Расстояния между графами

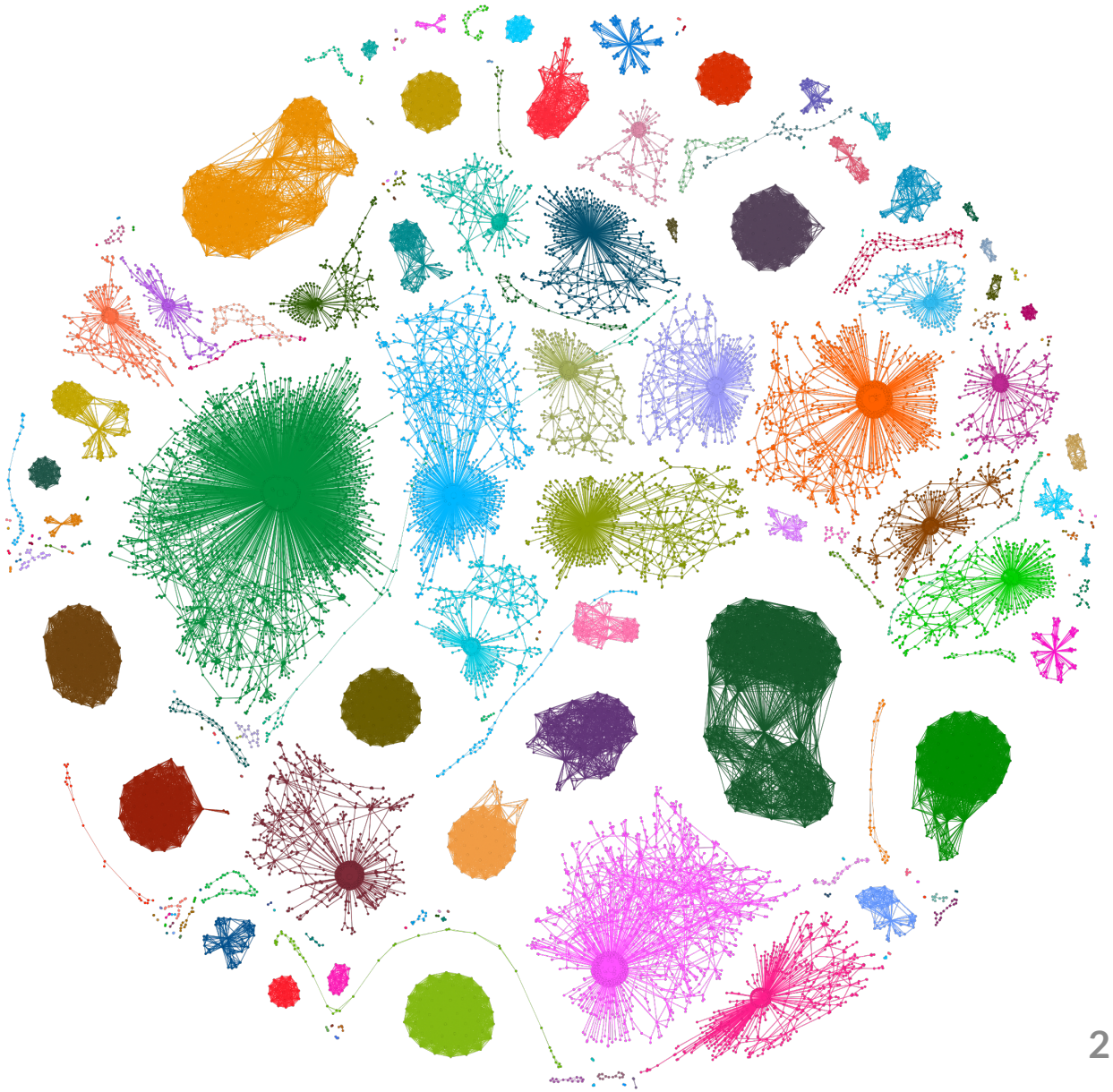
Антон Цицулин

18.03.2021

Зачем нам расстояния между графами?

Через расстояние мы можем решать разные задачи:

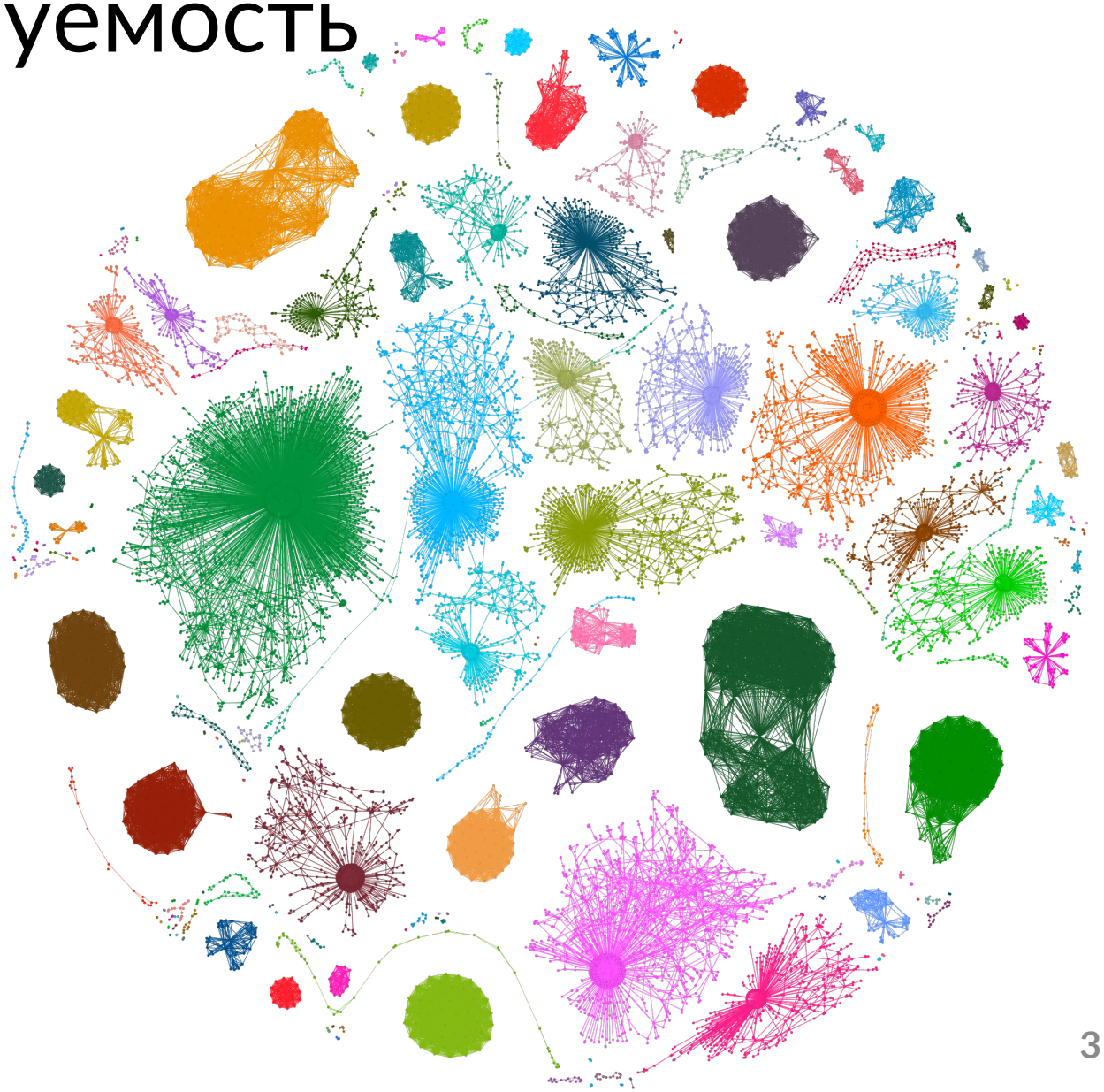
- Классификацию/регрессию
- Кластеризацию
- Поиск аномалий
- Дебаггинг пайплайнов ❤️
- ...



Главное – масштабируемость

Два источника проблем:

