества ребер, их соединяющих.
Введем следующие обозначения:

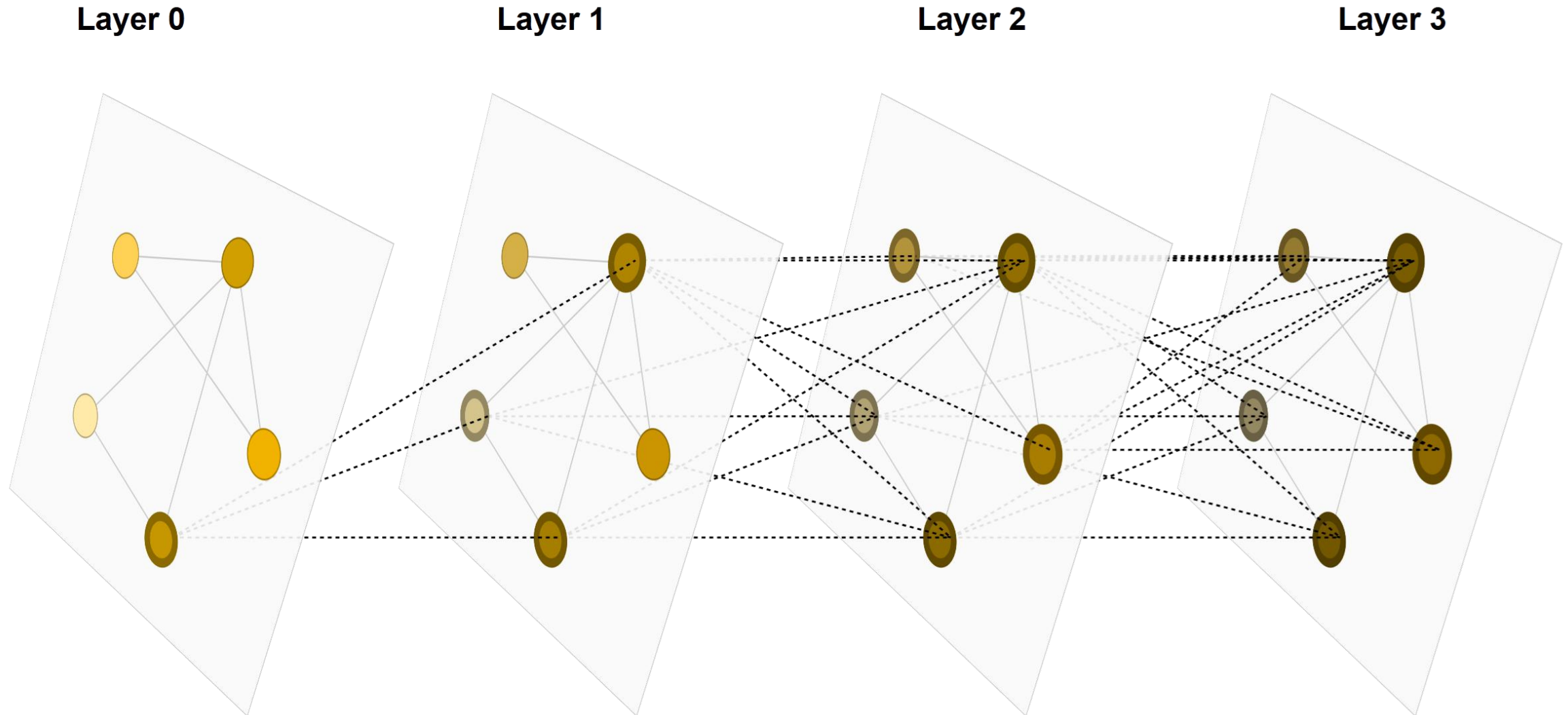
- V_n – атрибуты вершин графа;
- E_n – атрибуты ребер графа;
- U_n – глобальные атрибуты графа, то есть его атрибуты как единого целого.



Многие задачи сводятся к предсказанию атрибутов графа

Графовые нейросети (GNN)

Графовая нейросеть (GNN) – это дифференцируемое преобразование всех атрибутов графа



Графовая нейронная сеть

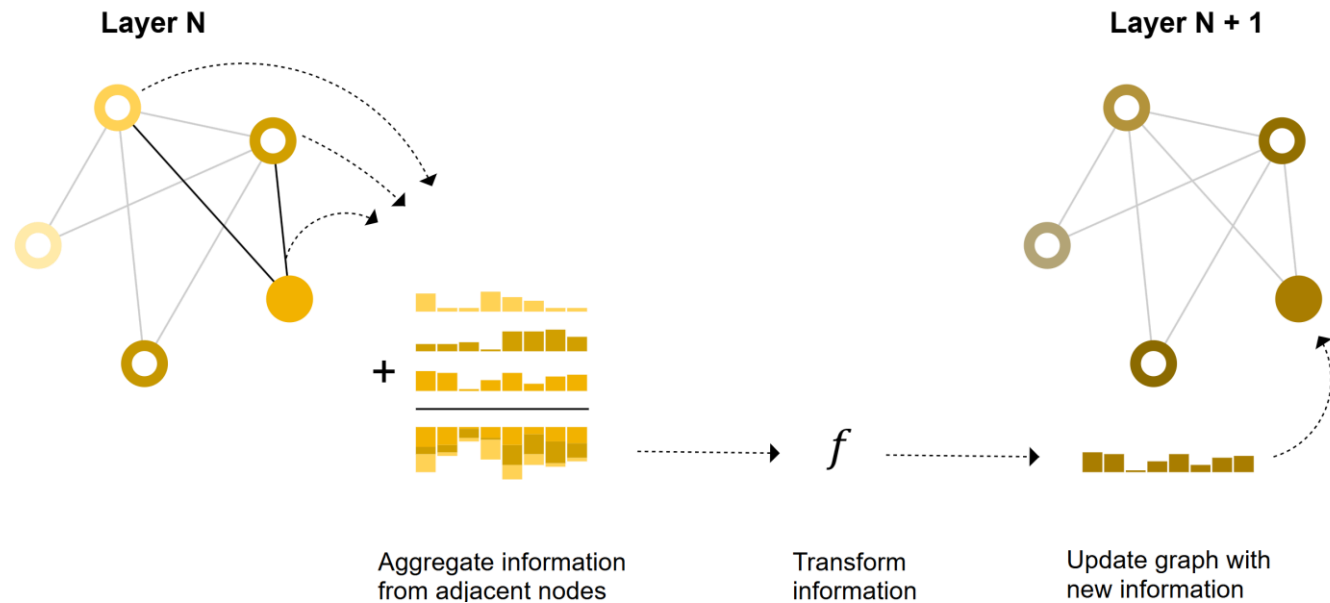
Message passing

Попробуем использовать информацию о структуре графа, чтобы атрибуты смежных вершин и ребер могли влиять друг на друга. Для этого используется техника под названием **message passing**.

