

# 图如何理解世界： 水文、人文与化学的统一建模

瑞中科学俱乐部 · 2026.05.08

**贾林林**

瑞士伯尔尼大学计算机科学系模式识别组

联系方式: [linlin.jia@unibe.ch](mailto:linlin.jia@unibe.ch)

Oh geez, R-Rick!  
平台给我推荐了一堆  
“好友” .....它咋知道  
这些人跟我相似的啊?



Morty

$u^b$

$u^b$   
UNIVERSITÄT  
BERN



Morty

Oh geez, R-Rick!  
平台给我推荐了一堆  
“好友” .....它咋知道  
这些人跟我**相似**的啊?



Rick

\*嗝\* Morty, 与其在这瞎猜,  
不如我带你去看看!  
人和人的网络构成**图**,  
算一下**相似度**就懂了!



Morty

Oh geez, R-Rick!  
平台给我推荐了一堆  
“好友”……它咋知道  
这些人跟我**相似**的啊?



Rick

\*嗝\* Morty, 与其在这瞎猜,  
不如我带你去看看!  
人和人的网络构成**图**,  
算一下**相似度**就懂了!





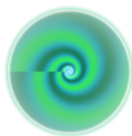
Morty

Oh geez, R-Rick!  
平台给我推荐了一堆  
“好友”……它咋知道  
这些人跟我**相似**的啊?



Rick

\*嗝\* Morty, 与其在这瞎猜,  
不如我带你去看看!  
人和人的网络构成**图**,  
算一下**相似度**就懂了!





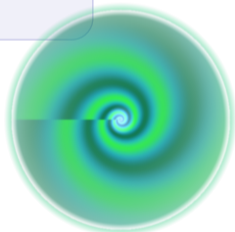
Morty

Oh geez, R-Rick!  
平台给我推荐了一堆  
“好友” .....它咋知道  
这些人跟我**相似**的啊?



Rick

\*嗝\* Morty, 与其在这瞎猜,  
不如我带你去看看!  
人和人的网络构成**图**,  
算一下**相似度**就懂了!



传送门已开启!

# 穿越时空，构建联系

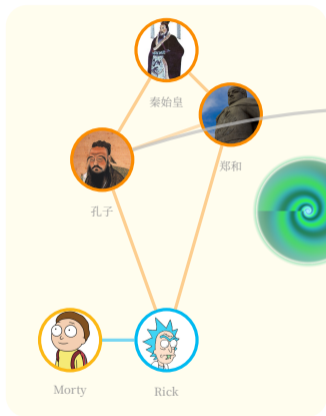
古代中国 (~公元前500年至今)



历史人物网络  
关系如何形成?

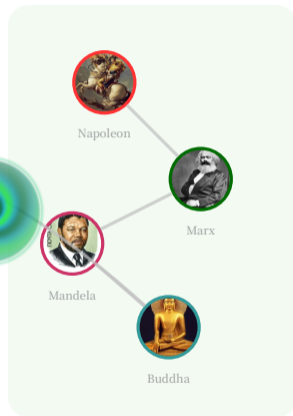
# 穿越时空，构建联系

古代中国 (~公元前500年至今)



历史人物网络  
关系如何形成?

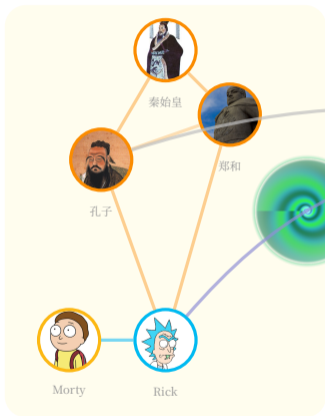
世界各地 (~1500年至今)



全球关系网络  
文明如何交汇?

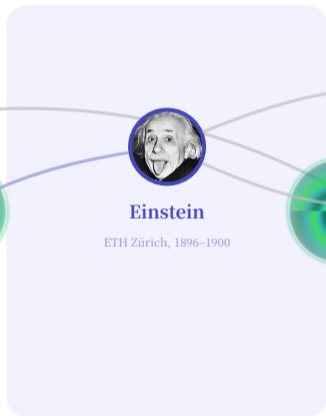
# 穿越时空，构建联系

古代中国 (~公元前500年至今)



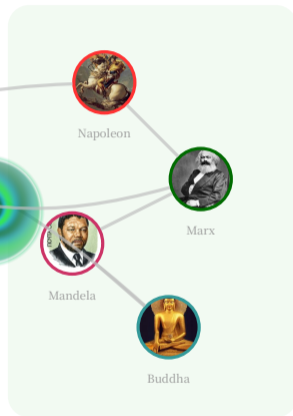
历史人物网络  
关系如何形成?