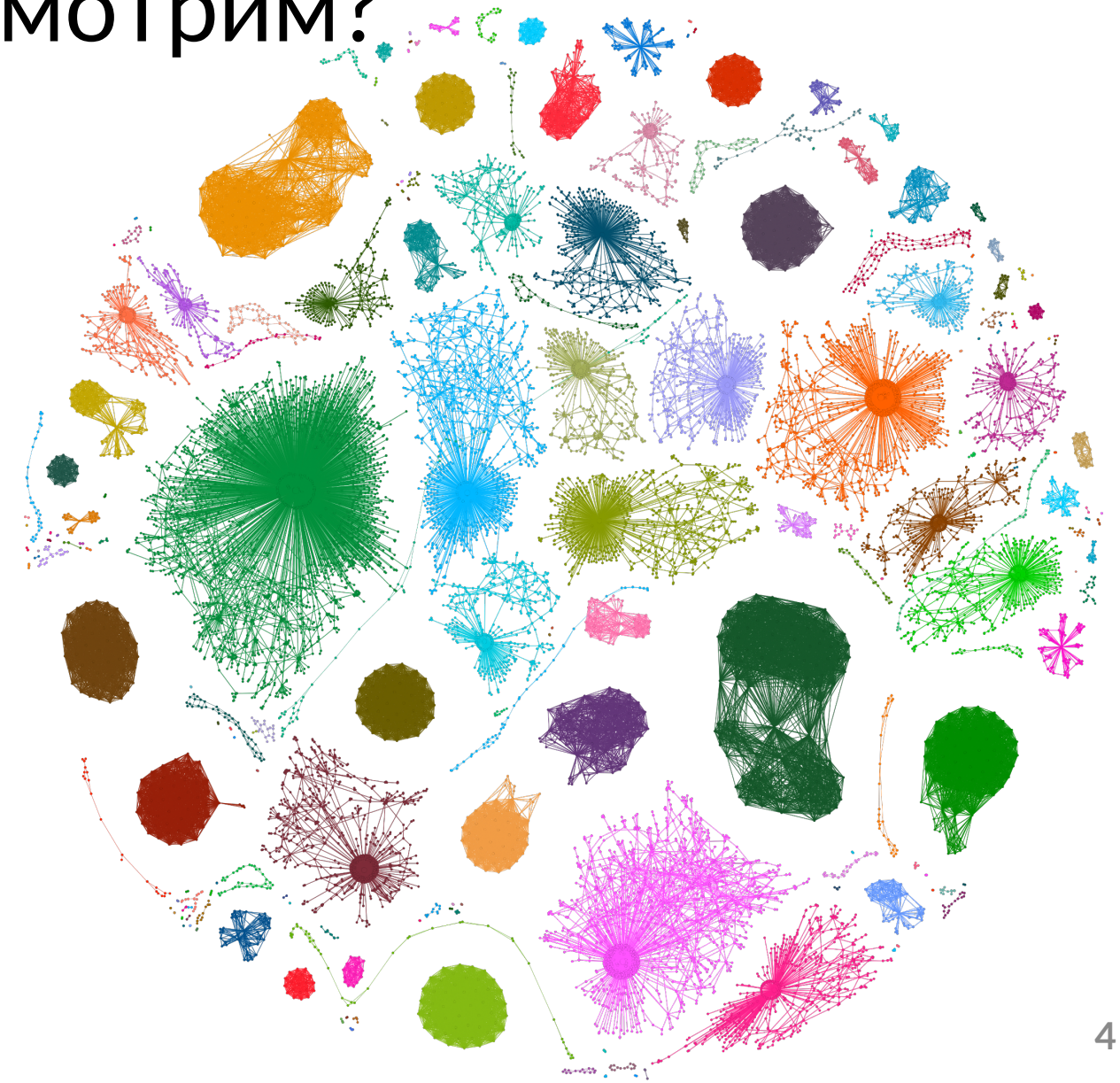
- Большие графы
- и куча маленьких



Какие графы мы рассмотрим?

Основные типы:

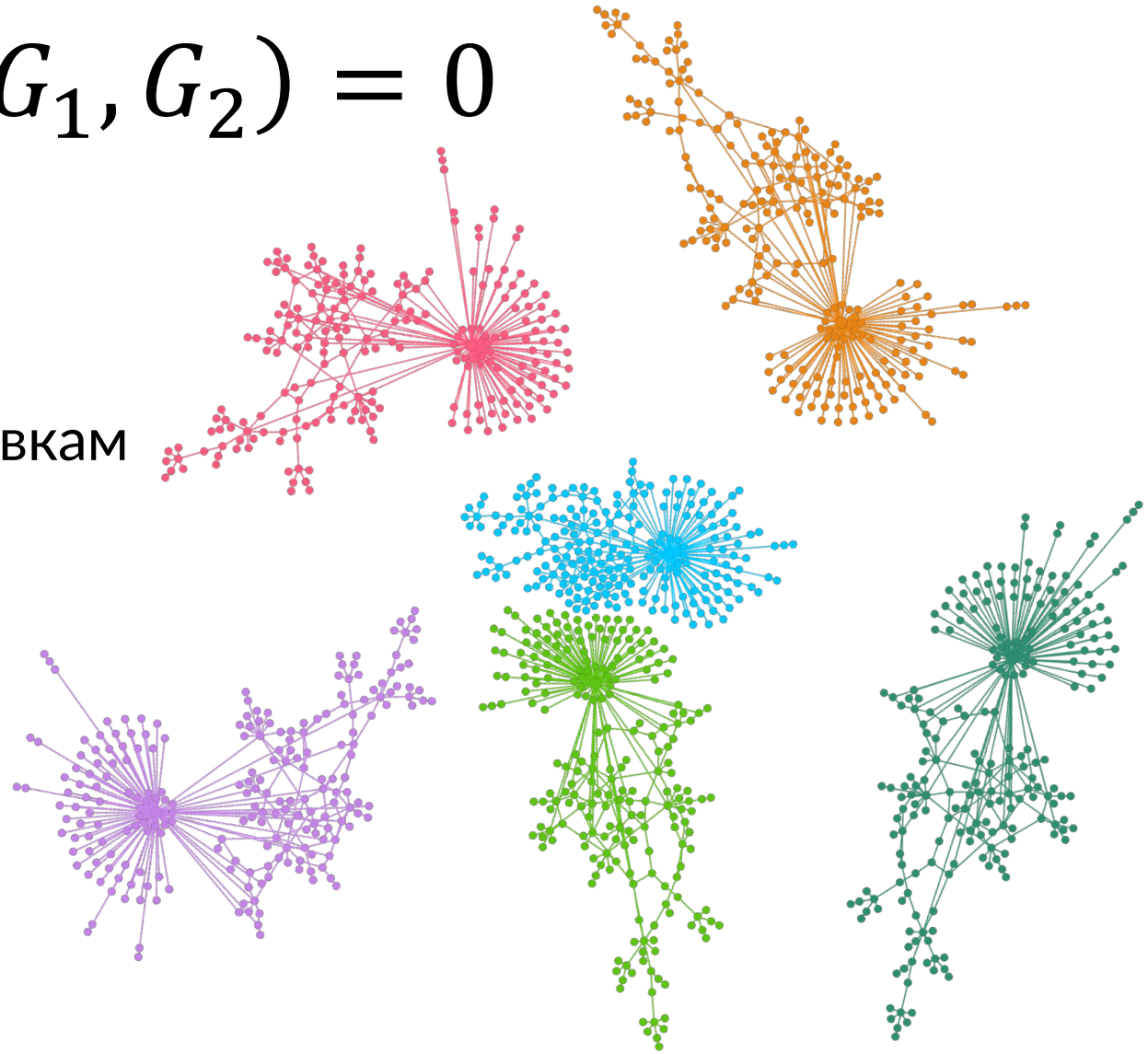
- Без атрибутов
- С атрибутами на вершинах/рёбрах
- Экзотику – не будем
  - signed, temporal, …



Изоморфизм $\Rightarrow d(G_1, G_2) = 0$

3 ключевых свойства:

- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру

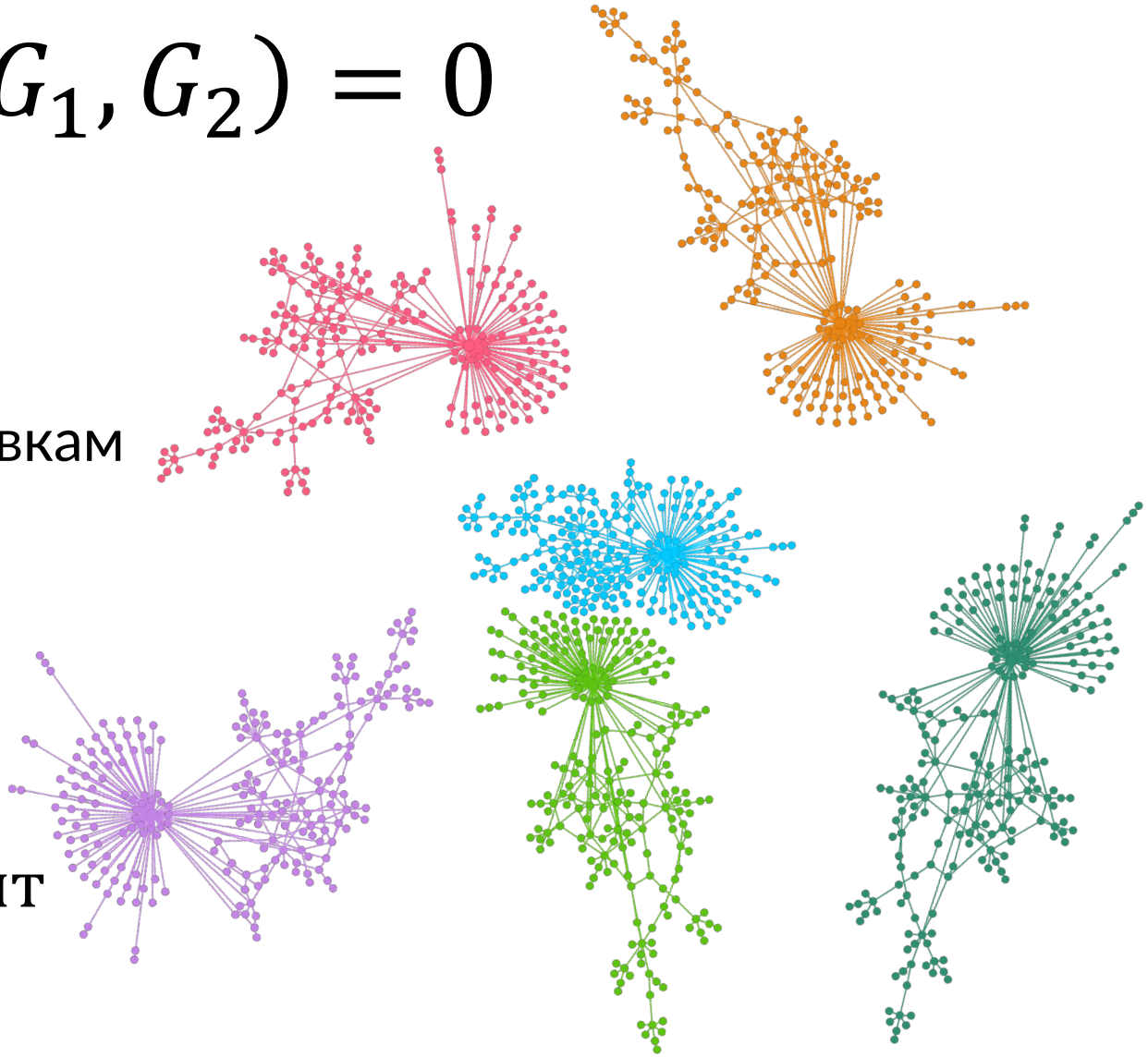


Изоморфизм $\Rightarrow d(G_1, G_2) = 0$

3 ключевых свойства:

- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру

$\Rightarrow$ метрики нам ждать не стоит



А если попробовать (почти-)изоморфизм?

