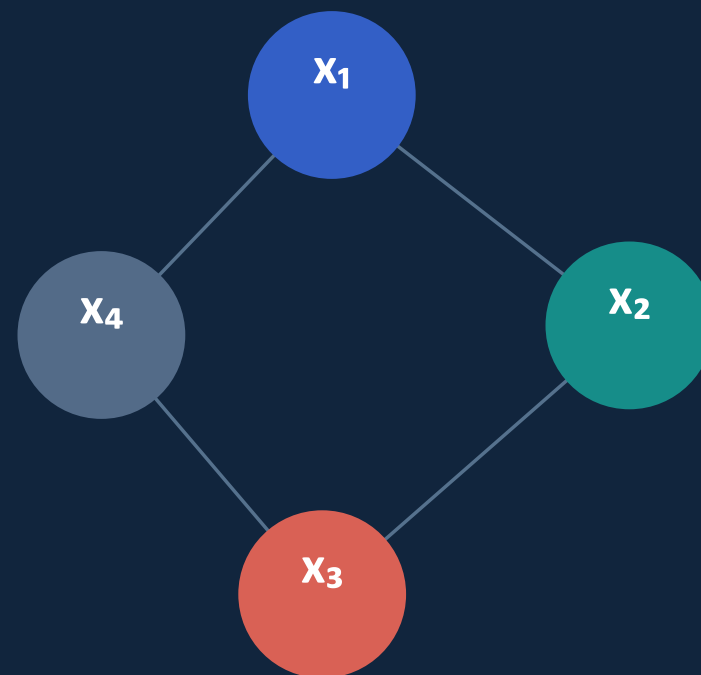


Моделирование динамики мнений по цифровым следам

Онлайн-сообщества
и программные репозитории

Хэ Юйлун | Yulong He

Санкт-Петербургский государственный университет



Состояния → зависимости

Хэ Юйлун | Yulong He



ОБРАЗОВАНИЕ

СПбГУ · 2024–2027

Аспирантура: теоретическая информатика и кибернетика

Магистратура · 2022–2024
Бакалавриат · 2018–2022

Прикладная математика и информатика, СПбГУ

ОПЫТ И ИССЛЕДОВАНИЯ

Huawei · с 2021

Инженер по машинному обучению и алгоритмам, Санкт-Петербург

ITMO AI Lab · 2025–2026
Разработчик, по совместительству

Кодовые представления · поиск кода
RAG · интерпретируемая динамика

Аспирант СПбГУ

Модели · данные · код

Связующее направление: численные представления цифровой активности и их интерпретация.

Как возникла динамика мнений?

Как общение меняет суждения людей?

От психологии групп — к математическому описанию согласия и разногласий.

1950-е

Социальная психология

Эксперименты: давление группы может менять высказываемые суждения.

[Аш \(1955\)](#)

1956–1974

Сетевые модели влияния

Люди связаны в сеть; мнения обновляются с учётом мнений других.

[Френч \(1956\)](#) · [ДеГроот \(1974\)](#)

1990

Не только консенсус

Привязка к исходной позиции помогает описать устойчивые разногласия.

[Фридкин–Джонсен \(1990\)](#)

В этом докладе — проверка моделей и оценка зависимостей по цифровым следам.

От вопроса «что думают люди?» — к вопросу «как меняются их мнения?»

Что такое динамика мнений?

Как меняются состояния участников под влиянием друг друга?

$$x_i(t+1) = \sum_{j=1}^n w_{ij} x_j(t)$$

$x_i(t)$ — состояние участника i

w_{ij} — вес источника j при обновлении i

В разных моделях возможны согласие, устойчивые различия и разделение на группы.

В этом докладе мы решаем обратную задачу: по траекториям оцениваем матрицу зависимостей.

УЧЕБНЫЙ ПРИМЕР

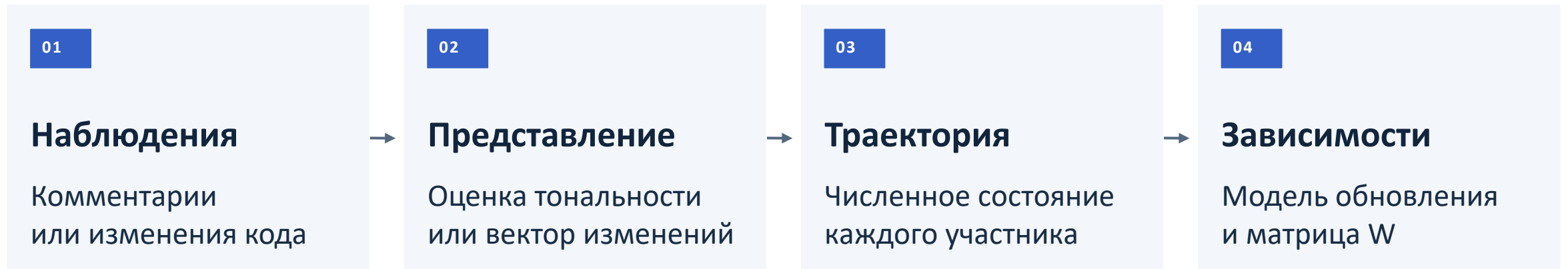
0,2 0,6 0,9

веса: 0,5 0,3 0,2

$0,5 \times 0,2 + 0,3 \times 0,6 + 0,2 \times 0,9$

= 0,46

Сначала — состояние, затем — динамика



WEIBO

Состояние = тональность
комментирующей аудитории

He, Y., Proskurnikov, A.V. & Sedakov, A. Mapping sentiment dependencies among online influencer communities on Weibo. *Social Network Analysis and Mining* (2026).
<https://doi.org/10.1007/s13278-026-01645-w>

ПРОГРАММНЫЙ КОД

Состояние = координата
семантического изменения кода

He, Y., Verbin, N., Kovalchuk, S., Social life of code: Modeling evolution through code embedding and opinion dynamics, *Journal of Computational Science* (2026) <https://doi.org/10.1016/j.jocs.2026.102824>.

Общая последовательность анализа сохраняется; смысл состояния определяется предметной областью.