Рассмотрим ее на примере вершин:

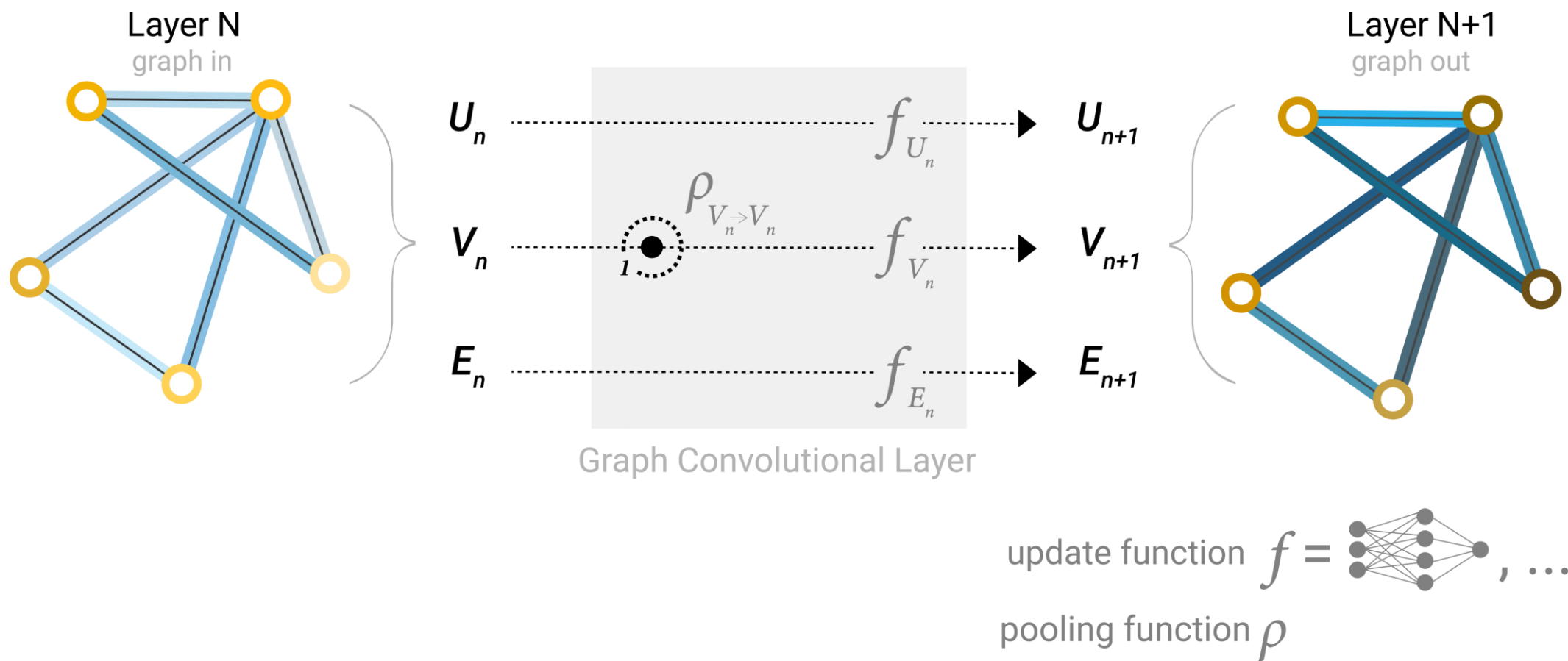
1. Для каждой вершины графа агрегируем атрибуты смежных с ней вершин;
2. Преобразовываем получившееся сообщение с помощью некоторой обучаемой функции f ;
3. Обновляем атрибуты вершины.

Аналогичные рассуждения справедливы и для ребер графа.



Message passing на примере вершин

Техника message passing напоминает **сверточный слой** – атрибуты элемента графа обновляются с учетом атрибутов соседей. Применив N таких слоев последовательно, мы можем организовать обмен сообщениями между частями графа, расположенными в N шагах друг от друга.