Два пути:

- 💔 Расстояния редактирования графов
  - NP-полная задача, APX-сложная
  - ОЧЕНЬ ПЛОХО аппроксимируются (100 вершин – радость)
- ❤️‍🩹 Алгебры на графах (Вейсфейлер–Леман)
  - Быстрые, классные алгоритмы
  - Без претензий на теоретические гарантии

Алгоритм Вейсфейлера–Лемана

Параметризованный* алгоритм для задачи изоморфизма:

1. Задаём «цвет» h_{v_i} для всех v

2. Для каждой вершины v задаём $h_{v_i} = hash(h_{v_i} + \sum_{j \in N(i)} h_{v_j})$

3. Повторяем до сходимости

*Параметризация – хеширование комбинаций вершин

Ядра Вейсфейлера–Лемана

Параметризованное* ядро для задачи расстояния графов:

1. Задаём «цвета» как степени вершин

2. Считаем «цветность» вершин до сходимости

3. Перенумеровываем все «цвета», считаем гистограмму

4. Ядро – скалярное произведение гистограмм

Ядра Вейсфейлера–Лемана

Хорошо работают с категориальными фичами

Без фичей тоже работает

Считаются быстро – $O(hm), h$ – количество итераций

Всё ещё ядра – $O(n^2)$ на обучение с SVM

Что если забыть про изоморфизм?

Считаем статистики вершин, у похожих графов они совпадают

Статистики инвариантны к перестановкам

Хорошо работает для поиска аномалий и дебаггинга

Пример – NetSimile

NetSimile: и без теории сойдёт

1. Собираем описания вершин

   Количество соседей, коэффициент кластеризации, средняя степень соседей, …

2. Считаем статистики по этим описаниям

   Среднее, медиана, коэффициенты асимметрии и эксцесса, …

3. Сравниваем эти статистики

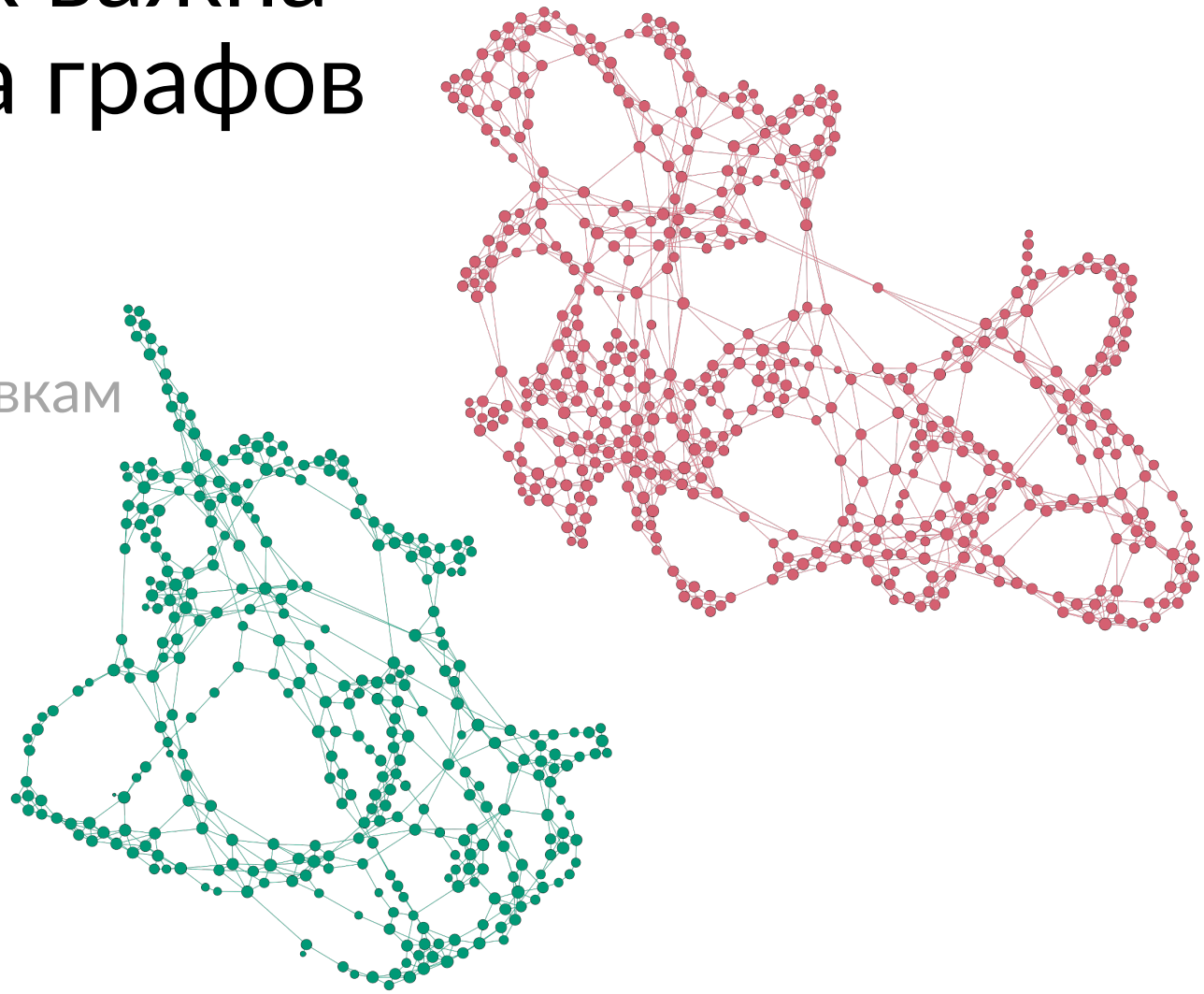
Сложность: $O(n\log(n) + m^{3/2})$

❤️ интерпретируемость; 💔 выразительность

В некоторых задачах важна локальная структура графов

3 ключевых свойства:

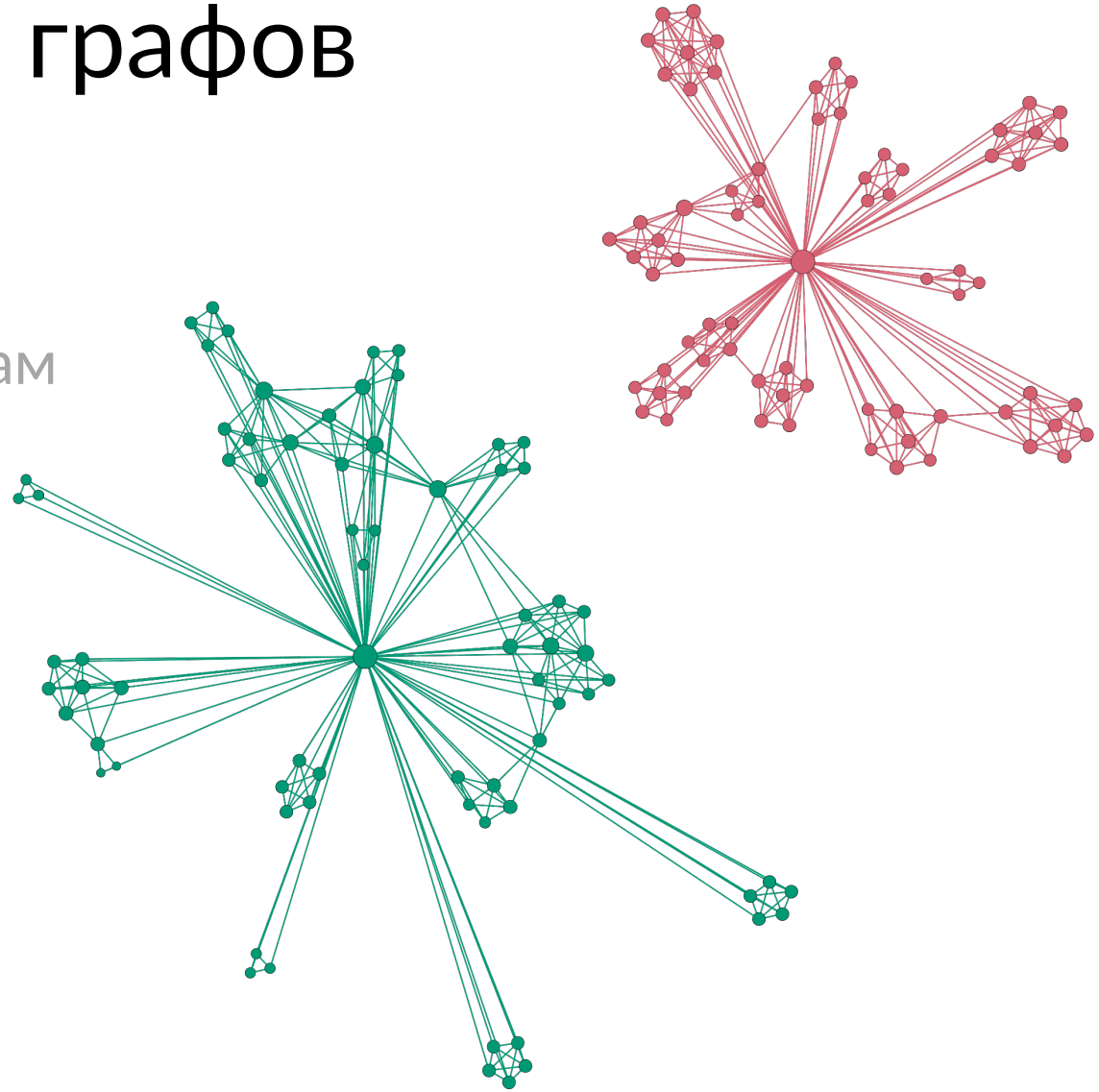
- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру



В некоторых задачах важна глобальная структура графов

3 ключевых свойства:

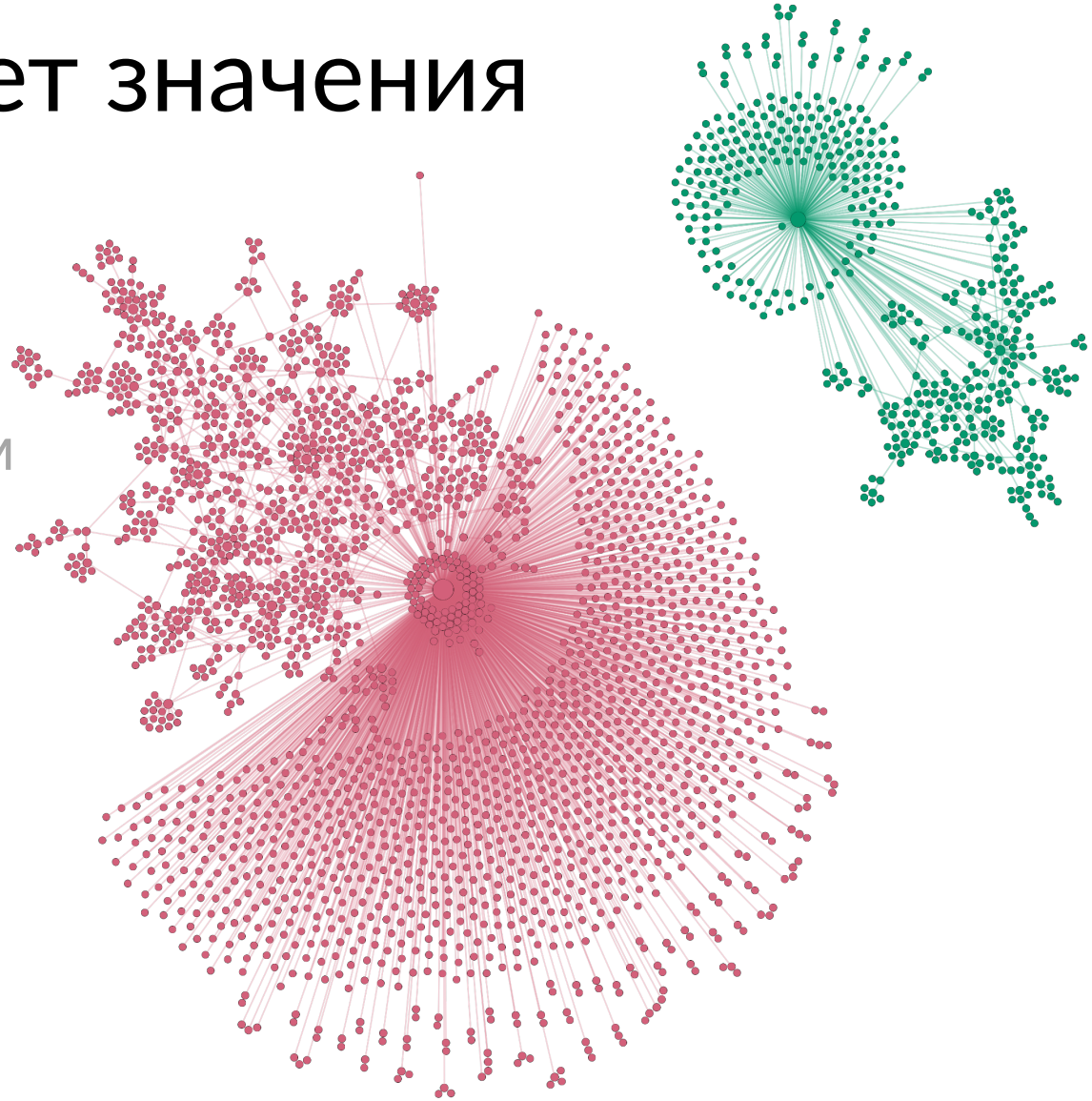
- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру



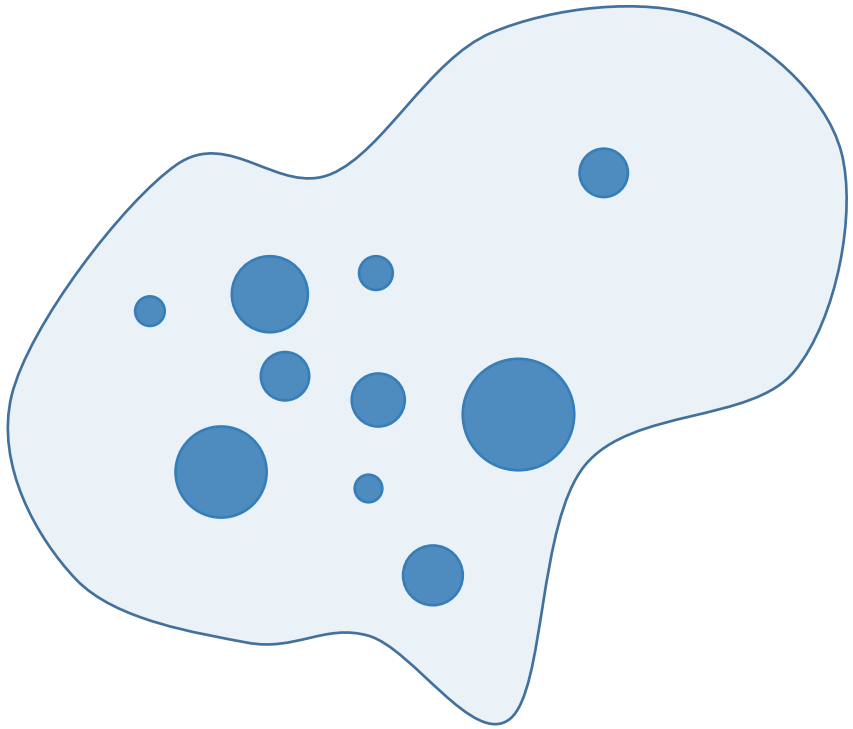
Иногда размер не имеет значения

3 ключевых свойства:

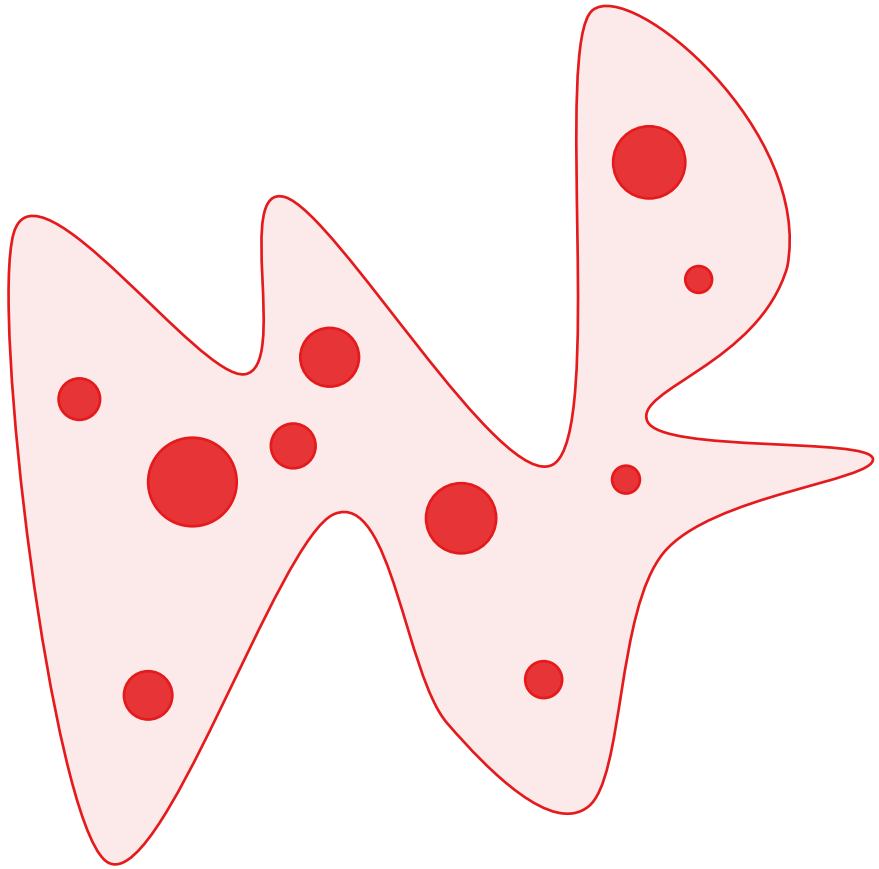
- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру



Расстояние Громова–Васерштейна

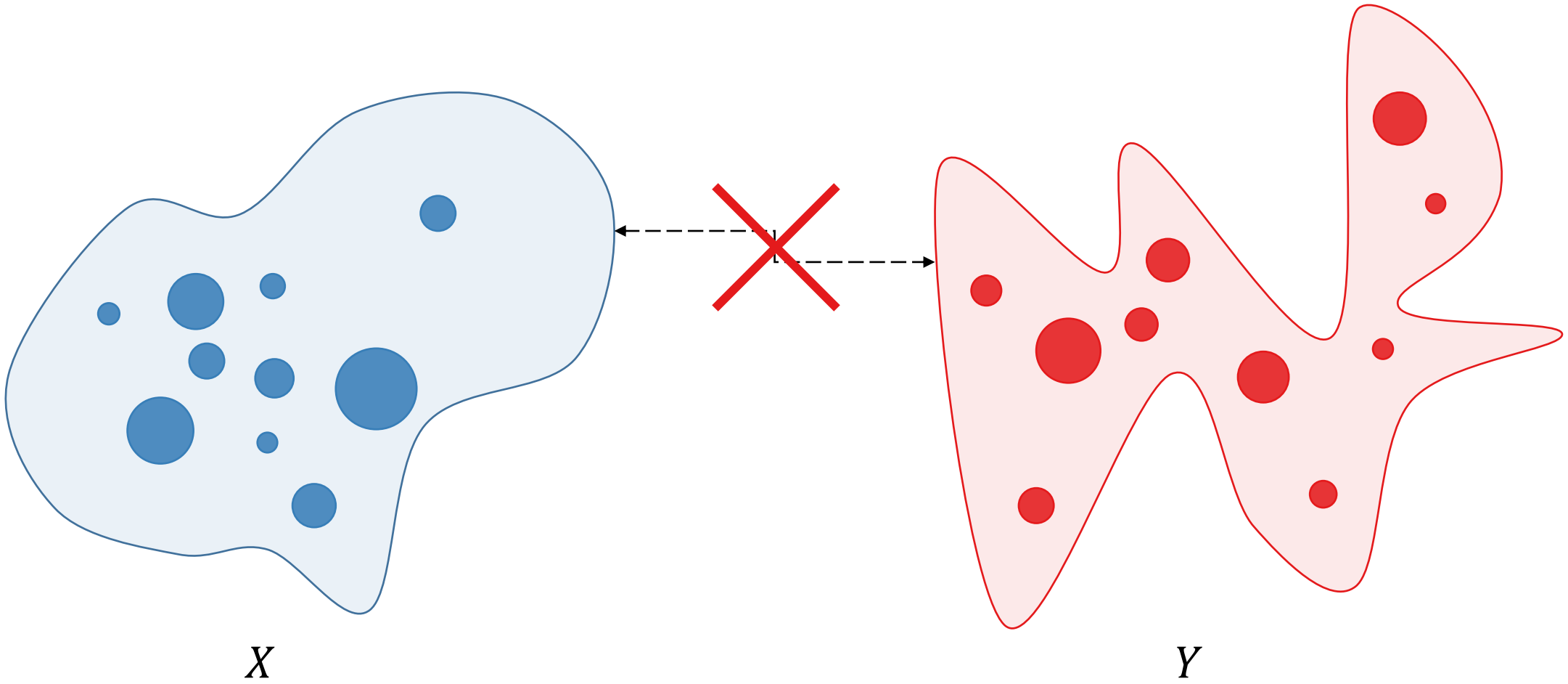


X

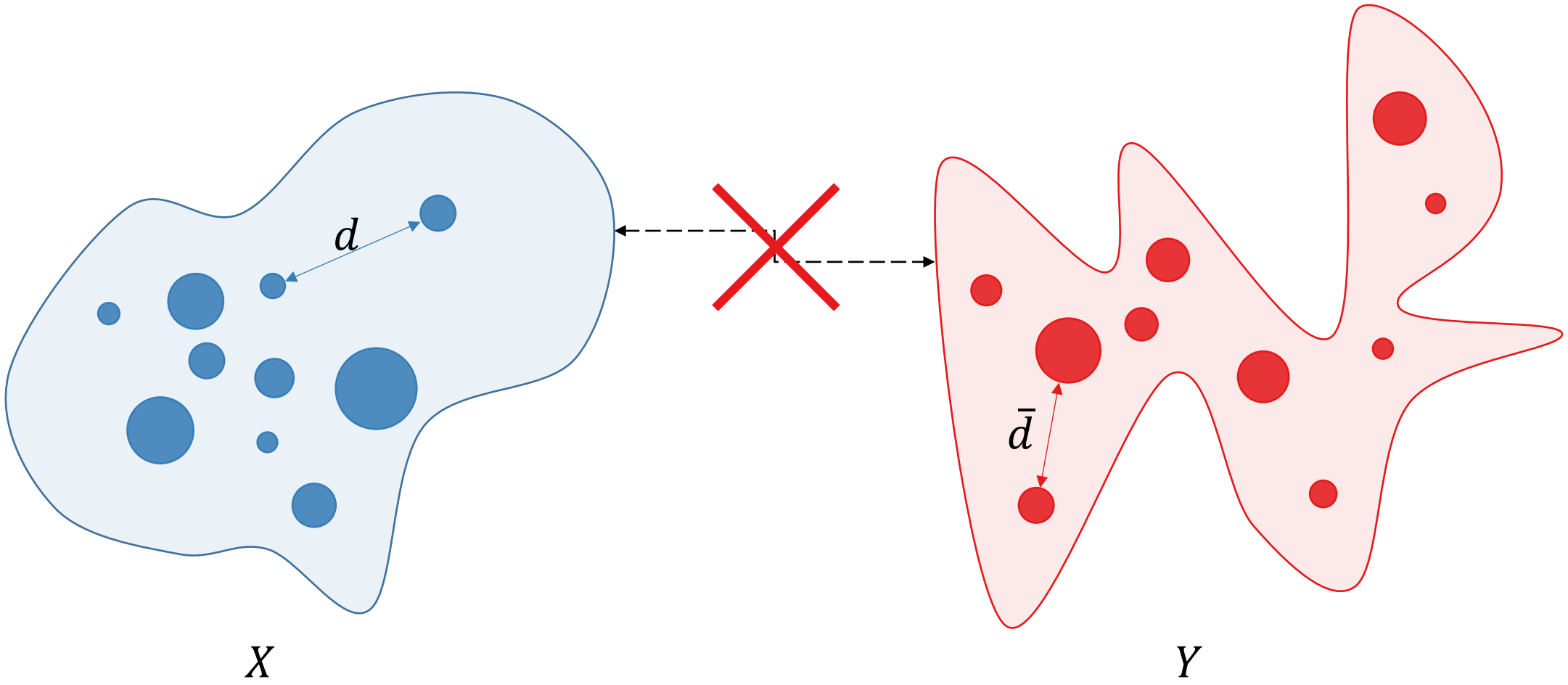


Y

Расстояние Громова–Васерштейна

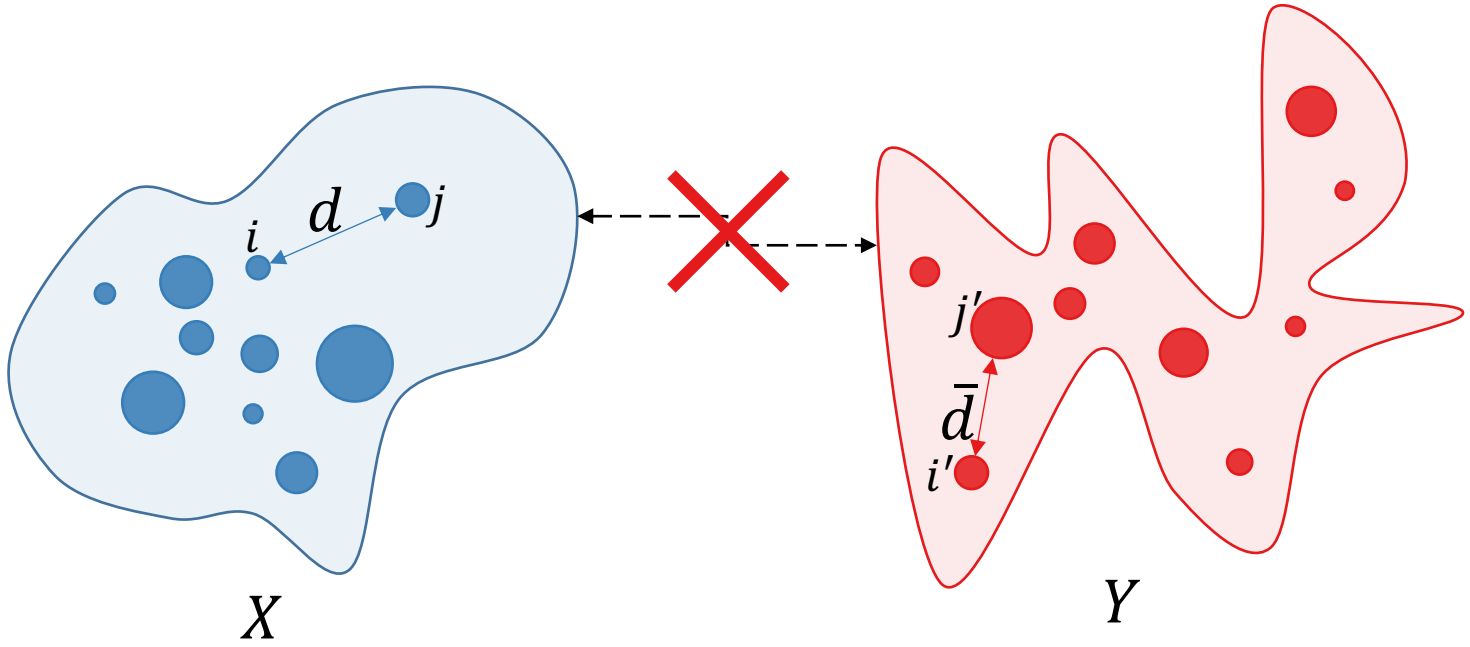


Расстояние Громова–Васерштейна



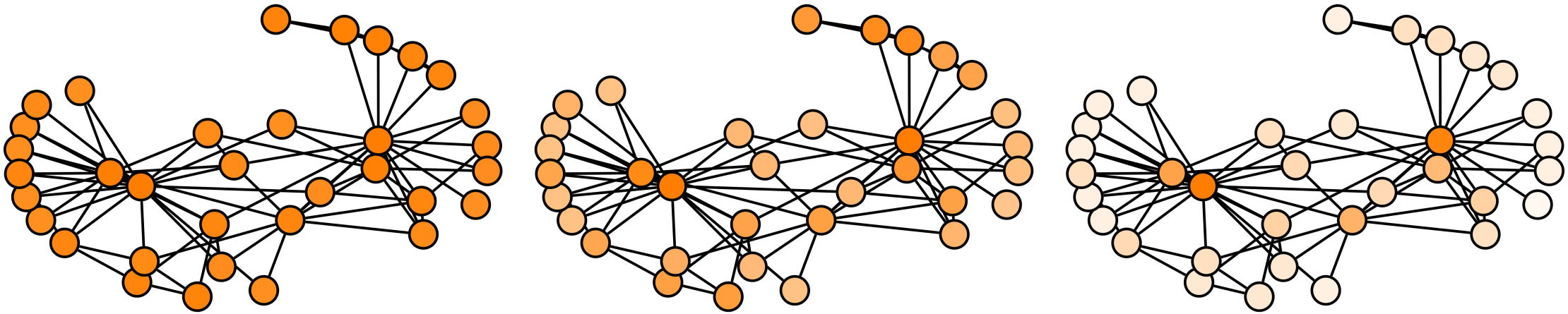
Расстояние Громова–Васерштейна

$$d_{\mathcal{GW},p}(X,Y) = \frac{1}{2}\left(\inf_{M}\sum_{i,j}\sum_{i',j'}\left|d(x_i,x_{i'}) - \bar{d}(y_j,y_{j'})\right|^p m_{ij}m_{i'j'}\right)^{1/p}$$