ETH 苏黎世



他们和 Einstein  
又有什么关系呢?

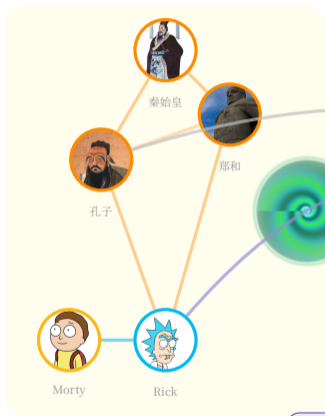
世界各地 (~1500年至今)



全球关系网络  
文明如何交汇?

# 穿越时空，构建联系

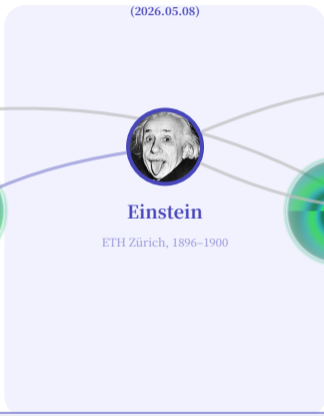
古代中国 (~公元前500年至今)



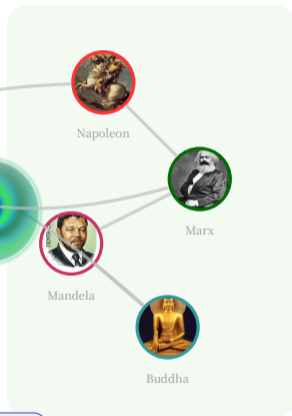
历史人物网络  
关系如何形成?

ETH 苏黎世

(2026.05.08)



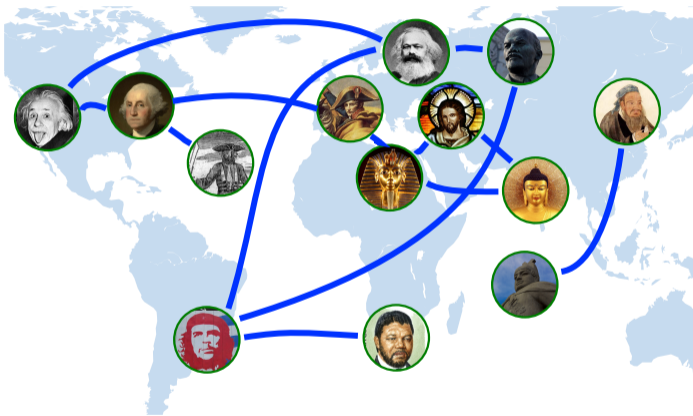
世界各地 (~1500年至今)



全球关系网络  
文明如何交汇?

因为今天的报告，他们产生了联系！  
人物 → 节点，关系 → 边  
— “社交网络”诞生了！

# 社交网络 $\rightarrow$ 图 $G = (V, E)$



13个跨越时空的历史人物，通过各种关系连接成一张“社交网络”



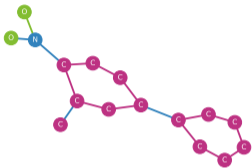
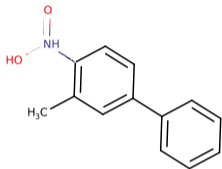
- 引言
- 案例一：智慧水文预测
- 案例二：计算化学
- 案例三：数字人文
- 通用平台
- 总结与展望