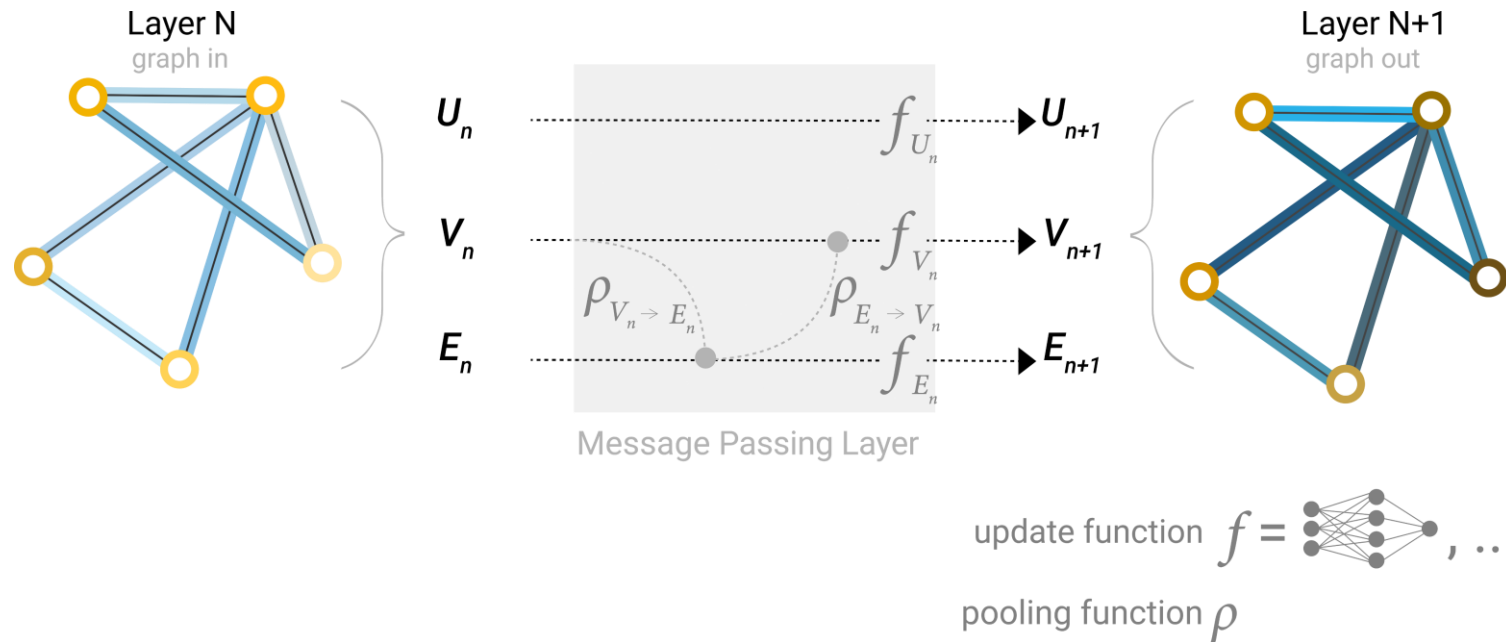


Message passing на примере вершин

Learned representations

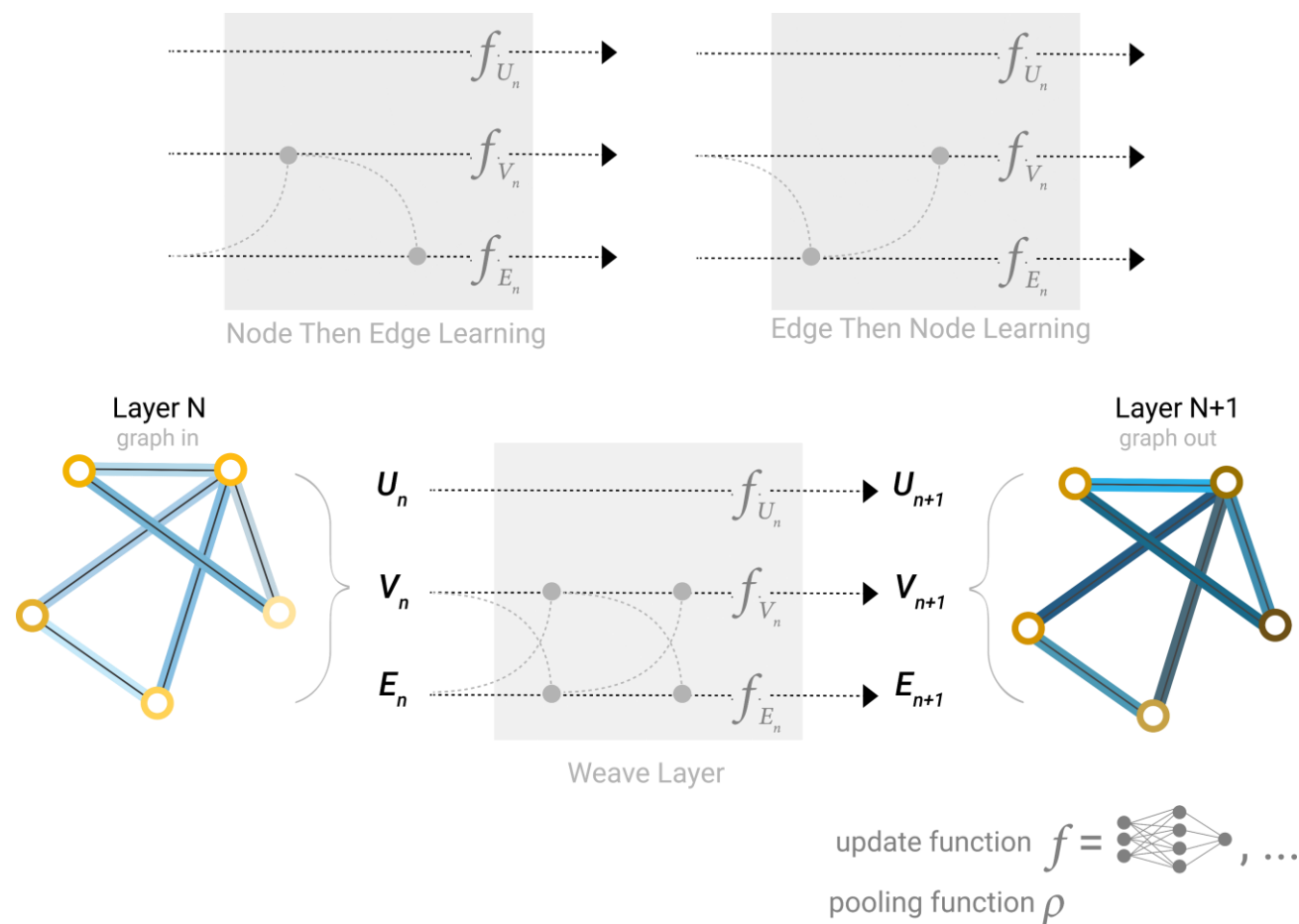
Выше мы показали, как техника message passing позволяет обмениваться сообщениями однотипным элементам графа. Но мы можем **адаптировать ее и для обмена информацией между элементами разного типа**. Пусть требуется изменить атрибуты вершин с учетом атрибутов смежных ребер. Тогда:

1. Для каждой вершины графа агрегируем атрибуты смежных с ней ребер;
2. Преобразовываем получившееся сообщение с помощью обучаемой функции f , которая отображает вектор из пространства атрибутов ребра в пространство атрибутов вершины;
3. Обновляем атрибуты каждой вершины с учетом сообщения.



Learned representations

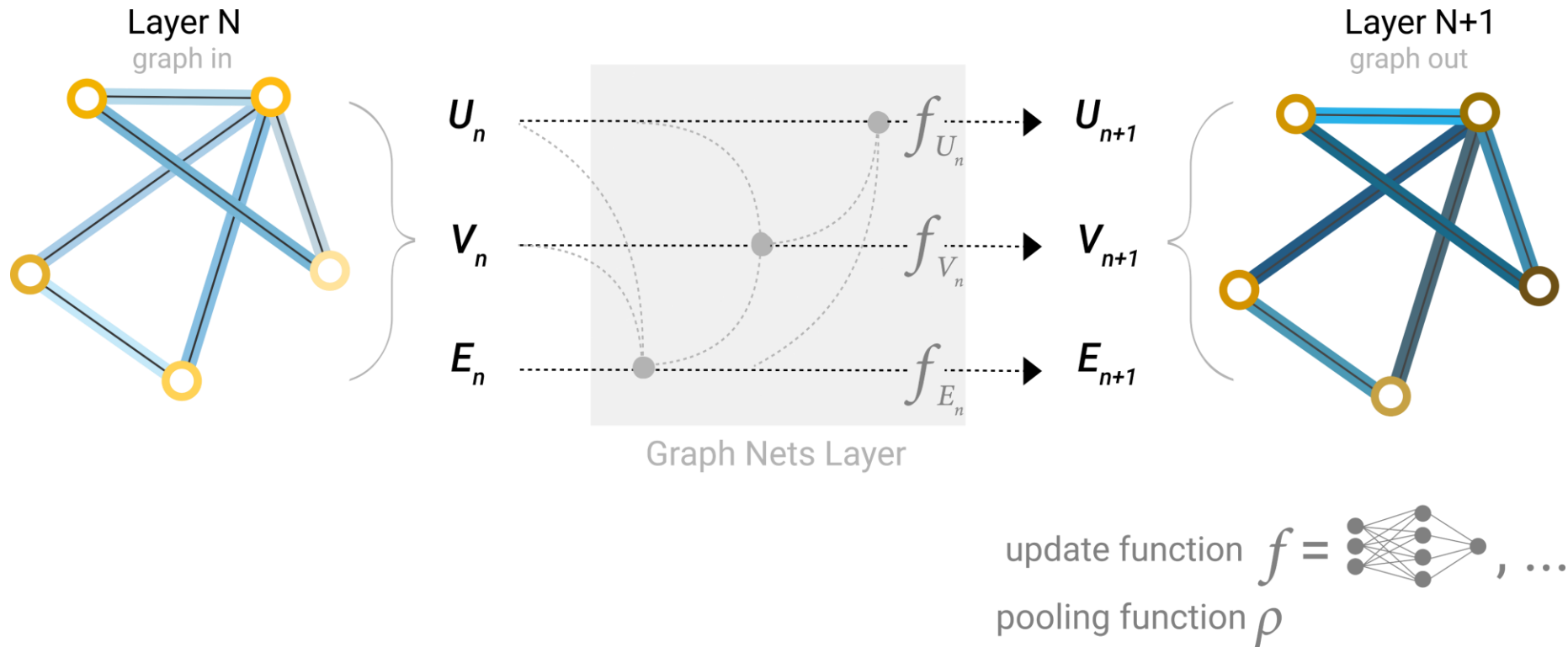
Данная техника получила название **learned representations**. Таким образом, мы можем создать слой для обмена сообщениями между вершинами и ребрами графа. В каком порядке и между какими элементами отправлять эти сообщения – открытый вопрос. Разработано множество вариантов такого слоя.