В уравнении теплопроводности явно определён масштаб

$$\frac{\partial u_t}{\partial t} = -\mathcal{L}u_t$$

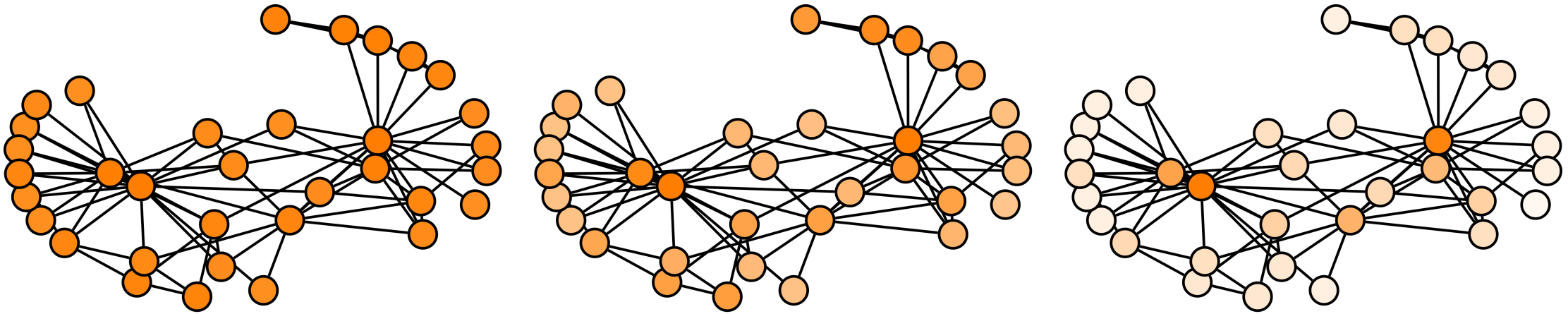


Small t Medium t Large t

В уравнении теплопроводности явно определён масштаб

$$H_t = e^{-t\mathcal{L}} = \Phi e^{-t\Lambda}\Phi^\top = \sum_{j=1}^{n} e^{-t\lambda_j}\phi_j\,\phi_j^\top$$

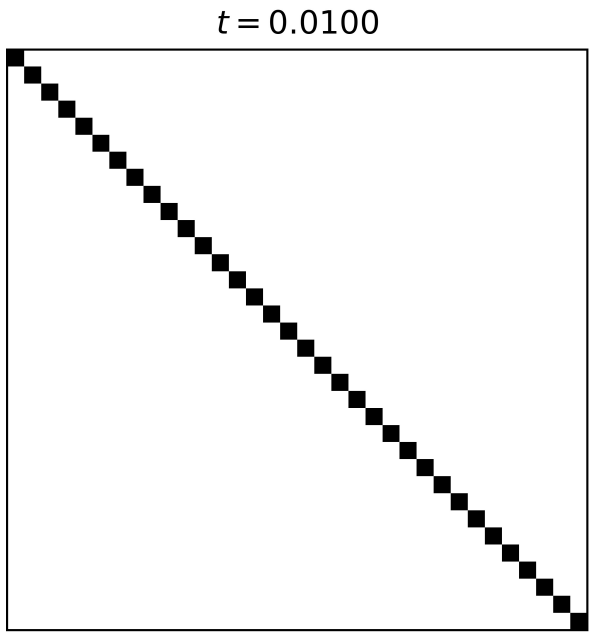
Матрица решений уравнения теплопроводности = heat kernel

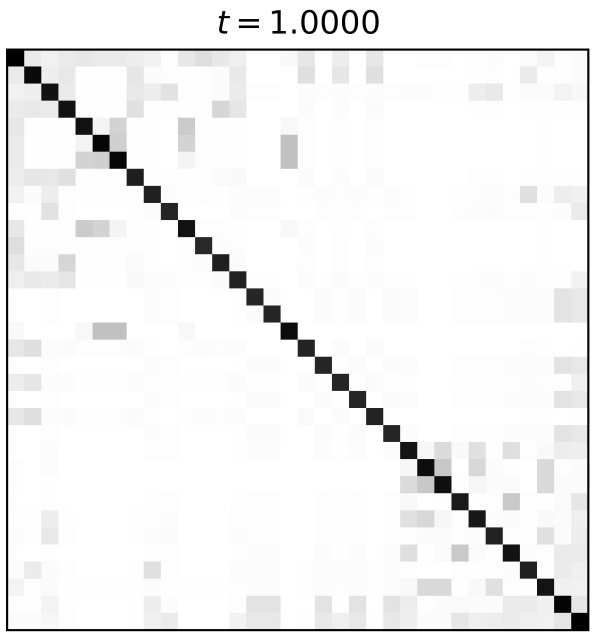


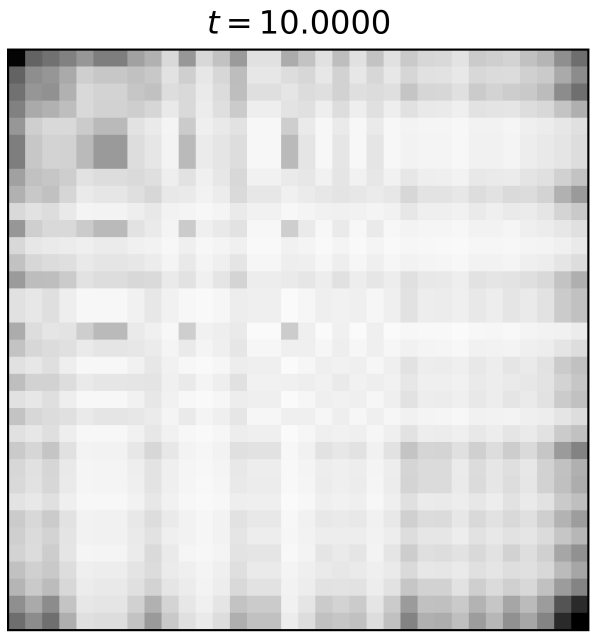
Small t Medium t Large t

Время определяет локальность оператора

$$H_t = e^{-t\mathcal{L}} = \Phi e^{-t\Lambda}\Phi^\top = \sum_{j=1}^{n} e^{-t\lambda_j}\phi_j\,\phi_j^\top$$