# 引言

# 图数据的多样性

$u^b$

化学信息学: 分子



计算机视觉: 3D 点云

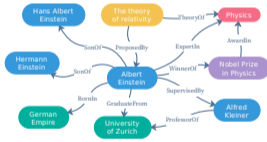


<sup>b</sup> UNIVERSITÄT  
BERN

社交媒体: 社交网络



知识图谱



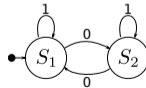
计算机视觉: 手写体



状态转移

状态转移表

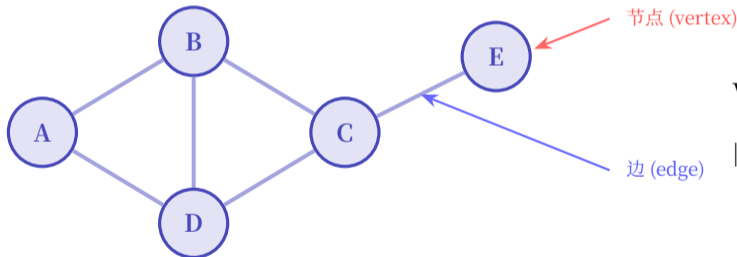
Current state \ Input	Input	
	0	1
$S_1$	$S_2$	$S_1$
$S_2$	$S_1$	$S_2$



# 什么是图 (Graph)?

## 定义

图  $G = (V, E)$  由一个有限**节点集**  $V$  和**边集**  $E \subseteq V \times V$  组成。  
节点和边上可以携带**标签** (类别信息) 和**属性** (连续特征)。