Различные варианты применения learned representations

Global representations

У техники message passing есть один недостаток – сложно организовать эффективный обмен сообщениями между элементами графа, которые находятся не далеко друг от друга. Глобальные атрибуты графа могут не только описывать свойства графа как единого целого, но и выступать в качестве «моста» между далекими его элементами. Поэтому технику learned representations можно применить и для **обмена сообщениями с глобальными атрибутами графа**.



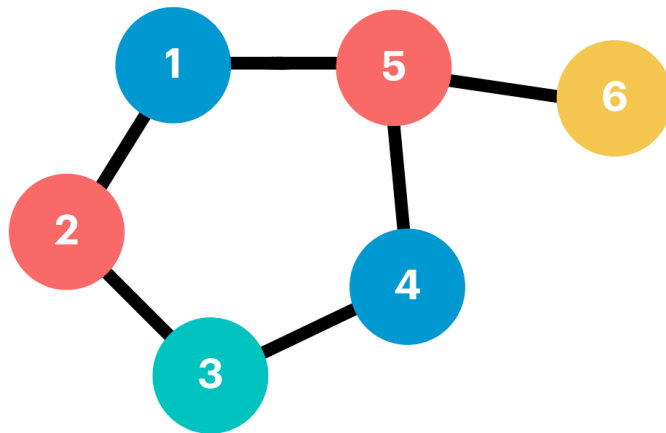
Пример использования global representations

Преимущества GNN

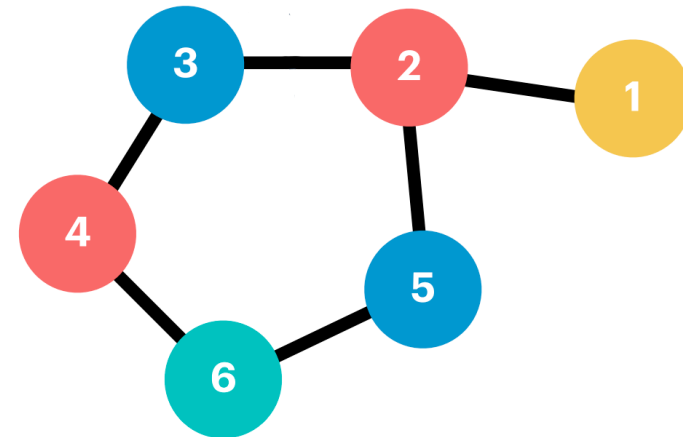
Отметим две **ключевые особенности GNN**, которые позволяют им эффективно решать некоторые задачи:

- **Локальность** – подобно свертке в сетях, работающих с изображениями, техника message passing позволяет учесть, что непосредственные соседи влияют на вершину больше, чем далекие части графа;
- **Инвариантность к перестановке вершин** – если пронумеровать вершины графа в другом порядке, предсказания графовой нейросети об атрибутах вершин останутся теми же с точностью до перестановки, а предсказание о графе как едином целом, не изменится;
- **Полнота** – доказывается, что при правильном выборе атрибутов и достаточном количестве слоев GNN может описать любое преобразование над графом.

$E = 0.5 \text{ kcal mol}^{-1}$



$E = 0.5 \text{ kcal mol}^{-1}$

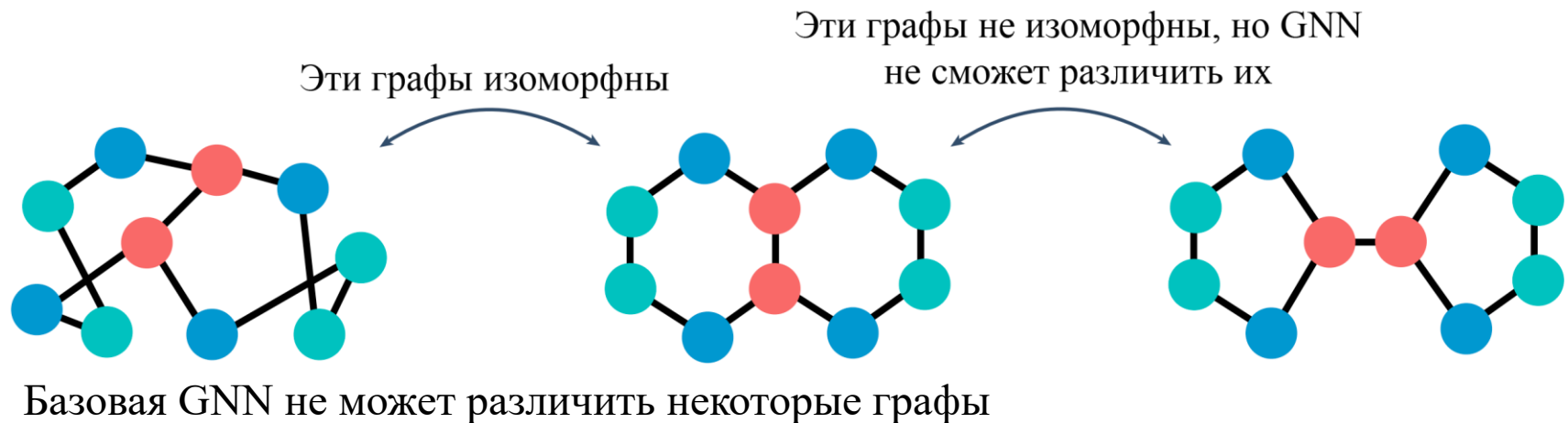


Инвариантность к перестановке при предсказании внутренней энергии молекулы

Недостатки GNN

Однако в базовом виде GNN имеют ряд недостатков:

- **Выразительность на глобальном уровне** – GNN воспринимает некоторые пары графов, как изоморфные, хотя это не так (случайные атрибуты, позиционное кодирование);
- **Выразительность на локальном уровне** – GNN не способна предсказать количество колец и клик в графе (вручную созданные начальные атрибуты, позиционное кодирование);
- **Пересглаживание** – с ростом числа слоев атрибуты все сильнее усредняются и становятся неотличимыми друг от друга (подмешивание оригинальных значений, ограничение распространения сообщений);
- **Пересжатие** – в графе могут быть узкие места. В этом случае в одну вершину, связующую две части графа, стекается очень много информации, которую приходится сжимать (глобальная вершина, перестройка вычислительного графа).



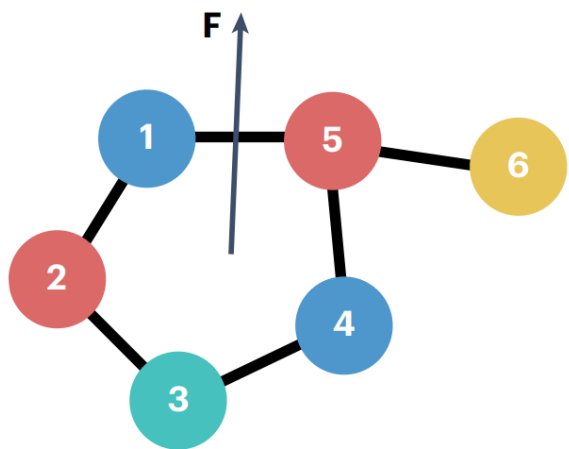
Геометрические графовые нейронные сети

Если граф является представлением некоторого геометрического объекта, и часть признаков каждого узла это его координаты в трехмерном пространстве, требуется чтобы GNN была **устойчива** не только к перестановкам вершин, но и к движению графа как единого целого.