WEIBO

Состояние = тональность
комментирующей аудитории



ПРОГРАММНЫЙ КОД

Состояние = координата
семантического изменения кода

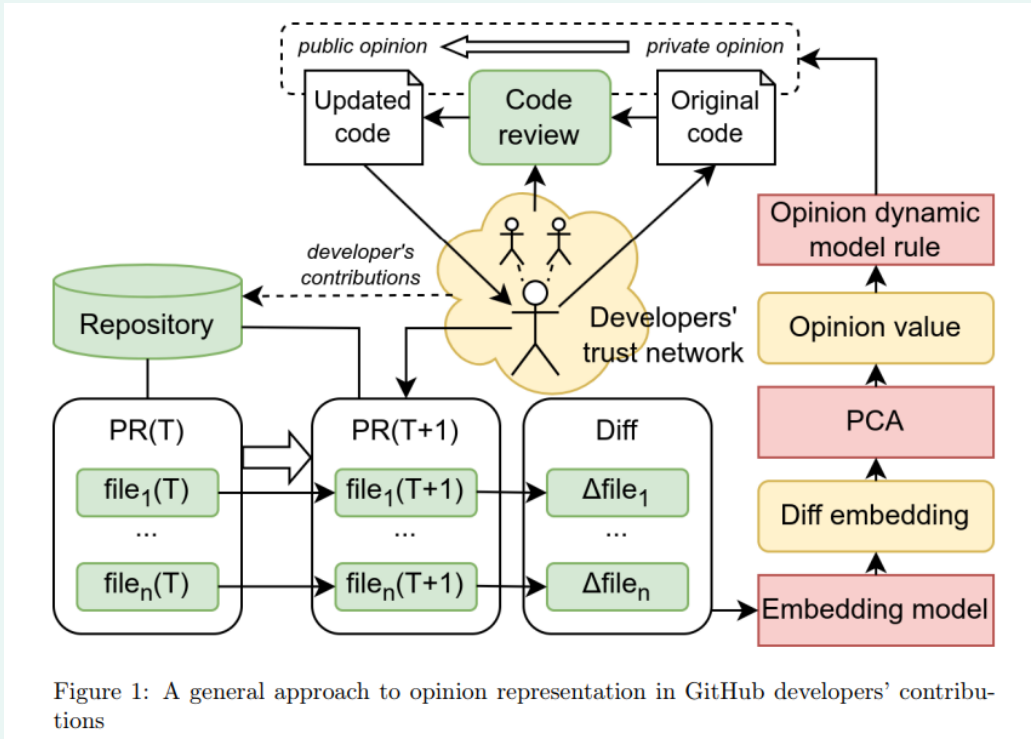


Figure 1: A general approach to opinion representation in GitHub developers' contributions

Общая последовательность анализа сохраняется; смысл состояния определяется предметной областью.

Зачем изучать динамику мнений?

ОНЛАЙН-СООБЩЕСТВА · WEIBO

Обсуждение нового смартфона

В одном блоге его поддерживают,
в другом — критикуют.

**Сближаются ли реакции аудиторий
или различия сохраняются?**

ПРОГРАММНЫЕ РЕПОЗИТОРИИ · GITHUB

Совместная разработка

Участник меняет обработку ошибок;
похожие правки появляются у коллег.

**Связаны ли изменения кода
у разных разработчиков?**

ЧТО ДАЁТ МОДЕЛЬ

ОПИСАТЬ

Как меняются состояния?

ОЦЕНИТЬ СВЯЗИ

Какие траектории связаны?

ПРОГНОЗИРОВАТЬ

Что изменится дальше?

От отдельных комментариев и правок — к модели совместных изменений.

Weibo: агент — комментирующая аудитория

200 → 7

От тематического рейтинга
к отобранным блогам

12

Полумесячных периодов
16.05–15.11.2023

7 × 12

Аудиторий × периодов
Матрица наблюдений

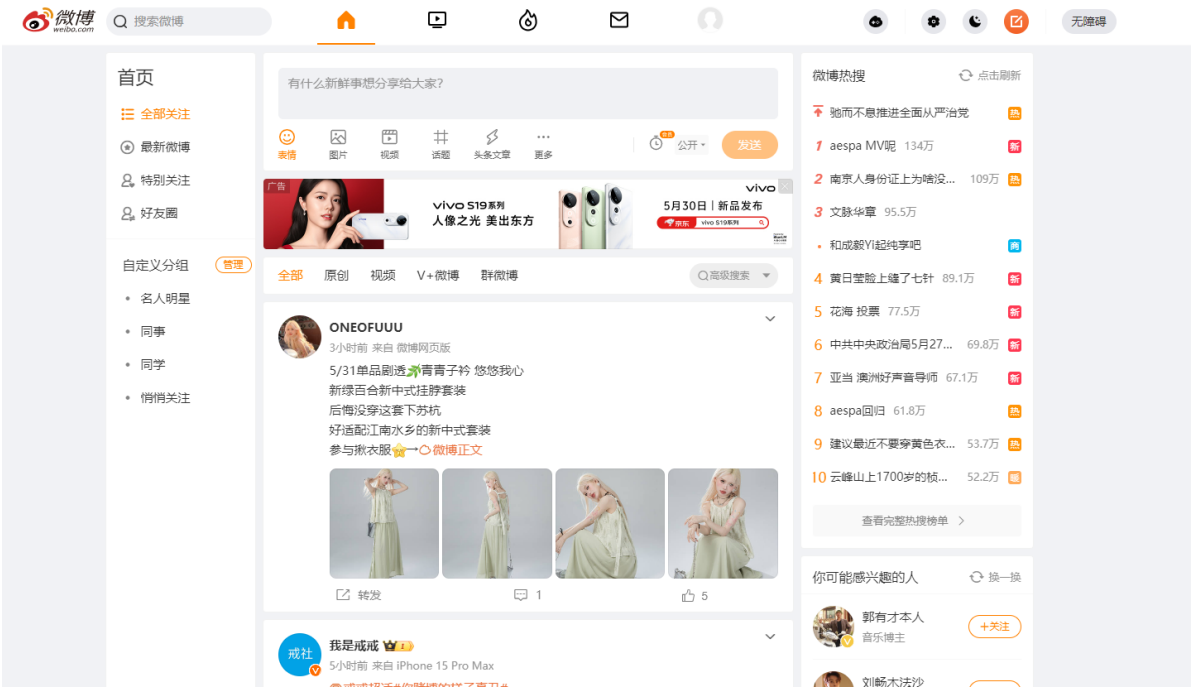
ОТБОР НАБЛЮДЕНИЙ

Технологическая тематика и публикации о Huawei.
Активные верифицированные блоги с >1 млн читателей.
Только комментарии первого уровня.