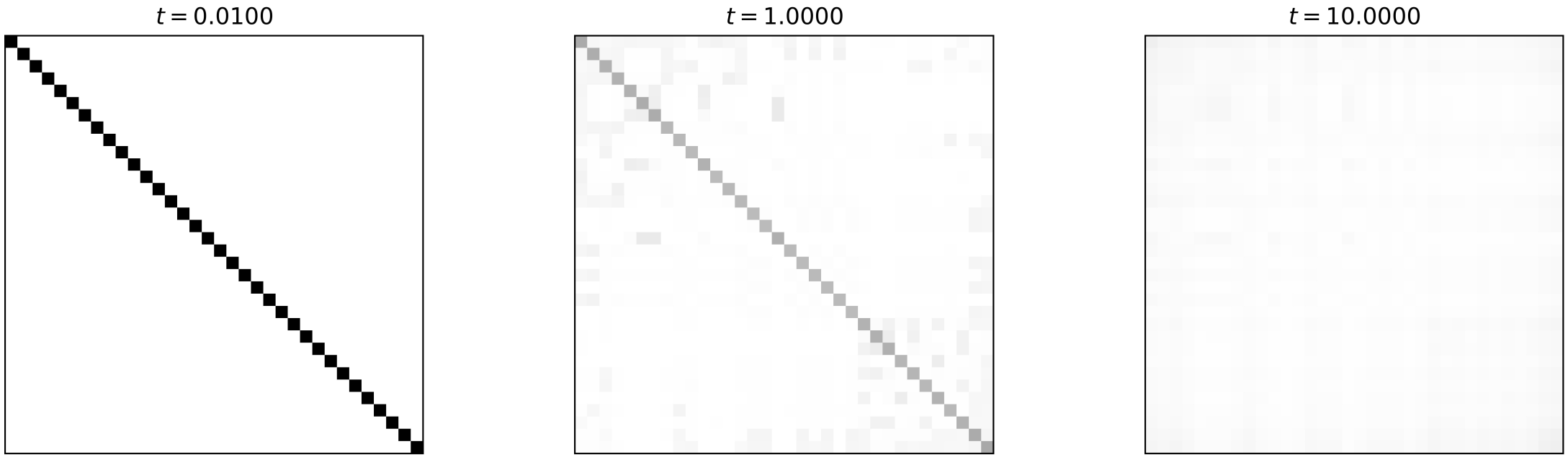






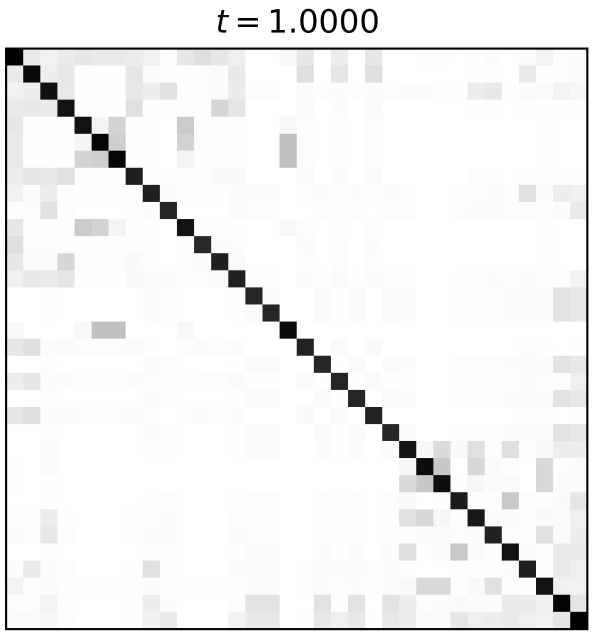
Время определяет локальность оператора

$$H_t = e^{-t\mathcal{L}} = \Phi e^{-t\Lambda}\Phi^\top = \sum_{j=1}^{n} e^{-t\lambda_j}\phi_j\,\phi_j^\top$$



Спектральный Громов–Васерштейн = Громов–Васерштейн + heat kernel

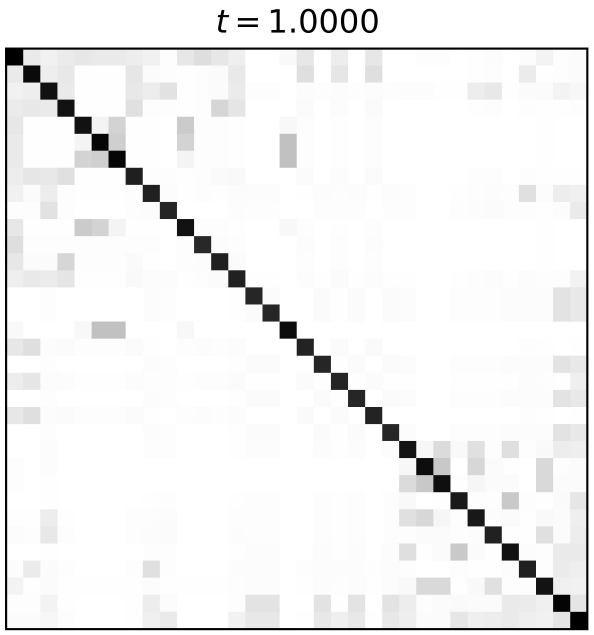
$$d^{\mathrm{spec}}_{\mathcal{GW},p}(X,Y) = \inf_{M} \sup_{t>0} e^{-2\left(t+t^{-1}\right)} \cdot \left(\sum_{i,j} \sum_{i',j'} \left| H^X_t(x_i, x_{i'}) - H^Y_t(y_j, y_{j'}) \right|^p m_{ij} m_{i'j'} \right)^{1/p}$$



Heat kernel на всех t не помогает нам решить задачу о расстоянии между графами

У спектрального Громова–Васерштейна есть простая оценка снизу!

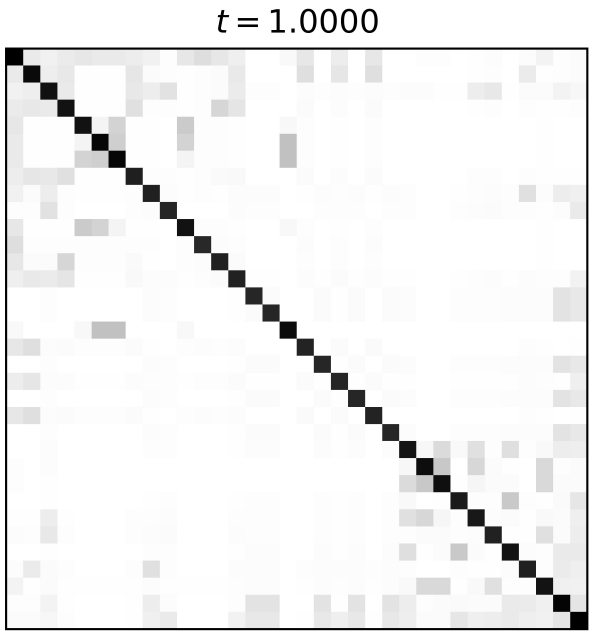
$$d_{\mathcal{GW},p}^{\mathrm{spec}}(X,Y) = \inf_{M} \sup_{t>0} e^{-2\left(t+t^{-1}\right)} \cdot \left(\sum_{i,j} \sum_{i',j'} \left| H_t^X(x_i, x_{i'}) - H_t^Y(y_j, y_{j'}) \right|^p m_{ij} m_{i'j'} \right)^{1/p} \geq \sup_{t>0} e^{-2\left(t+t^{-1}\right)} \cdot \left| \operatorname{tr}(H^X) - \operatorname{tr}(H^Y) \right|$$



Оценка снизу позволяет оценить разницу в heat kernel’ах на всех t

У спектрального Громова–Васерштейна есть простая оценка снизу!

$$d_{\mathcal{GW},p}^{\mathrm{spec}}(X,Y) = \inf_{M} \sup_{t>0} e^{-2\left(t+t^{-1}\right)} \cdot \left(\sum_{i,j} \sum_{i',j'} \left| H_t^X(x_i, x_{i'}) - H_t^Y(y_j, y_{j'}) \right|^p m_{ij} m_{i'j'} \right)^{1/p} \geq \sup_{t>0} e^{-2\left(t+t^{-1}\right)} \cdot \left| \operatorname{tr}(H^X) - \operatorname{tr}(H^Y) \right|$$



Оценка снизу позволяет оценить разницу в heat kernel’ах на всех t

Нужно просто посчитать след!

Network Laplacian Spectral Descriptors

$$h_t = \mathrm{tr}(H_t) = \sum_j e^{-t\lambda_j}$$

Логарифмически сэмплируем t, и сравниваем h_t через L_2
h_t зависит от размера!

Инвариантность к размеру=нормализация

$$h_t = \mathrm{tr}(H_t) = \sum_j e^{-t\lambda_j}$$

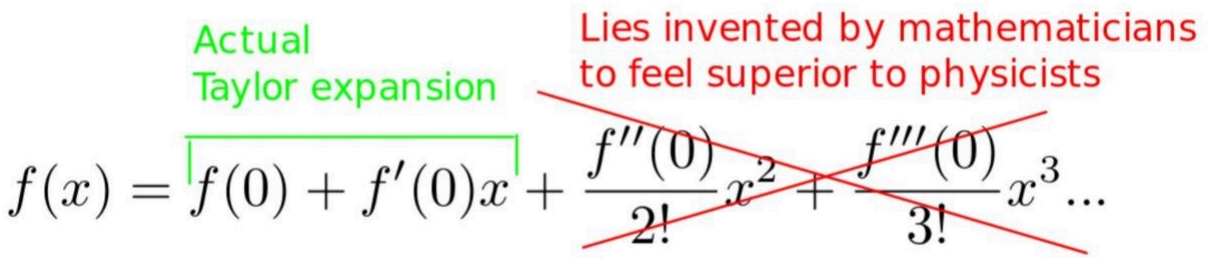
Нормализуем на h_t полного (K) или нуль-графа $\overline{K}$
Подсчёт всех λ – дорогое удовольствие сложностью $O(n^3)$

Масштабируемость

Предлагаем три варианта:

1. Используя ряд Тейлора: $h_t = \mathrm{tr}(e^{-t\mathcal{L}}) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}((-t\mathcal{L})^k)}{k!} \approx n - t\ \mathrm{tr}(\mathcal{L}) + \frac{t^2}{2}\mathrm{tr}(\mathcal{L}^2) + \dots$

   Второе слагаемое – распределение степеней; третье – кол-во треугольников



Масштабируемость

Предлагаем три варианта:

1. Используя ряд Тейлора: $h_t = \mathrm{tr}(e^{-t\mathcal{L}}) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}((-t\mathcal{L})^k)}{k!} \approx n - t\ \mathrm{tr}(\mathcal{L}) + \frac{t^2}{2}\mathrm{tr}(\mathcal{L}^2) + \dots$

   Второе слагаемое – распределение степеней; третье – кол-во треугольников

2. Считать крайние с.з., аппроксимировать остальные

   Линейная экстраполяция – ограничение на размерность (Weyl's law)

Масштабируемость

Предлагаем три варианта:

1. Используя ряд Тейлора: $h_t = \mathrm{tr}(e^{-t\mathcal{L}}) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}((-t\mathcal{L})^k)}{k!} \approx n - t\ \mathrm{tr}(\mathcal{L}) + \frac{t^2}{2}\mathrm{tr}(\mathcal{L}^2) + \dots$

   Второе слагаемое – распределение степеней; третье – кол-во треугольников

2. Считать крайние с.з., аппроксимировать остальные

   Линейная экстраполяция – ограничение на размерность (Weyl's law)