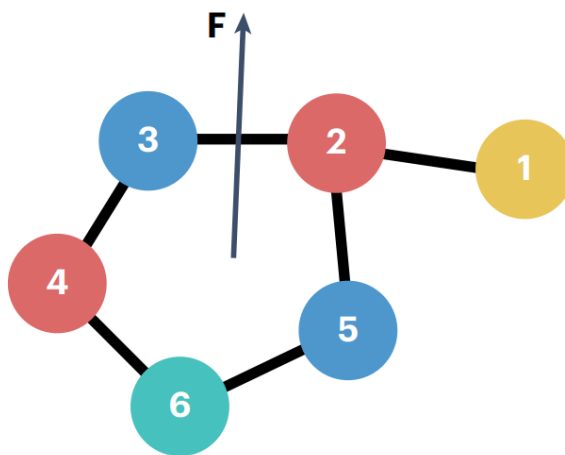
Что мы можем взять в качестве атрибутов вершин:

- **Абсолютные координаты вершин** → неустойчивость к перемещению и вращению графа;
- **Относительные координаты вершин** → неустойчивость к вращению графа;
- **Расстояния между вершинами** → потеря большого количества информации.

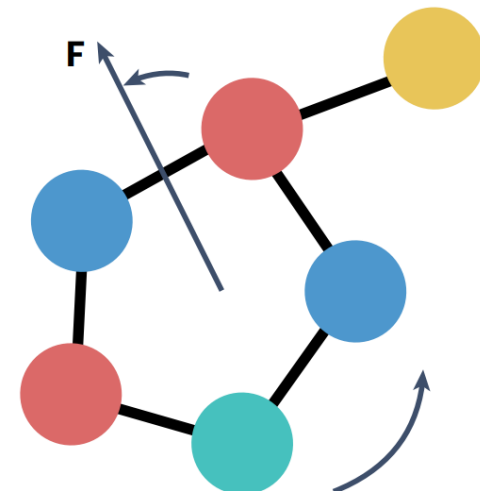
$E = 0.5 \text{ kcal mol}^{-1}$



$E = 0.5 \text{ kcal mol}^{-1}$



$E = 0.5 \text{ kcal mol}^{-1}$



Устойчивость GNN к движению графа как единого целого

Пример геометрической графовой нейронной сети

EGNN

Примером геометрической графовой нейронной сети может служить архитектура **EGNN**, предложенная в 2022 году. Доказывается, что данная архитектура **устойчива к перестановке вершин графа, а также вращению и перемещению графа как единого целого**. Слой данной архитектуры выглядит следующим образом:

$$\begin{aligned}m_{ij} &= \varphi_e(h_i^l, h_j^l, \|x_i^l - x_j^l\|, a_{ij}), \\v_i^{l+1} &= \varphi_v(h_i^l)v_i^{init} + C \sum_{i \neq j} (x_i^l - x_j^l)\varphi_x(m_{ij}) \\x_i^{l+1} &= x_i^l + v_i^{l+1}, \\m_i &= \sum_{i \neq j} m_{ij}, \\h_i^{l+1} &= \varphi_h(h_i^l, m_i)\end{aligned}$$

где

$h_i^l \in R^{nf}$ – **атрибуты вершины** v_i на слое l ,

$x_i^l \in R^n$ – **координаты вершины** v_i на слое l ,

$v_i^l \in R^n$ – **скорость вершины** v_i на слое l ,

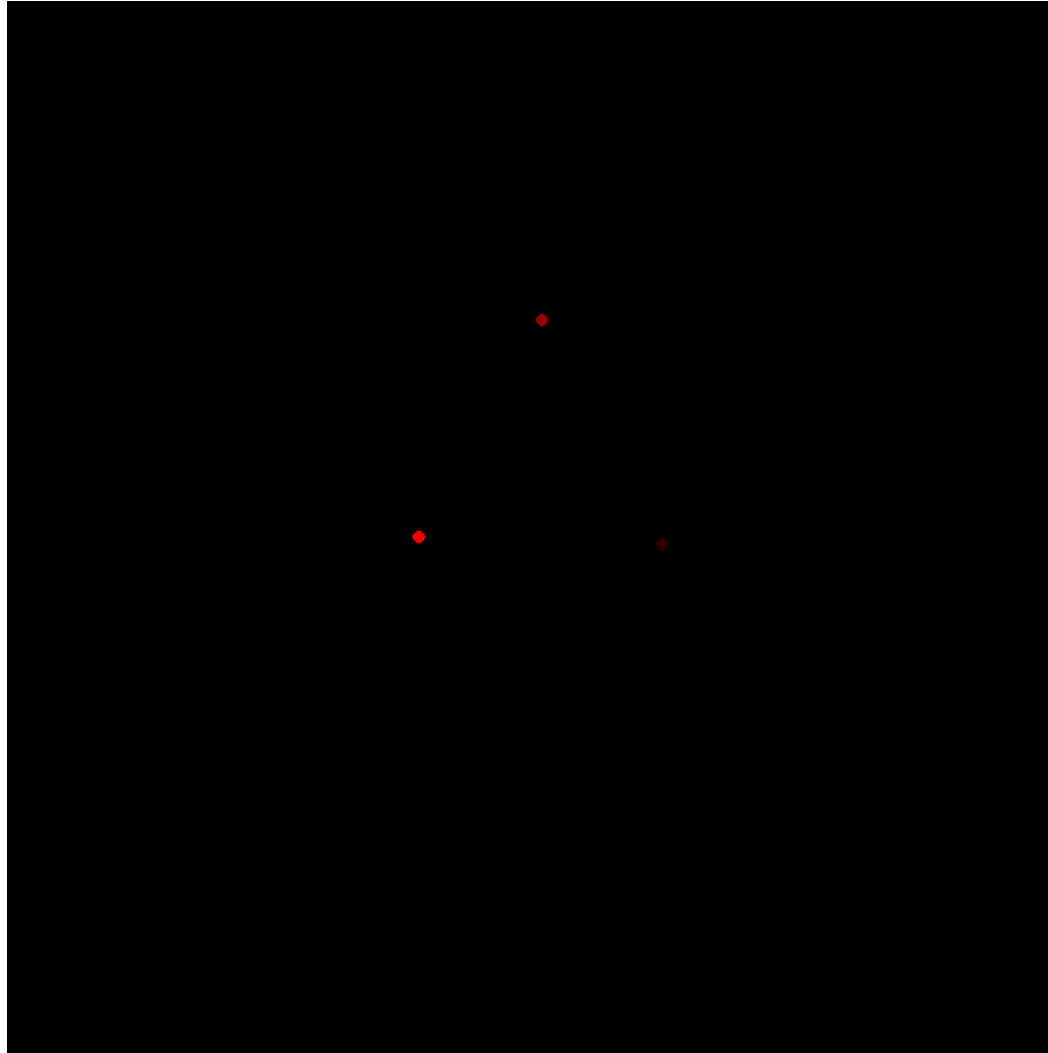
$m_{ij} \in R^{nf}$ – **атрибуты, агрегированные на ребре**,

a_{ij} – **атрибуты ребра**, находящегося между вершинами v_i и v_j ,

$N(i)$ – **соседи вершины** v_i .