**Мы измеряем реакцию тех, кто
комментирует, а не мнение
блогера.**

Состав аудитории может меняться; единица анализа задаётся общим потоком контента.

Dataset



user_name	user_id	Number_of_followers
科技小辛(Technology Xiaoxin)	1595443924	1.611 million
李杰灵(Li Jieling)	1783497251	2.982 million
小白测评(Novice evaluation)	2022252207	3.47 million
数码疯报(Digital Crazy News)	2561744167	2.656 million
定焦数码(fixed focus digital)	5821279480	1.072 million
勇气数码君(Courage Digital Master)	7109370363	1.708 million
白问视频(Baiwen video)	7239083016	1.105 million

Table 2: Self-media bloggers on sina

1. fans > 1,000,000,
2. every 15 days will publish >=1 post of theme 'huawei'

comment_likes change	0.1447%
comment remove	4.5563%
post_likes_change	0.0194%
post_remove	0%

Table 1: Change

We think that Sina Weibo, twitter like this type of platforms are **Quick forgetting**.

Анализ тональности: обучение не равно применению

ОБУЧЕНИЕ

Размеченные тексты



bert-base-chinese



Настроенный
классификатор

ПРОВЕРКА

Отдельный Weibo-тест
2018



Предсказания классов



Precision · Recall · F1 ·
AUC

ПРИМЕНЕНИЕ

Комментарии из
выборки 2023



Готовый классификатор



Оценки $\sigma(t) \in [0, 1]$

Для обучения сравниваются отзывы об отелях и корпус weibo_senti_100k.

Выход — численная оценка тональности текста, а не непосредственное измерение эмоции человека.

Почему выбран BERT, дообученный на Weibo?

Модель / корпус обучения	Precision	Recall	F1	Accuracy	AUC
Naive Bayes / Weibo	0,7444	0,8609	0,7984	0,7000	0,6044
BiLSTM / Weibo	0,8950	0,6174	0,7307	0,6860	0,8071
BERT / отзывы об отелях	0,8754	0,8145	0,8438	0,7920	0,8672
BERT / Weibo	0,8531	0,9594	0,9031	0,8580	0,9469

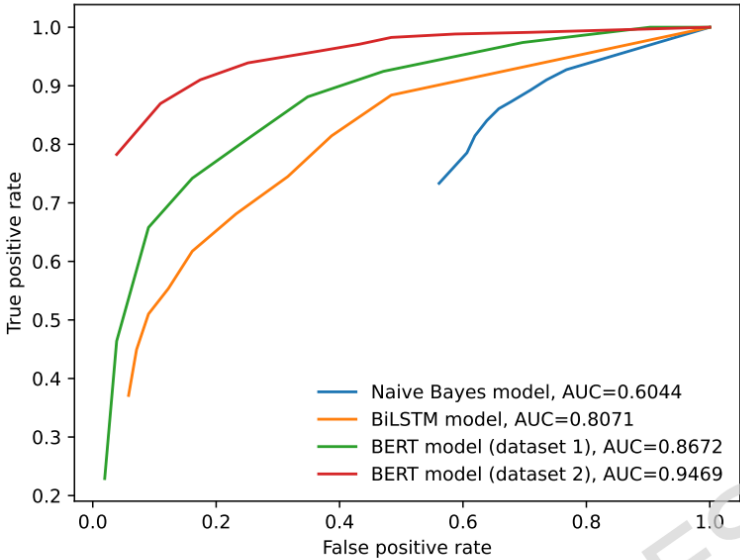


Fig. A2: Sentiment analysis algorithms: ROC curves.

Лучшая F1: 0,9031

Лучшие Recall, F1, Accuracy и AUC.
Но Precision выше у BiLSTM.

Домен обучения важен: тексты платформы дают более подходящий измерительный инструмент.

От комментария к состоянию сообщества

СНАЧАЛА — СЕНТИМЕНТ ПУБЛИКАЦИИ

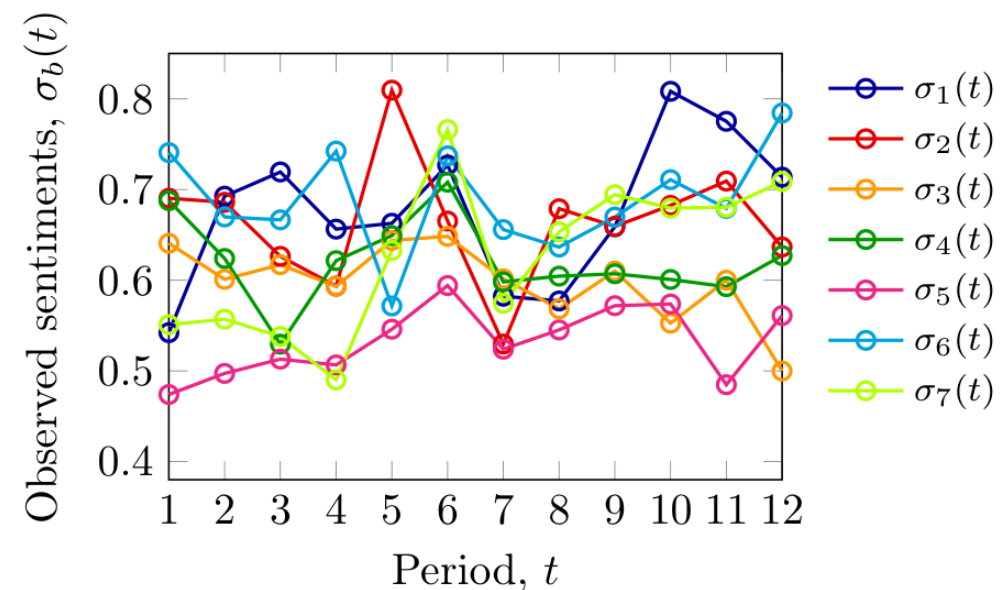
$$\sigma_p(t) = \frac{\sum_{c \in C_{p,b}(t)} \lambda_c(t) \sigma_c(t)}{\sum_{c \in C_{p,b}(t)} \lambda_c(t)}$$

Вес λ_c — число лайков комментария.