3. **Стохастическая оценка следа**

   Оценка следа Хатчинсона + алгоритм Ланцоша + метод Гаусса

Стохастическая оценка следа

Stochastic Lanczos Quadrature

$$\sum_i f(\lambda_i(M)) \approx \texttt{slq}(M, f)$$

Оценка следа Хатчинсона

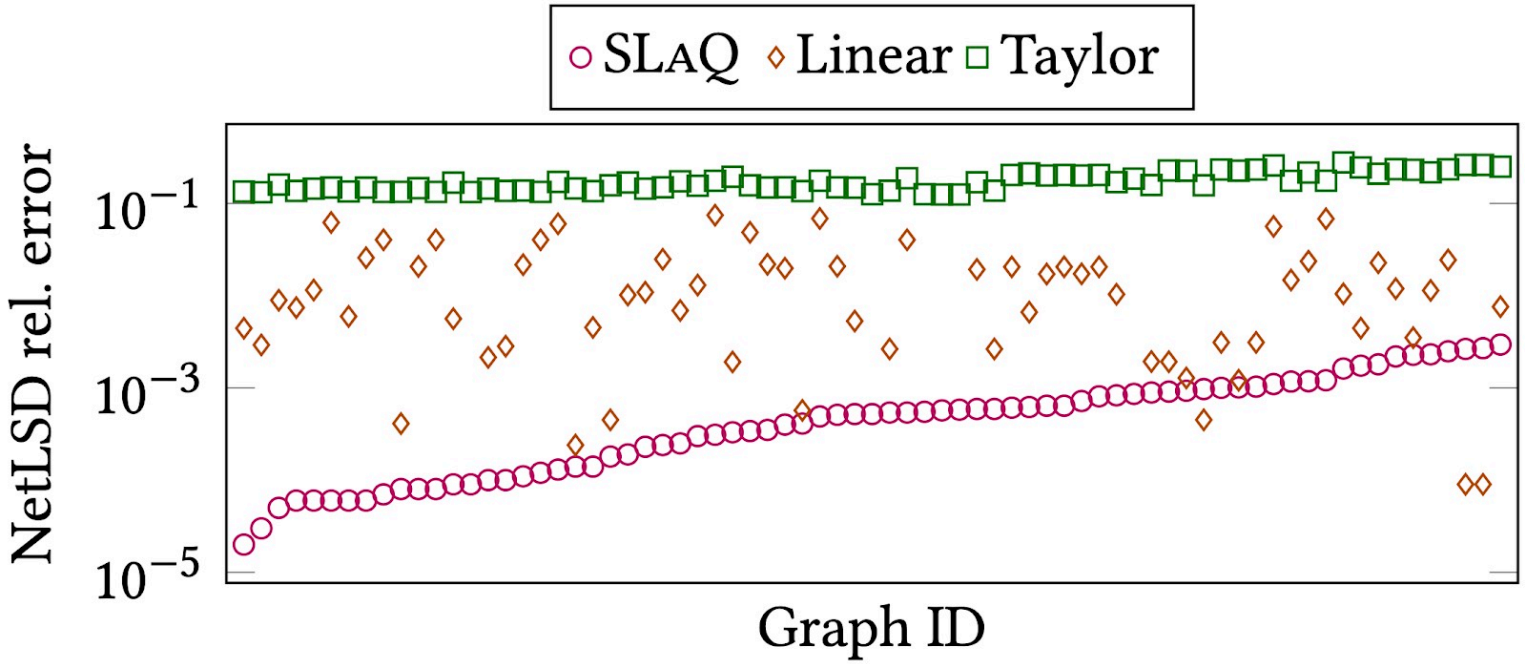
$$z^T f(M) z \approx \sum_{k=1}^{s} w_k f(x_k)$$

Метод Гаусса

$$\texttt{Tr} f(M) = \mathbb{E}[z^T f(M) z]$$

Алгоритм Ланцоша: s весов w_k в точках x_k

Качество аппроксимации



SLaQ уменьшает ошибку относительно интерполяции в 20-30x и в 200-250x относительно разложений Тейлора!

Детекция графов с сообществами

3 ключевых свойства:

- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру

NetLSD

NIPS'17
ASONAM'13

	$n \sim \mathcal{P}(\lambda)$				
Method	64	128	256	512	1024
$h(G)$	54.39	59.01	60.82	57.99	53.80
$h(G)/h(\bar{K})$	54.53	62.27	70.83	76.45	78.40
$w(G)$	56.23	63.77	69.57	71.66	70.34
$w(G)/w(\bar{K})$	55.51	63.85	72.12	77.59	79.39
FGSD	55.44	54.99	53.86	52.74	50.92
NetSimile	59.55	56.57	59.41	66.23	60.58

Точность классификации графов SBM против Эрдёша–Реньи

Детекция перемешанных графов

3 ключевых свойства:

- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру

NetLSD

NIPS'17
ASONAM'13

	dataset					
Method	MUTAG	PROTEINS	NCI1	ENZYMES	COLLAB	IMDB-B
$h(G)$	76.03	91.81	69.74	92.51	59.82	67.18
$h(G)/h(\bar{K})$	79.12	94.90	74.55	95.20	65.85	70.58
$w(G)$	78.18	93.04	70.54	94.03	69.01	75.26
$w(G)/w(\bar{K})$	79.72	89.00	74.14	90.77	70.35	75.54
FGSD	77.79	60.11	64.08	53.93	55.18	56.23
NetSimile	77.11	85.73	58.58	87.38	54.43	54.44

Точность классификации

Классификация на реальных данных

3 ключевых свойства:

- Инвариантность к перестановкам
- Адаптируемость к масштабу
- Инвариантность к размеру

NetLSD

NIPS'17
ASONAM'13

	dataset					
Method	MUTAG	PROTEINS	NCI1	ENZYMES	COLLAB	IMDB-B
$h(G)$	86.47	64.89	66.49	31.99	68.00	68.04
$h(G)/h(\bar{K})$	85.32	65.73	67.44	33.31	69.42	70.17
$w(G)$	83.35	66.80	70.78	40.41	75.77	68.63
$w(G)/w(\bar{K})$	81.72	65.58	67.67	35.78	77.24	69.33
FGSD	84.90	65.30	75.77	41.58	73.96	69.54
NetSimile	84.09	62.45	66.56	33.23	73.10	69.20

Точность классификации

А что с диплёрнингом?

DDGK: неглубокое обучение

1. Кодируем каждый граф при помощи нейросети
   Добавляем лосс на реконструкцию матрицы смежности
2. Замораживаем веса кодировщика
3. Считаем реконструкцию каждого графа из закодированных эмбеддингов другого графа
4. Хорошая реконструкция = графы близки

DDGK: неглубокое обучение

💔 Не end-to-end

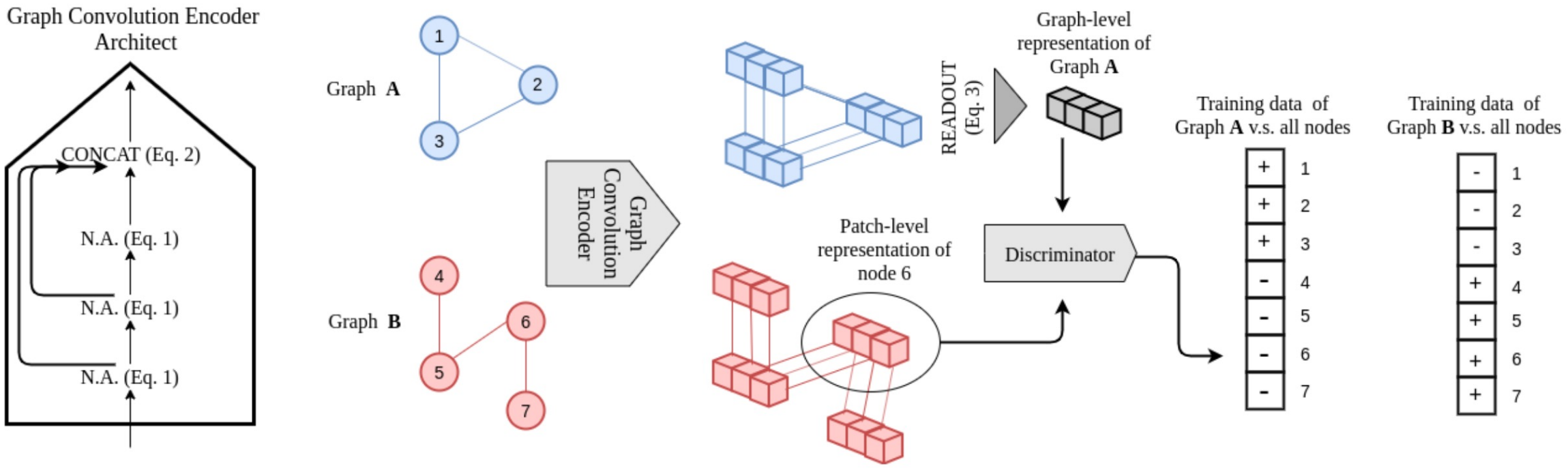
💔 Медленное обучение

❤️ Нормальное качество

InfoGraph: инфомакс для графов

1. Кодируем каждый граф при помощи нейросети
   Берём представления вершин и агрегированное для графа
2. Попарно пытаемся понять, принадлежат ли представления агрегату одного или другого графа

InfoGraph: инфомакс для графов



InfoGraph: инфомакс для графов

❤️ Первый end-to-end метод

💔 Медленное обучение

❤️ Хорошее качество

Сравнение методов

Метод	Без атрибутов	С атрибутами	Скорость	Качество
WL	✓	✓	✓	✓
NetSimile	✓	✗	✓	±
NetLSD	✓	✗	✓✓	±
DDGK	✓	✓	✗	±
InfoGraph	✗	✓	✗	✓

Сравнение типовметодов

Метод	Без атрибутов	С атрибутами	Скорость	Качество
Ядра	✓	✓	✓	✓
Статистики	✓	✗	✓	±
Спектральные	✓	✗	✓✓	±
Диплёрнинг	✗	✓	✗	✓

Вопросики?

код	github.com/xgfs/netlsd + graph_embedding/slaq
сайт	tsitsul.in/talks/sberloga-graphs
мейл	anton@tsitsul.in