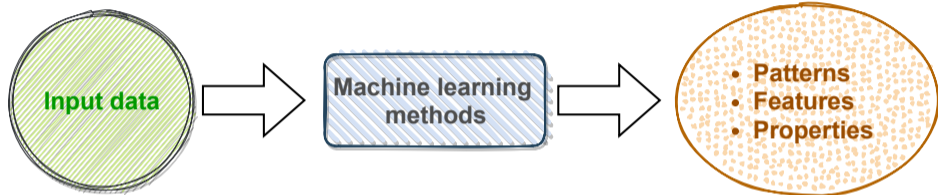


$$V = \{A, B, C, D, E\}$$

$$|V| = 5, \quad |E| = 6$$

**核心优势:** 图能够自然地表达**实体之间的关系**,  
不受限于固定维度或网格结构。

# 图空间的挑战



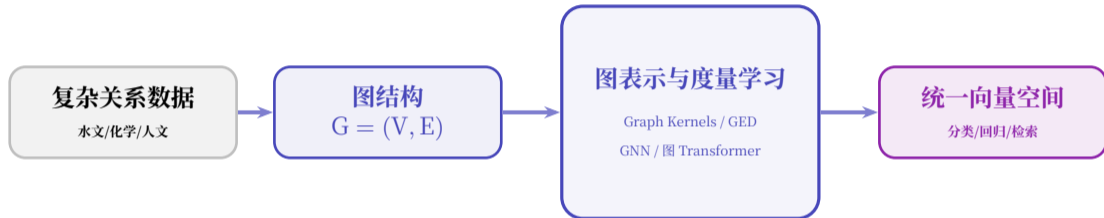
## 欧氏空间

- 连续空间
- 元素维度固定
- 传统ML直接适用

## 图空间

- 离散空间
- 节点、边数量可任意变化
- 多种标签/属性，图同构问题
- **无法直接使用传统ML方法!**

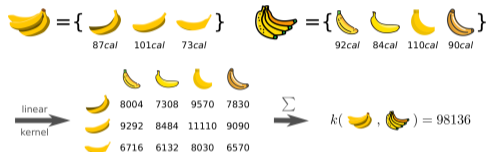
# 我们的方法：图表示学习与度量学习



**核心思想：** 将不同领域的复杂关系统一表示为图，  
再通过图学习方法映射到可计算的向量空间

# 图的相似度如何计算?

## Graph Kernels



提取子结构特征  $\phi(G)$

$$k(G_1, G_2) = \langle \phi(G_1), \phi(G_2) \rangle$$

共同子结构越多  $\rightarrow$  越相似

## Graph Edit Distance



最少编辑操作  $G_1 \rightarrow G_2$

$$\text{ged}(G_1, G_2) = \min_{\pi} \sum_i C(\pi_i)$$

插入/删除/替换节点和边

**Graph Neural Networks:** 学习图的向量表示  $\mathbf{h}_G$ , 通过

$$d(G_1, G_2) = \|\mathbf{h}_{G_1} - \mathbf{h}_{G_2}\|$$
 端到端学习

三种方法各有优劣, 后续案例中将根据场景灵活选用

# 案例一：水文

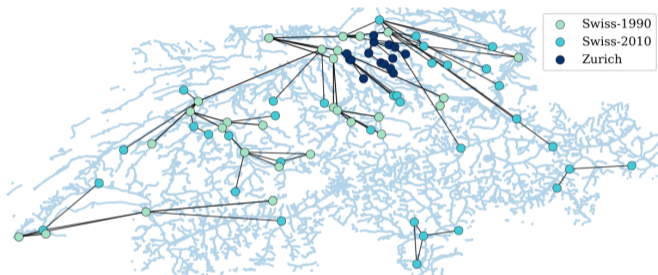
# 为什么关注河流水温?

- **生态指标:** 水温直接影响水生生物的生存  
鱼类繁殖、藻类生长、溶解氧含量
- **气候变化:** 全球变暖导致河流水温上升  
瑞士河流过去30年平均升温 ~1.5°C
- **水资源管理:** 饮用水、工业冷却、农业灌溉  
水温超标可能导致生态灾难
- **核心问题:** 能否利用气象数据和空间结构  
来精确预测河流各站点的水温?

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
12705	6.29	5.27	6.02	5.68	5.9	7.21	6.54	4.99	5.07	6.44	2.98	6.52	6.4	1.07	6.29	4.15										
12706	6.3	5.3	6.02	5.71	5.85	6.76	4.3	3.9	4.78	6.25	2.53	6.47	6.88	1.18	6.19	4.05										
12707	6.22	4.95	6.04	5.11	5.81	6.13	3.98	2.42	3.8	6.19	1.05	6.24	6.12	0.4	6.07	3.77										
12708	6.05	4.69	5.38	4.86	5.27	5.59	3.29	2.05	4.01	6.15	0.44	6	0.77	0.44	6.02	2.7										
12709	6.11	4.76	5.78	4.88	5.61	7.02	4.03	4.68	5.85	6.28	2.78	6.17	1.02	1.31	6.17	4.99										
12710	6.18	5.22	5.98	5.54	6	6.78	4.44	3.23	4.66	6.26	2.16	6.33	1.56	1.54	6.1	4.5										
12711	6.17	4.82	5.91	4.84	5.23	5.84	3.8	2.24	4.58	6.25	1.47	6.18	1.51	1.42	6.07	4.38										
12712	6.36	4.86	6.25	5.13	5.63	6.95	4.45	3.92	5.93	6.28	2.87	6.26	2.44	2.31	6.14	4.95										
12713	6.34	5.22	6.3	5.29	5.65	6.4	4.58	3.25	4.88	6.19	2.75	6.36	1.86	1.68	6.08	4.57										
12714	6.36	5	6.04	5.16	5.67	6.02	3.03	2.15	4.44	6.08	1.71	6.22	0.77	0.83	6.03	4.31										
12715	6.29	4.7	5.79	4.97	5.96	6.07	3.61	2.11	4.82	6.13	1.41	6.14	0.54	0.91	6.03	3.88										
12716	6.18	4.87	5.94	5.32	5.61	6.89	4.85	3.63	5.51	6.14	3.16	6.26	1.96	2.02	6.11	4.41										
12717	5.98	4.97	6.68	5.49	5.75	5.84	3.47	2.77	3.82	5.88	2.19	6.16	0.54	0.97	6	3.37										
12718	5.67	4.52	6.45	4.74	5.22	4.84	2.7	2.17	3.12	5.88	1.11	5.95	0.89	0.29	5.95	2.43										
12719	5.37	4.21	5.14	4.59	5.03	5.85	2.45	1.61	3.05	5.82	0.55	5.79	0.12	0.31	5.76	1.47										
12800	5.18	4.17	4.75	4.23	4.96	5.05	2.49	1.19	2.79	5.73	0.18	5.99	0.14	0.31	5.71	1.67										
12801	5.11	4.2	4.66	4.19	4.84	5.21	2.2	1.51	3.44	5.74	0.17	5.42	0.15	0.68	5.69	2										
12802	5.22	4.18	4.91	4.25	5.13	5.81	2.54	2.18	2.78	5.64	0.67	5.42	0.17	0.46	5.8	2.12										
12803	5.47	4.29	5.07	4.66	5.09	6.03	3.3	3.18	4.37	5.83	1.89	5.82	0.17	0.98	5.78	2.5										
12804	5.76	4.7	5.65	5.36	5.56	7.21	4.71	4.7	4.95	5.76	3.05	5.84	0.84	0.96	5.47	2.84										
12805	5.71	4.67	5.75	4.87	5.34	6.42	4.11	3.74	4.47	5.76	2.41	5.87	1.12	1.09	5.76	3.74										
12806	5.59	4.49	5.3	4.47	4.9	5.43	3.43	2.51	3.39	5.6	0.87	5.87	0.03	0.49	5.87	3.19										
12807	5.27	4.21	4.8	3.88	4.68	5.1	2.74	1.66	2.78	5.45	0.26	5.34	0	0.31	5.89	2.56										
12808	4.91	3.97	4.51	3.47	4.47	4.95	2.27	1.19	2.43	5.25	0.06	5.09	0.02	0.26	5.3	1.9										
12809	4.47	3.46	4.09	3.41	4.19	3.9	1.14	0.24	1.88	5.18	0	5.02	0.04	0.26	5.42	1.95										
12810	4.04	3.24	3.52	3.07	3.84	3.16	0.68	0	1.26	4.87	0	4.74	0.06	0.25	5.4	1.4										
12811	3.79	2.9	3.12	2.99	3.93	3.27	0.95	0.11	1.3	4.98	0.01	4.83	0.07	0.24	5.39	1.32										
12812	3.49	2.86	2.85	3.1	3.8	2.86	0.2	0.1	1.34	4.7	0.01	4.79	0.09	0.25	5.47	1.26										
12813	3.31	2.65	2.45	2.77	3.55	2.38	0.14	0.1	1.31	4.41	0.01	4.65	0.11	0.25	5.29	1.22										
12814	3.5	2.96	2.89	3.2	3.92	3.74	1.12	0.14	1.96	4.96	0.03	4.76	0.16	0.32	5.23	1.18										
12815	3.74	3.05	3.25	4.16	4.2	5.03	1.92	0.28	3.05	4.45	0.09	5.11	0.17	0.41	5.26	1.15										