ЗАТЕМ — СЕНТИМЕНТ АУДИТОРИИ ЗА ПЕРИОД

$$\sigma_b(t) = \frac{\sum_{p \in P_b(t)} \lambda_p(t) \sigma_p(t)}{\sum_{p \in P_b(t)} \lambda_p(t)}$$

Вес λ_p — число лайков публикации.



Наблюдаемые траектории: $x^e_b(t) = \sigma_b(t)$

Два уровня весов различаются: лайки комментариев → лайки публикаций → состояние аудитории.

От усреднения к памяти о собственной позиции

FRENCH-DEGROOT

$$\mathbf{x}(t + 1) = W\mathbf{x}(t)$$

$W \geq 0$, сумма каждой строки равна 1.
Вся динамика задаётся усреднением.

ФРИДКИН-ДЖОНСЕН

$$\mathbf{x}(t + 1) = SW\mathbf{x}(t) + (I - S)\mathbf{z}$$

s_i — восприимчивость к другим.
 $1 - s_i$ — привязка к базовому состоянию z_i .

FDGM: КОНЕЧНАЯ ПАМЯТЬ ВМЕСТО ПОСТОЯННОГО БАЗОВОГО СОСТОЯНИЯ

$$\mathbf{x}(t + 1) = SW\mathbf{x}(t) + (I - S)\mathbf{x}(t - \tau)$$

Направление коэффициента: w_{ij} соответствует $j \rightarrow i$. В Weibo вектор $\mathbf{z}$ оценивается, а не наблюдается.

ЕРО: наблюдаемое выражение и скрытая позиция

1. ОБНОВЛЯЕТСЯ ЧАСТНОЕ СОСТОЯНИЕ

$$\mathbf{x}(t+1) = S[\Delta\mathbf{x}(t) + (I - \Delta)A\mathbf{x}^e(t)] + (I - S)\mathbf{z}$$

2. ФОРМИРУЕТСЯ ВЫРАЖЕННОЕ СОСТОЯНИЕ

$$\mathbf{x}^e(t+1) = \Phi\mathbf{x}(t+1) + (I - \Phi)A\mathbf{x}^e(t)$$

$\mathbf{x}(t)$ — скрытое частное состояние

$\mathbf{x}^e(t)$ — наблюдаемое выраженное состояние

Φ — вес частной позиции в выражении

$$W = \Delta + (I - \Delta)A$$

Δ — собственный вес; A — веса других

$\Phi = I$: ЕРО сводится к FJ. $S = I$: сокращённая ЕРО, используемая в исследовании кода.

Оценивание и прогноз — разные этапы

ОСНОВНАЯ ВРЕМЕННАЯ СХЕМА



Периоды 1–10: оценивание

11–12: прогноз

ПОДГОНКА

Параметры + скрытая траектория

Минимизируем остатки двух уравнений EPO при ограничениях на веса и состояния.

ПРОВЕРКА

Прогноз выраженных состояний

Сравниваем модель с отложенными наблюдениями по RMSE.

Остаток оптимизации, ошибка подгонки траектории и ошибка прогноза — не одно и то же.

Table 3: The opinion dynamics models under consideration (in experiments, we test lags $\tau = 1$ and $\tau = 2$).

Model Name	Dynamics and Variables	Parameters and Constraints
FDG model	$x(t+1) = Wx(t),$ $x_b = \sigma_b$: sentiment in blog $b \in \mathcal{B}$	W : row-stochastic matrix
FJ model	$x(t+1) = SWx(t) + (I - S)z,$ $x_b = \sigma_b$: sentiment in blog $b \in \mathcal{B}$	W : row-stochastic matrix, S : diagonal matrix with values in $[0, 1],$ z : constant vector with values in $[0, 1]$
EPO model	$x(t+1) = S(\text{diag}(W)x(t) + (W - \text{diag}(W))x^e(t))$ $+ (I - S)z,$ $x^e(t) = \Phi x(t) + (I - \Phi)Ax^e(t-1),$ $x_b^e = \sigma_b$: sentiment in blog $b \in \mathcal{B}$ x_b : private opinion in the blog (hidden)	W, A : nonnegative row-stochastic matrices, $\text{diag } A = 0$ S, Φ, D : diagonal matrices with values in $[0, 1]$ with $W = D + (I - D)A,$ z : constant vector with values in $[0, 1]$
Reduced EPO model ($S = I$)	$x(t+1) = \text{diag}(W)x(t) + (W - \text{diag}(W))x^e(t),$ $x^e(t) = \Phi x(t) + (I - \Phi)Ax^e(t-1),$ x_b^e, x_b : same as in the EPO model	W, A : nonnegative row-stochastic matrices, $\text{diag } A = 0,$ Φ, D : diagonal matrices with values in $[0, 1]$ with $W = D + (I - D)A$
FDGM model with lag τ	$x(t+1) = SWx(t) + (I - S)x(t-\tau),$ $x_b = \sigma_b$: sentiment in blog $b \in \mathcal{B}$	W : nonnegative row-stochastic matrix, S : diagonal matrix with values in $[0, 1]$
EPO model with lag τ	$x(t+1) = S(\text{diag}(W)x(t) + (W - \text{diag}(W))x^e(t))$ $+ (I - S)z,$ $x^e(t) = \Phi x(t) + (I - \Phi)Ax^e(t-\tau-1),$ x_b^e, x_b : same as in the EPO model	same as in the EPO model
Reduced EPO model with lag τ	$x(t+1) = \text{diag}(W)x(t) + (W - \text{diag}(W))x^e(t),$ $x^e(t) = \Phi x(t) + (I - \Phi)Ax^e(t-\tau-1),$ x_b^e, x_b : same as in the EPO model	same as in the reduced EPO model

Table 5: Performance of the opinion dynamics models (best ✓, competitive ★). Unconstrained VAR models are reported for reference in Section 5.5 and are excluded from the marking.