Задача

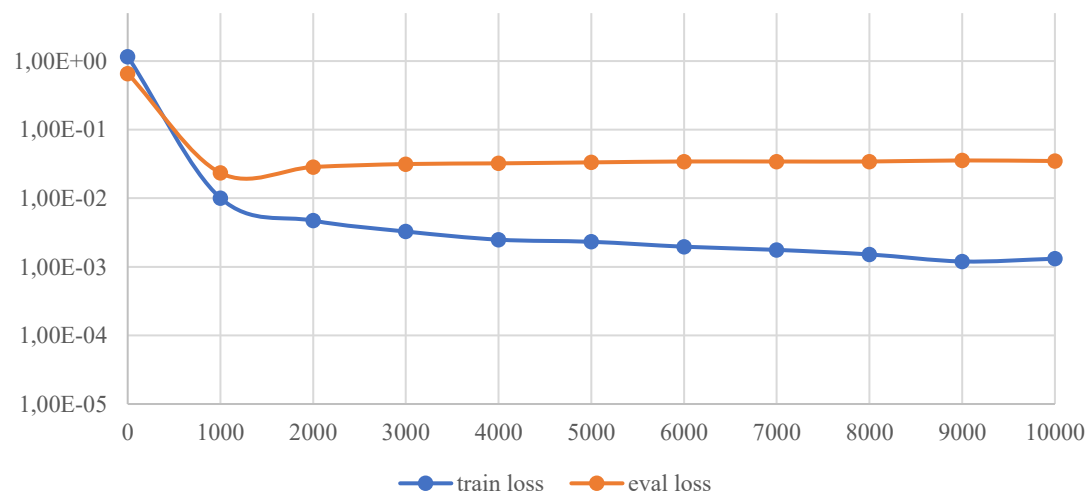
Попробуем применить EGNN на практике. Рассмотрим **задачу движения трех тел** одинаковой массы и попробуем по начальным условиям предсказать состояние системы через **500 шагов**.



Задача трех тел

Этапы решения задачи:

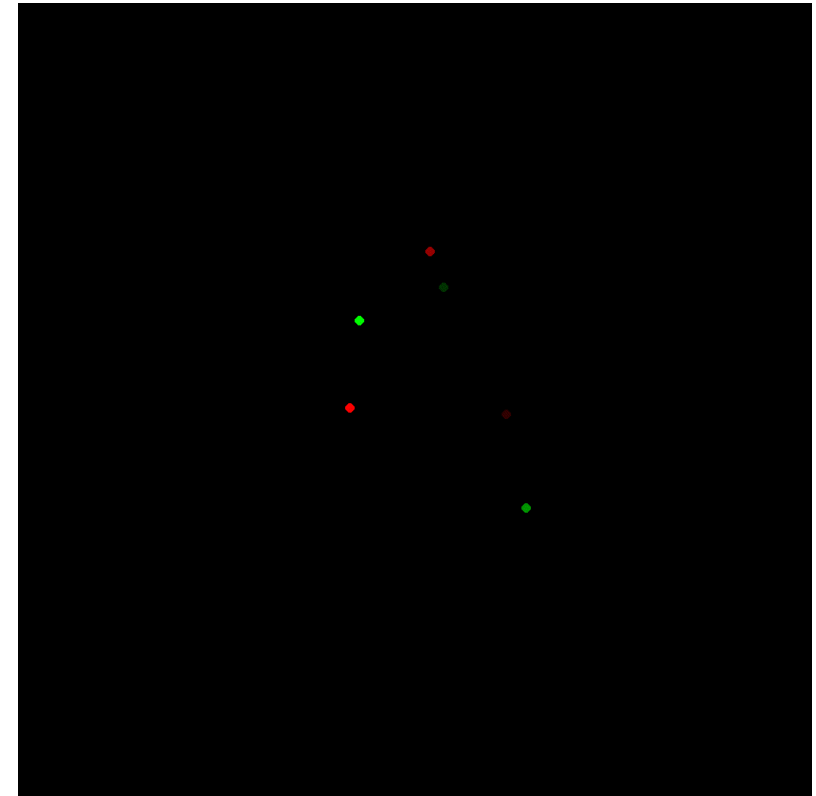
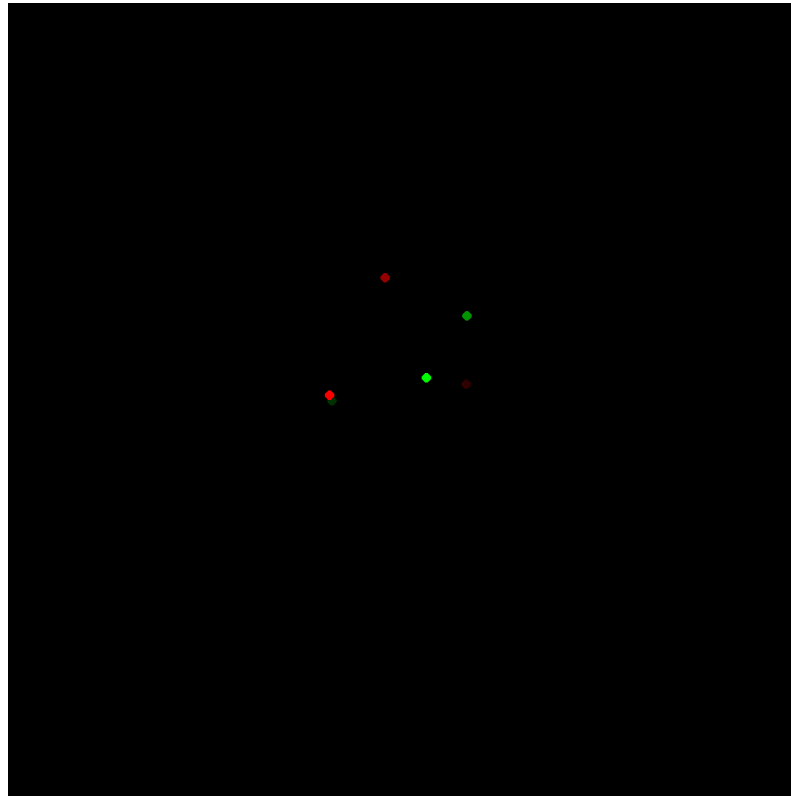
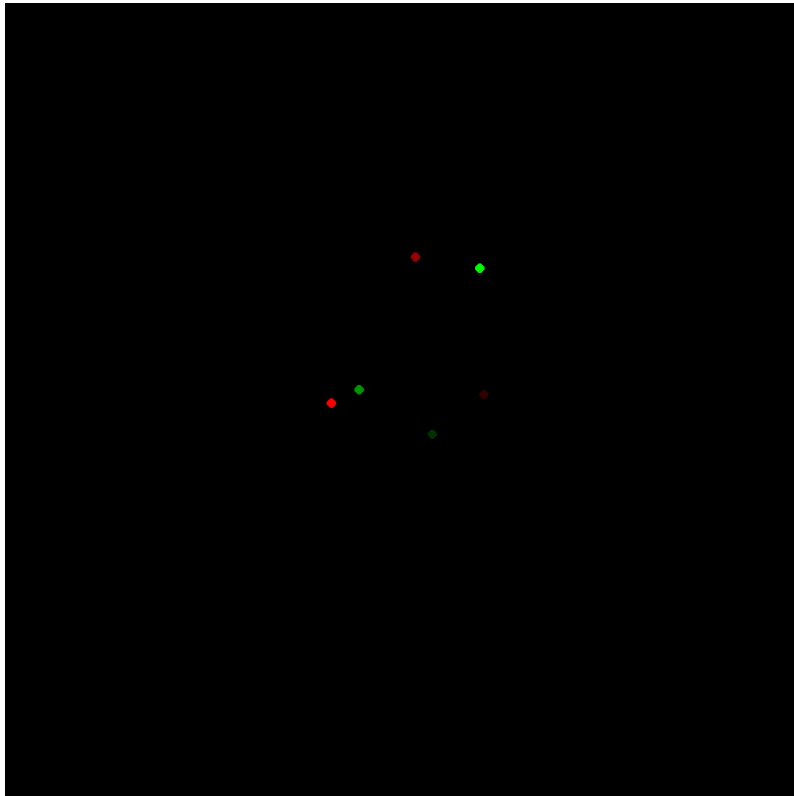
1. **Численным методом** рассчитаем 10.000 траекторий, длиной 500 шагов. 8.000 траекторий используем для обучения, а остальные — для валидации (простой симулятор с GitHub);
2. Из каждой траектории возьмем **координаты**, а также **векторы скоростей** для первого шага и **координаты для последнего** (всего 18 чисел);
3. Будем подавать в нейронную сеть **координаты, векторы скоростей**, а также **попарные расстояния**. Будем считать граф **полным**. В качестве loss функции будем использовать **MSE** (параметры нейросети взяты из оригинальной статьи);
4. Выполним **10.000 итераций** (параметры тренировки взяты из оригинальной статьи, 3 часа на RTX 3070).