瑞士河流水温时空数据

# 河流即是图



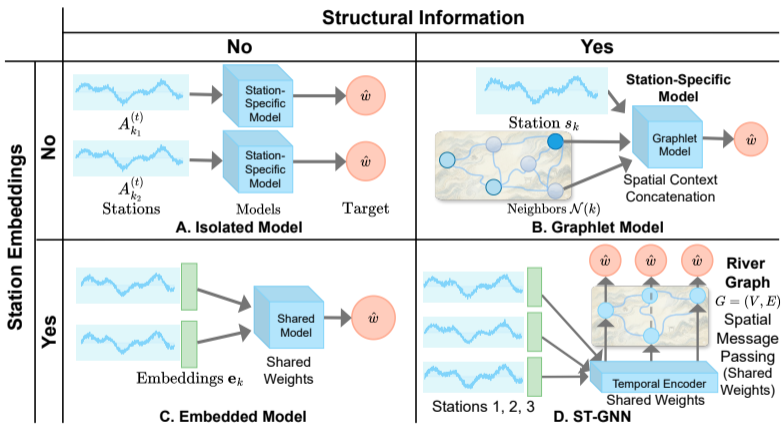
**Swiss-1990**  
28 站点 · 30 年  
1596 km<sup>2</sup>/站

**Swiss-2010**  
63 站点 · 10 年  
1039 km<sup>2</sup>/站

**Zurich**  
15 站点 · 13 年  
74 km<sup>2</sup>/站

监测站 → 节点, 河段 → 边 河流网络天然形成图结构!