Model	Sum of residuals	MAE	MAPE	RMSE Periods 1–10 (in-sample)	RMSE Period 11 (out-of-sample)	RMSE Period 12 (out-of-sample)	RMSE Periods 11–12 (out-of-sample)
FDG model	0.2000	0.1308	7.7957%	0.1461	0.1463	0.1927	0.1711
FJ model	0.1647	0.1257	7.5226%	0.1401	0.1330 ★	0.1704	0.1529
EPO model	0.0773	0.1129	6.7539%	0.1319	0.1422	0.1563	0.1494
Reduced EPO model	0.0879	0.1161	6.9463%	0.1347	0.1346	0.1457 ★	0.1402 ✓
FDGM model with lag 1	0.1754	0.1219	7.2959%	0.1423	0.1496	0.1512	0.1504
FDGM model with lag 2	0.1412	0.0987 ★	5.8621% ★	0.1242 ★	0.1238 ★	0.1641	0.1454
EPO model with lag 1	0.0681 ★	0.1222	7.3173%	0.1411	0.1335	0.1559	0.1451
Reduced EPO model with lag 1	0.0783	0.1277	7.6704%	0.1434	0.1347	0.1476 ★	0.1413 ★
EPO model with lag 2	0.0530 ✓	0.0883 ★	5.2564% ★	0.1154 ★	0.1584	0.1340 ✓	0.1467
Reduced EPO model with lag 2	0.0639 ★	0.0854 ✓	5.0497% ✓	0.1114 ✓	0.1208 ✓	0.1641	0.1441 ★

Weibo: разреженная и асимметричная структура

СТОЛБЕЦ j — ИСТОЧНИК; СТРОКА i — ПОЛУЧАТЕЛЬ

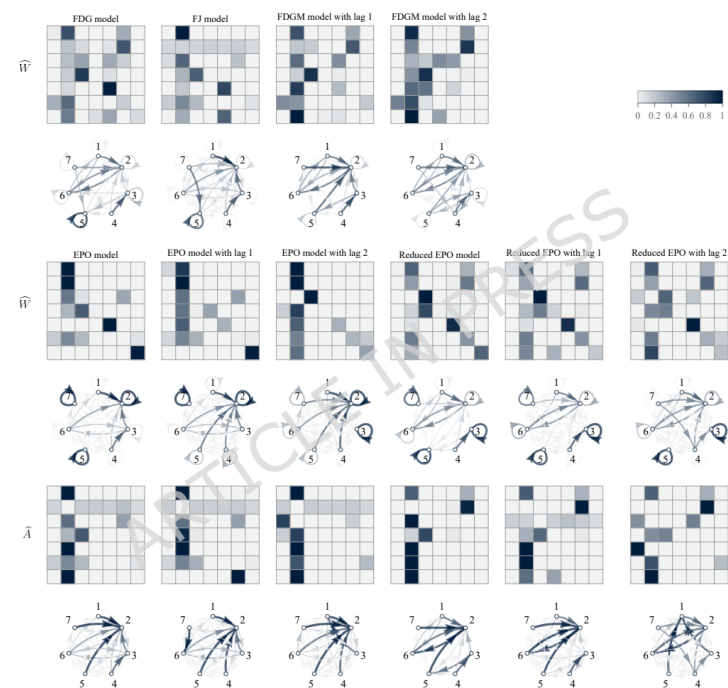


Fig. 4: Fitted matrices $\widehat{W}$ and $\widehat{A}$ as heatmaps and their associated graphs.

3 → 4

$$w_{43} = 0,7827$$

Самая сильная внешняя связь

2 ↔ 6

$$w_{26} = 0,5698$$

$$w_{62} = 0,5585$$

В модели различаются собственная устойчивость, внешняя зависимость и роль источника для других.

Код: от каталога PR к 21 активному разработчику

ИСХОДНЫЙ РЕСУРС ДЛЯ ОТБОРА

«88.6 Million Developer Comments from GitHub»

Каталог комментариев и идентификаторов PR → выбор репозиториев → получение code diffs

Репозиторий	PR в исходной сводке	Активных разработчиков
swiftlang/swift	195 тыс.	7
ceph/ceph	164 тыс.	7
pytorch/pytorch	143 тыс.	7

Отбор: верхний 1% по активности + непрерывное участие; по 7 разработчиков на проект.

Финальная единица наблюдения — разработчик × месяц. Числа PR выше не равны объёму обработанных diffs.

Что именно представляет кодовый embedding?

01

ИЗМЕНЕНИЕ ФАЙЛА

$$\mathbf{v}_f = E(f_{\text{new}}) - E(f_{\text{old}})$$

Разность векторов
новой и старой версии.

02

PULL REQUEST

$$\mathbf{v}_p = \frac{1}{|F_p|} \sum_{f \in F_p} \mathbf{v}_f$$

Среднее по файлам
внутри одного PR.

03

РАЗРАБОТЧИК × МЕСЯЦ

$$\mathbf{v}_d(t) = \frac{1}{|P_d(t)|} \sum_{p \in P_d(t)} \mathbf{v}_p$$

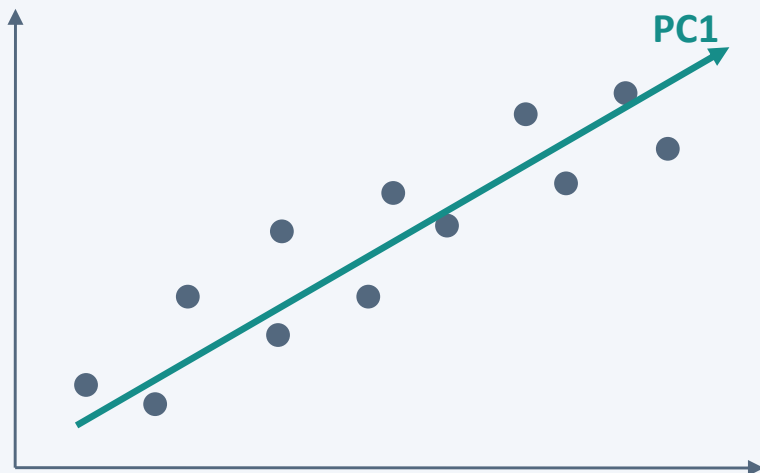
Среднее по PR автора
за временной период.