Процесс обучения

Результат

Удается более-менее точно предсказывать состояние системы после 500 шагов.

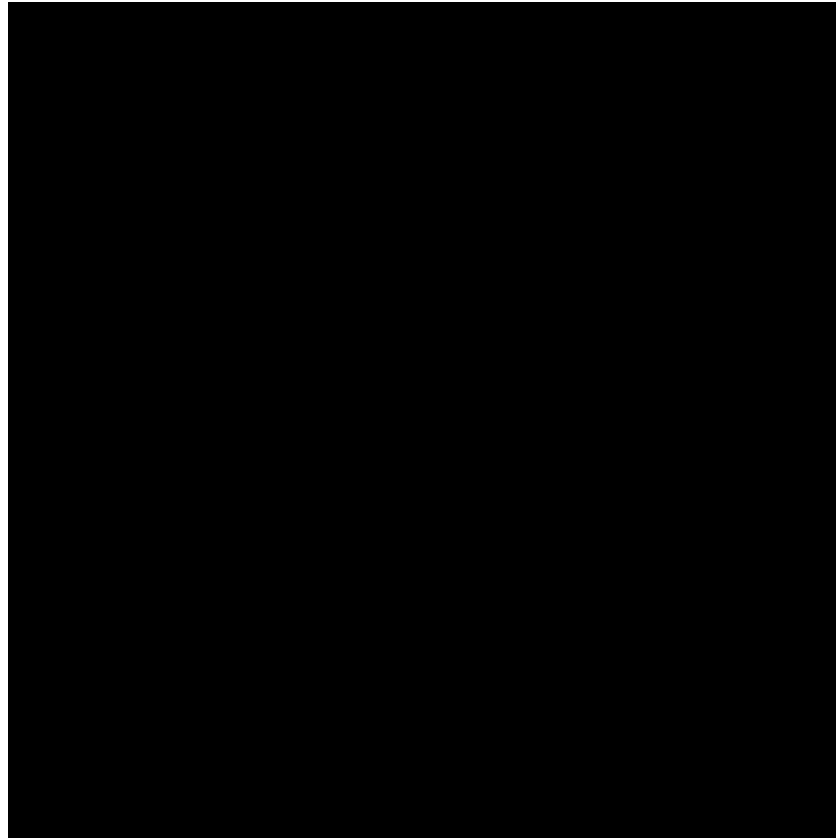


Задача трех тел

Ограничения текущей реализации

Что не удалось сделать:

1. Предсказание не на **500**, а на **1 шаг** вперед (решение расходится из-за накапливающихся ошибок, так как ничем не ограничено);
2. **Коллизии** тел между собой и с границами области (нейросеть обучается очень медленно или не обучается вовсе).



Задача с коллизиями

Дальнейшие действия

Тестирование более свежей архитектуры:

- FastEGNN (Improving Equivariant Graph Neural Networks on Large Geometric Graphs via Virtual Nodes Learning, 2024);
- Aether (Latent Field Discovery In Interacting Dynamical Systems With Neural Fields, 2024);
- HEGNN (Are High-Degree Representations Really Unnecessary in Equivariant Graph Neural Networks?, 2025);
- GOTENNET (Rethinking Efficient 3D Equivariant Graph Neural Networks, 2025).

Обработка коллизий:

- Offset Geometric Contact (2025).