# 四种建模架构



Isolated  $\xrightarrow{+邻居}$  Graphlet  $\xrightarrow{+共享权重}$  Embedded  $\xrightarrow{+图结构}$  ST-GNN

Benchmarking Transformers on Spatio-Temporal River Water Temperature Modeling, submitted to ICPR 2026

# 时序建模: LSTM vs Transformer

## LSTM

- 循环结构, 逐步处理序列
- 门控机制控制信息流
- 训练稳定, 参数较少
- 长距离依赖能力有限

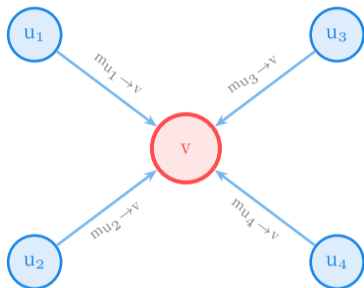
## Transformer

- 自注意力机制, 并行处理
- 直接建模任意位置间的关系
- 全局感受野
- 需要位置编码 (PE)

### 关键发现

- Transformer 在精度上**优于 LSTM 3-12%** (相同窗口长度时)
- 但 LSTM 在结合图结构时**更稳定**
- Transformer 计算时间**高 31-308%**
- **Station embeddings** 对两种模型都至关重要

# 空间建模：图神经网络 (GNN)



1. 消息:  $m_{u \rightarrow v} = \phi(h_u, h_v, e_{uv})$

2. 聚合:  $\bar{m}_v = \bigoplus_{u \in \mathcal{N}(v)} m_{u \rightarrow v}$

3. 更新:  $h'_v = \psi(h_v, \bar{m}_v)$

变体: MPNN, GAT, GCN, GIN, GraphSAGE

上游站点的水温信息通过图结构传播到下游  
→ 捕获河流网络中的空间依赖关系

## 传统指标的局限

RMSE、MAE、NSE 等单一指标无法全面评估模型在**不同条件**下的表现。

## Weighted Temporal Performance Score (WT-Score)

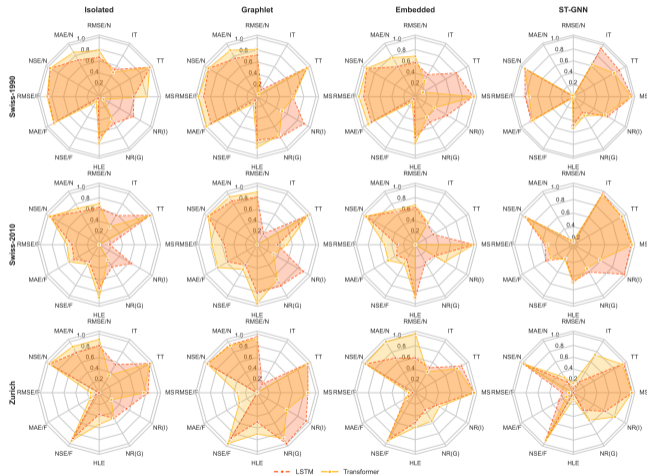
$$\text{WT-Score} = \frac{\sum_i w(c_i) M(c_i)}{\sum_i w(c_i)}$$

一个**通用**的加权评估框架：对条件变量  $c$ （如窗口长度、预测步长、噪声强度）按权重  $w(c)$  聚合性能得分  $M(c)$ 。根据评估目标**实例化**为：

- **HLE**（历史长度效率）： $c = \text{窗口长度}$ ,  $w(l) = 2^{-l/h_1/2}$  – 强调短窗口性能
- **MHFA**（多步预测精度）： $c = \text{预测步长}$ ,  $w(h) = 1$  – 各步长均等评估
- **NR**（噪声鲁棒性）： $c = \text{噪声强度}$ ,  $w(\sigma) = \sigma^\gamma$  – 强调高噪声条件

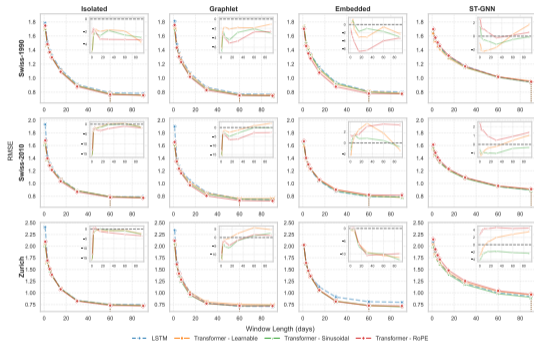
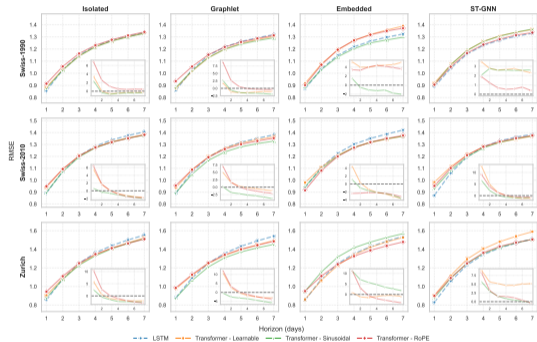
→ 统一框架，多维度实例化，为模型选择提供全面指导