Один и тот же кодировщик $E = \text{intfloat/e5-base-v2}$ применяется к обеим версиям.

Изменения файлов собираются в сопоставимые помесечные траектории каждого разработчика.

PCA: выбираем ось наибольшей вариации

ГЕОМЕТРИЧЕСКАЯ ИДЕЯ



Учебная схема, не эмпирические данные

МАКСИМИЗАЦИЯ ДИСПЕРСИИ ПРОЕКЦИИ

$$u_1 = \arg \max_{\|u\|_2=1} \text{Var}[u^T(\mathbf{v} - \bar{\mathbf{v}})]$$

ПОЛУЧЕНИЕ КООРДИНАТЫ

$$\xi_d(t) = u_1^T (\mathbf{v}_d(t) - \bar{\mathbf{v}})$$

u_1 — направление PC1; $\bar{\mathbf{v}}$ — средний вектор.
Одна координата заменяет исходный вектор.

PC1 — статистическая координата изменений, а не шкала «хороший код — плохой код».

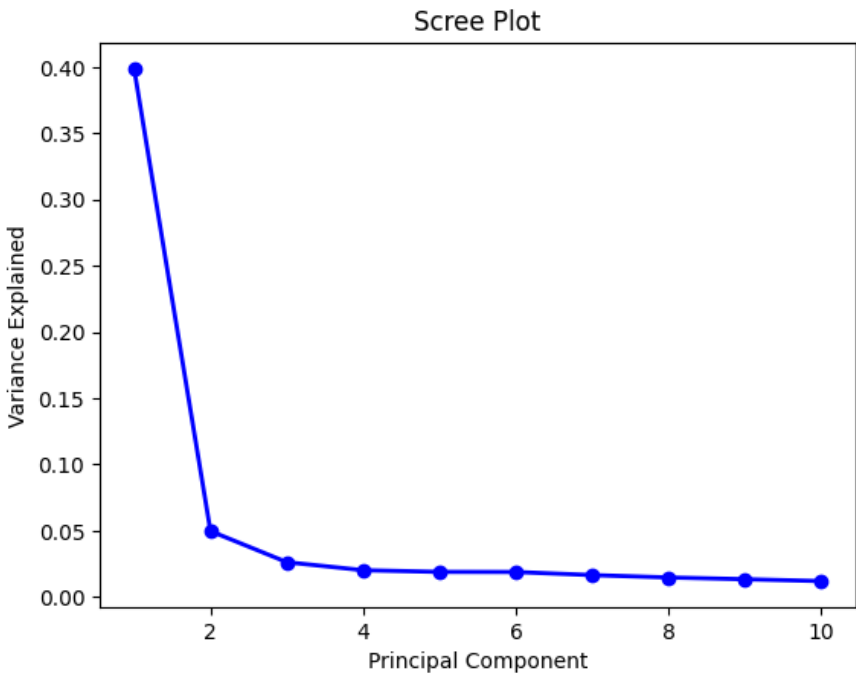
PCA: баланс сохранения структуры и простоты

СРЕДНИЕ ПО ТРЁМ ПРОЕКТАМ

Метод	Trustworth. ↑	Continuity ↑	MRRE ↓
PCA	0,6682	0,8256	0,1789
UMAP	0,6706	0,8164	0,1872
LLE	0,6453	0,7585	0,2437
MDS	0,5500	0,6593	0,3394

Trustworthiness / Continuity: сохранение соседств.
MRRE: ошибка относительных рангов.

ПРИМЕР: SWIFT · КОМПОНЕНТЫ PC1–PC10

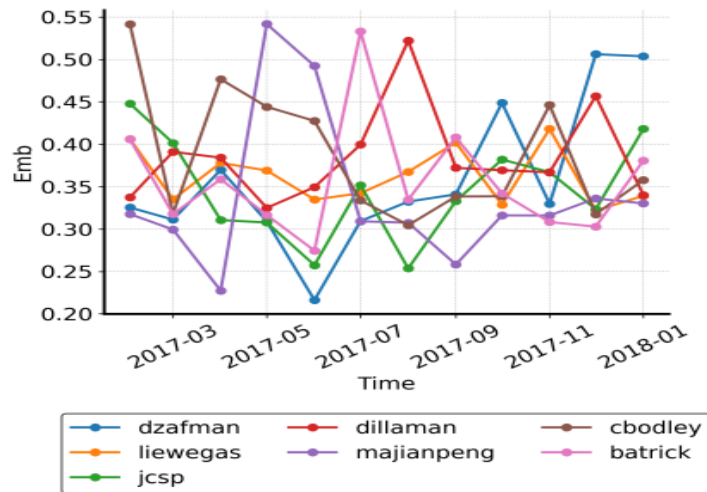


PCA имеет лучшие средние Continuity и MRRE; выбор одной PC1 — компромисс с потерей информации.

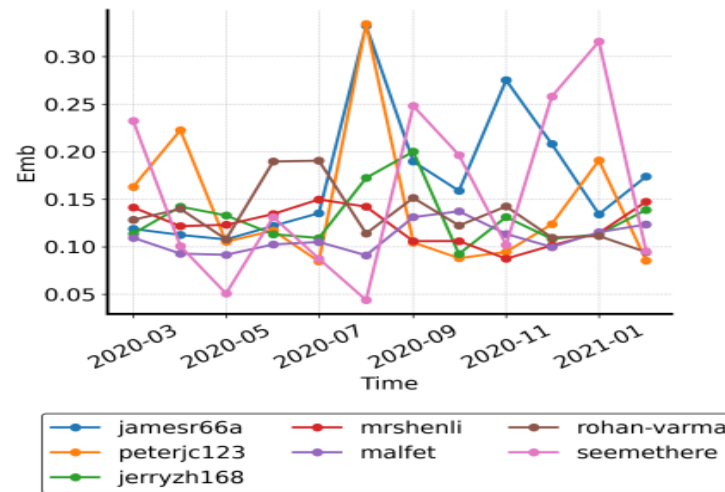
Из PC1 получается наблюдаемая траектория