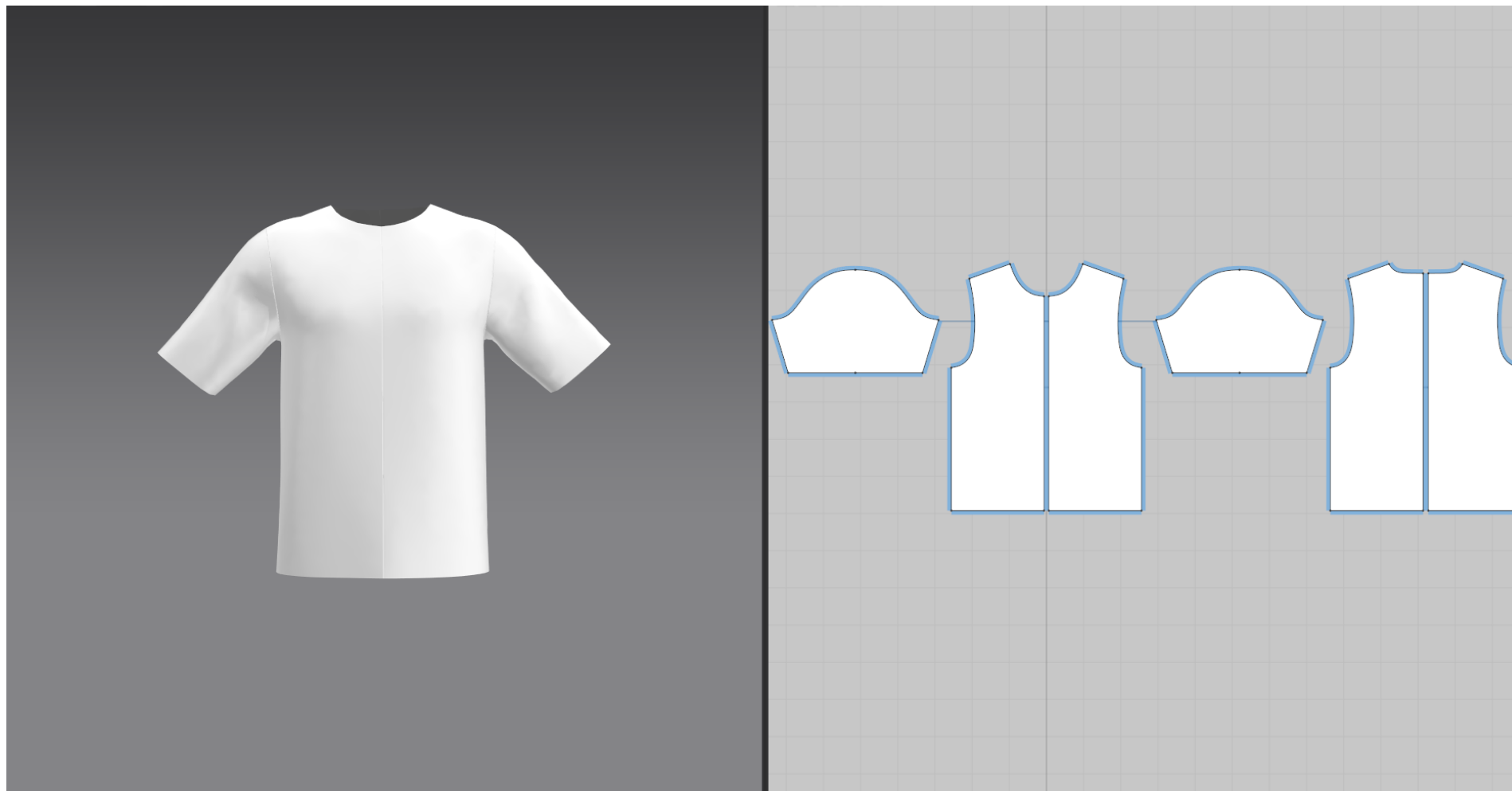
Table 3: MSE and time-consuming ratio with EGNN [1] on N -body system.

	5-body		20-body		50-body		100-body	
	MSE ($\times 10^{-2}$)	Relative Time	MSE ($\times 10^{-2}$)	Relative Time	MSE ($\times 10^{-2}$)	Relative Time	MSE ($\times 10^{-2}$)	Relative Time
Linear	7.72	0.01	10.12	0.02	11.81	0.02	12.69	0.01
MPNN [47]	1.80	0.49	2.50	0.51	2.96	0.50	3.55	0.45
SchNet [48]	11.31	2.93	17.72	6.24	22.14	31.63	22.14	27.04
RF [46]	1.51	0.54	3.41	0.65	4.75	0.67	5.72	0.49
GVP-GNN [49]	7.26	2.36	5.76	2.38	7.07	2.42	7.55	2.33
EGNN [1]	0.65	1.00	1.01	1.00	1.00	1.00	1.36	1.00
TFN _{$l \leq 2$}	1.49	2.69	1.86	3.19	2.20	2.87	3.42	6.58
TFN _{$l \leq 3$}	1.76	3.91	1.87	4.54	1.94	4.89	OOM	-
SE(3)-Tr. _{$l \leq 2$}	3.24	4.94	3.19	5.88	2.54	5.97	2.33	5.15
HEGNN _{$l \leq 1$}	0.52	1.77	0.79	1.84	0.88	1.60	1.13	1.45
HEGNN _{$l \leq 2$}	0.47	1.88	0.78	1.94	0.90	1.71	0.97	1.55
HEGNN _{$l \leq 3$}	0.48	2.11	0.80	2.23	0.84	1.84	0.94	1.61
HEGNN _{$l \leq 6$}	0.69	2.14	0.86	2.43	0.96	2.18	0.86	1.90

Более свежие архитектуры

Переход к **сверточной нейросети**?

- TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style (2020).



Выкройка ткани

E(n) Equivariant Graph Neural Networks

Victor Garcia Satorras¹ Emiel Hooeboom¹ Max Welling¹

Abstract

This paper introduces a new model to learn graph neural networks equivariant to rotations, translations, reflections and permutations called E(n)-Equivariant Graph Neural Networks (EGNNs). In contrast with existing methods, our work does not require computationally expensive higher-order representations in intermediate layers while it still achieves competitive or better performance. In addition, whereas existing methods are limited to equivariance on 3 dimensional spaces, our model is easily scaled to higher-dimensional spaces. We demonstrate the effectiveness of our method on dynamical systems modelling, representation learning in graph autoencoders and predicting molecular properties.

1. Introduction

Although deep learning has largely replaced hand-crafted features, many advances are critically dependent on inductive biases in deep neural networks. An effective method to restrict neural networks to relevant functions is to exploit the *symmetry* of problems by enforcing equivariance with respect to transformations from a certain symmetry group. Notable examples are translation equivariance in Convolutional Neural Networks and permutation equivariance in Graph Neural Networks (Bruna et al., 2013; Defferrard et al., 2016; Kipf & Welling, 2016a).

Many problems exhibit 3D translation and rotation symmetries. Some examples are point clouds (Uy et al., 2019), 3D molecular structures (Ramakrishnan et al., 2014) or N-body particle simulations (Kipf et al., 2018). The group corresponding to these symmetries is named the Euclidean group: SE(3) or when reflections are included E(3). It is often desired that predictions on these tasks are either equivariant or invariant with respect to E(3) transformations.

¹UvA-Bosch Delta Lab, University of Amsterdam, Netherlands. Correspondence to: Victor Garcia Satorras <v.garciasatorras@uva.nl>, Emiel Hooeboom <e.hooeboom@uva.nl>, Max Welling <m.welling@uva.nl>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

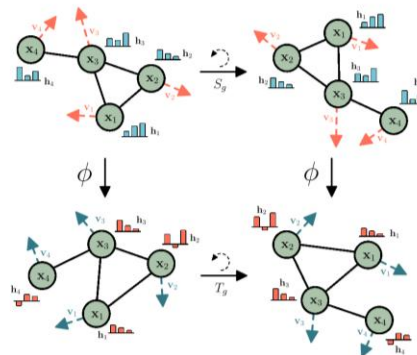


Figure 1. Example of rotation equivariance on a graph with a graph neural network ϕ

Recently, various forms and methods to achieve E(3) or SE(3) equivariance have been proposed (Thomas et al., 2018; Fuchs et al., 2020; Finzi et al., 2020; Köhler et al., 2020). Many of these works achieve innovations in studying types of higher-order representations for intermediate network layers. However, the transformations for these higher-order representations require coefficients or approximations that can be expensive to compute. Additionally, in practice for many types of data the inputs and outputs are restricted to scalar values (for instance temperature or energy, referred to as type-0 in literature) and 3d vectors (for instance velocity or momentum, referred to as type-1 in literature).

In this work we present a new architecture that is translation, rotation and reflection equivariant (E(n)), and permutation equivariant with respect to an input set of points. Our model is simpler than previous methods in that it does not require the spherical harmonics as in (Thomas et al., 2018; Fuchs et al., 2020) while it can still achieve competitive or better results. In addition, equivariance in our model is not limited to the 3-dimensional space and can be scaled to larger dimensional spaces without a significant increase in computation.