$$x_d^e(t) = \frac{\xi_d(t) - \xi_{\min}}{\xi_{\max} - \xi_{\min}} \in [0, 1]$$

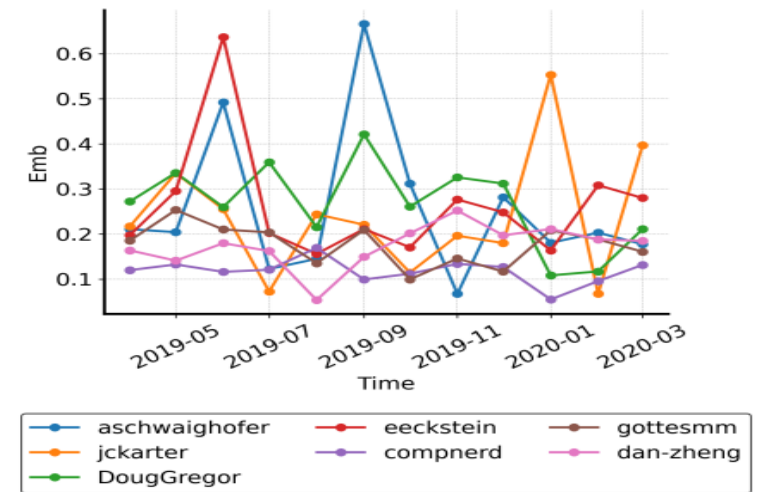
Проекция на PC1 → min–max
→ выраженное состояние $x_d^e(t)$



(a) ceph



(b) pytorch



(c) swift

Figure 3: Code “views” from the 7 most active users of repositories

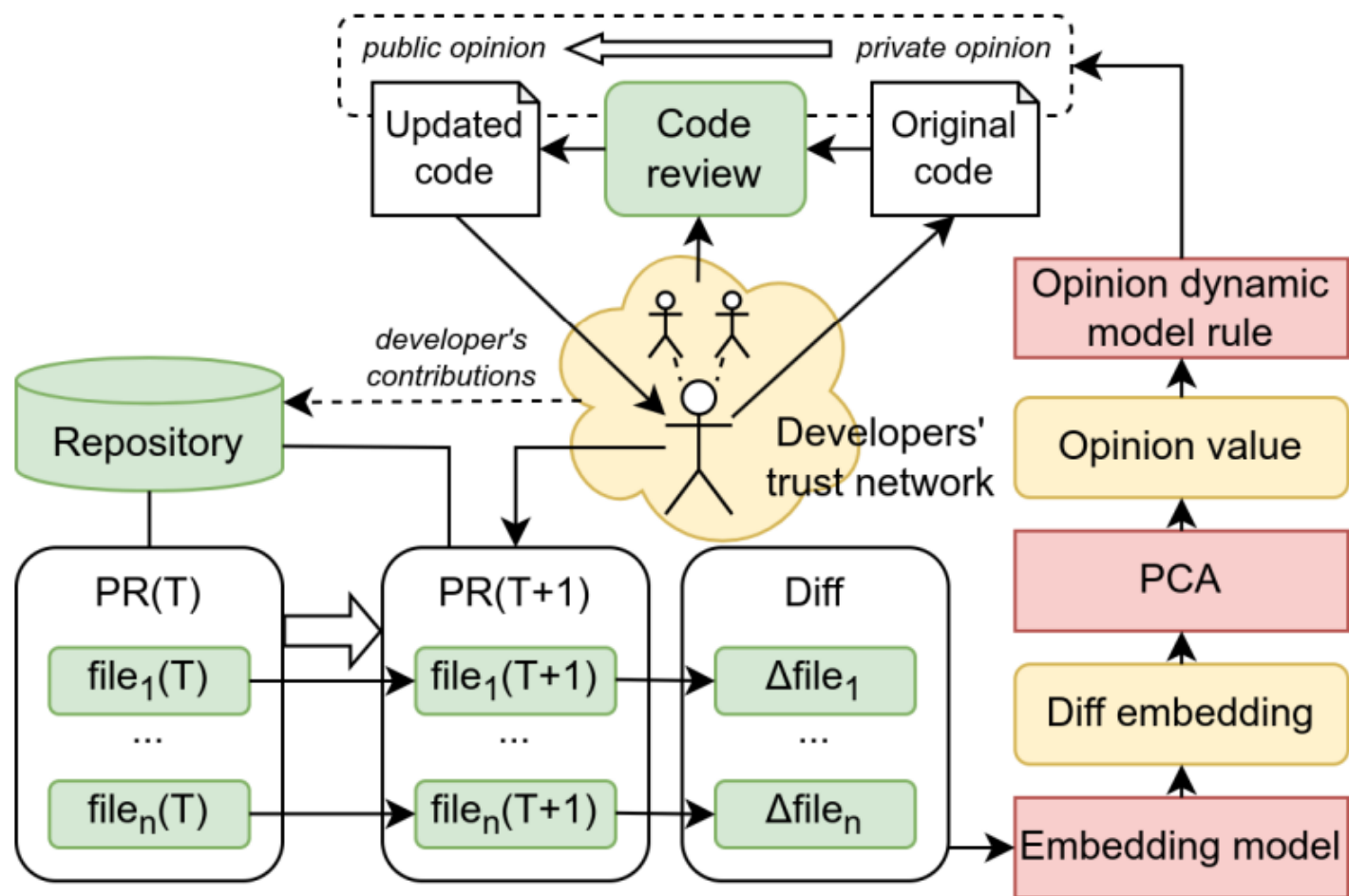


Figure 1: A general approach to opinion representation in GitHub developers' contributions

Код: подгонка и два отложенных периода

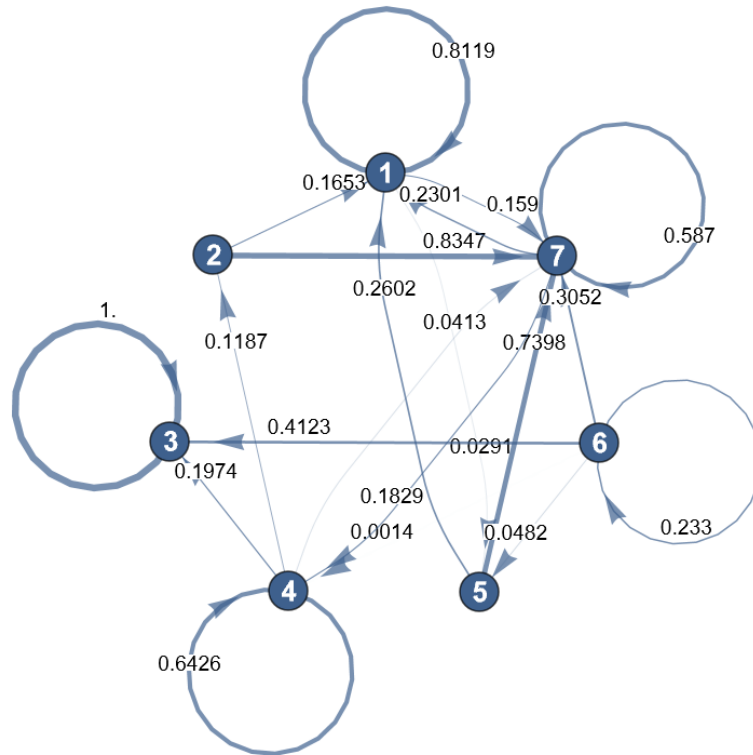
RMSE · МЕНЬШЕ — ЛУЧШЕ

Репозиторий	Подгонка: 1–10	Прогноз: 11	Прогноз: 12
Ceph	0,0801	0,2264	0,1739
PyTorch	0,0594	0,1762	0,0907
Swift	0,1311	0,4071	0,1915

Для всех трёх проектов ошибка периода 12 ниже ошибки периода 11.
Поведение ошибки зависит не только от длины горизонта.

В этой таблице PyTorch прогнозируется точнее, а Swift оказывается наиболее сложным случаем.

Структура зависимости разработчиков: пример Swift



ПРИМЕР РЕКОНСТРУКЦИИ: SWIFT

Реконструированная матрица не является прямой записью межличностного доверия.

ЧТО МОЖНО ЧИТАТЬ В W

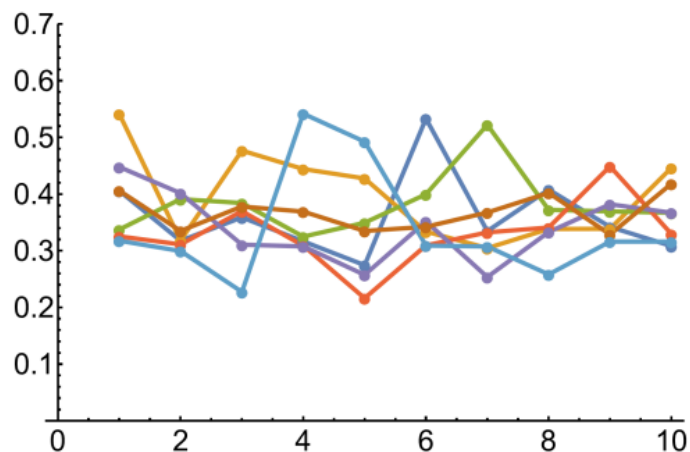
Собственная зависимость: w_{ii}

Внешний вклад: $w_{ij}, j \neq i$

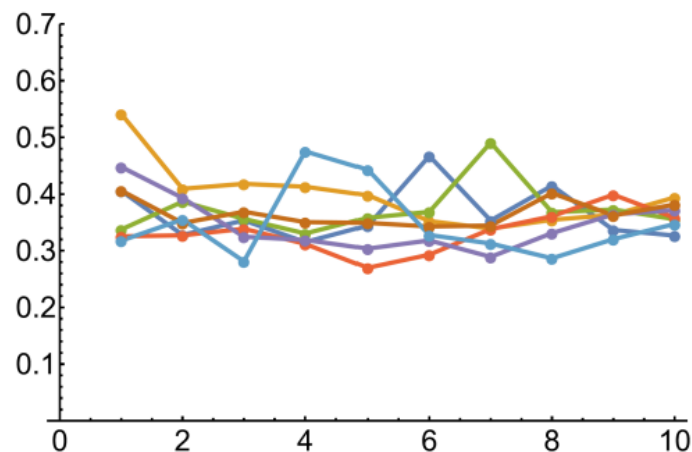
Это параметры модели изменения состояний. Роли, доверие и намерения требуют независимой предметной проверки.

Резерв: наблюдаемая и оценённые траектории Серр

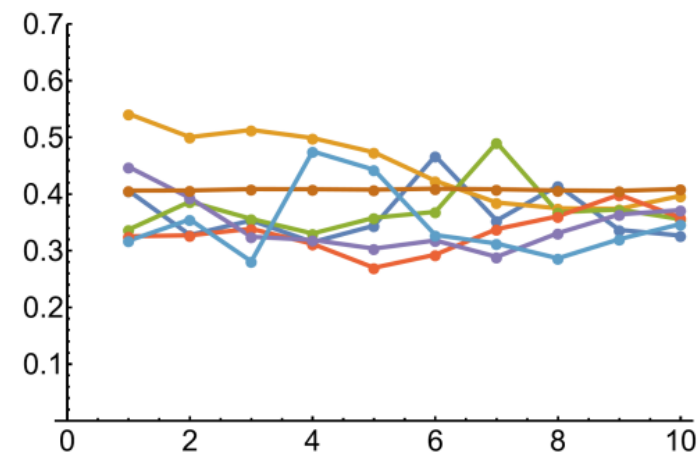
НАБЛЮДЕНИЯ



ВЫРАЖЕННЫЙ СЛОЙ



ЧАСТНЫЙ СЛОЙ



Серр: наблюдаемые значения → оценённый выраженный слой → оценённый частный слой.

Близость частного и выраженного слоёв — результат подгонки модели, а не измерение убеждений.

Спасибо за внимание!

<https://hreyulog.github.io/>

Under review:

Modeling Stylistic Co-evolution in Symbolic Music Heritage Collections

[Yulong He](#), [Ivan Smirnov](#), [Yanming Li](#) (<https://arxiv.org/abs/2607.23957>)

Opinion dynamics and mutual influence with LLM agents through dialog simulation

[Yulong He](#), [Dutao Zhang](#), [Sergey Kovalchuk](#), [Pengyi Li](#), [Artem Sedakov](#) (<https://arxiv.org/abs/2602.12583>)