# 结果：多维度性能对比



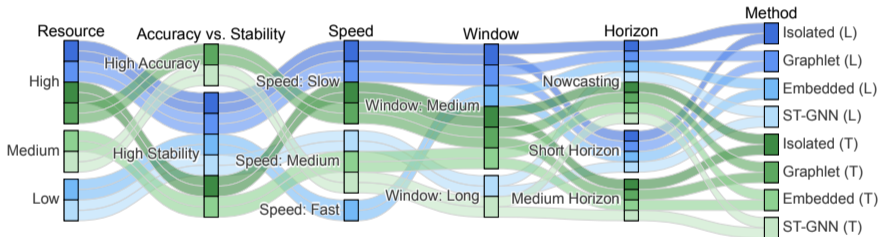
雷达图展示各模型在不同评估维度上的综合表现 · 共训练 13,200 个模型

# 结果：多步预测性能



- Nowcasting RMSE: 0.7–0.9°C; 7天预测 RMSE: 1.2–1.3°C
- Embedding 模型实现 88–99% 参数削减

# 基于经验的模型选择建议



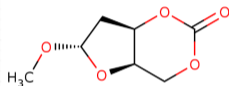
根据应用场景（精度/效率/鲁棒性）选择最优建模架构

## 案例二：化学

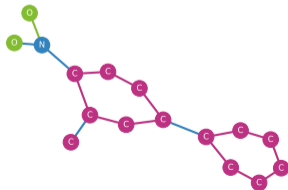
# 从结构到功能：分子的性质在哪里？

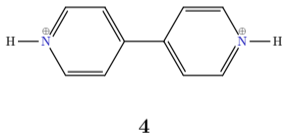
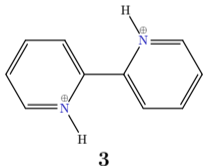
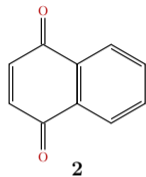
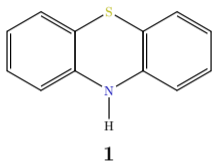
- 分子性质由**结构**决定，而非仅是组成元素
- **氧化还原电位** (Redox Potential):
  - 衡量分子得失电子的能力
  - 电池、药物、有机合成的关键参数
  - 实验测量昂贵且耗时
- **SMILES** 表示法 → 分子图
  - 原子 → 节点
  - 化学键 → 边

C=C1OC[C@@H]2[C@@H](C[C@@H](OC)O2)O1



SMILES 与分子图表示



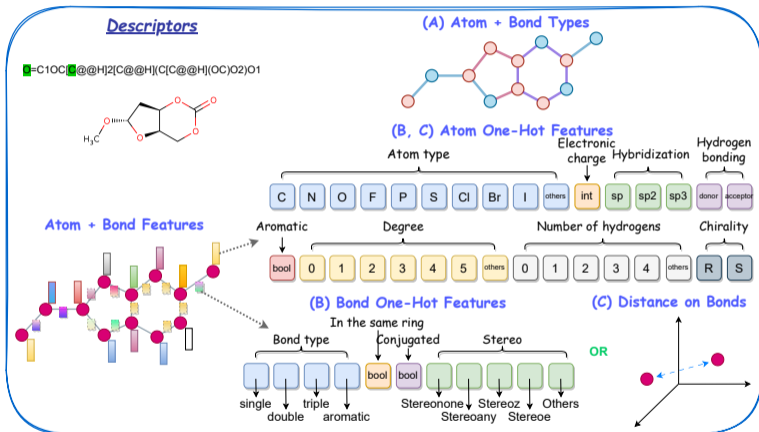


四大分子家族的代表性结构

159 有机化合物 · 318 氧化还原反应 · 4 分子家族

吩噻嗪 (Phenothiazines) · 醌 (Quinones) · 2,2'-联吡啶盐 · 紫精 (Viologens)

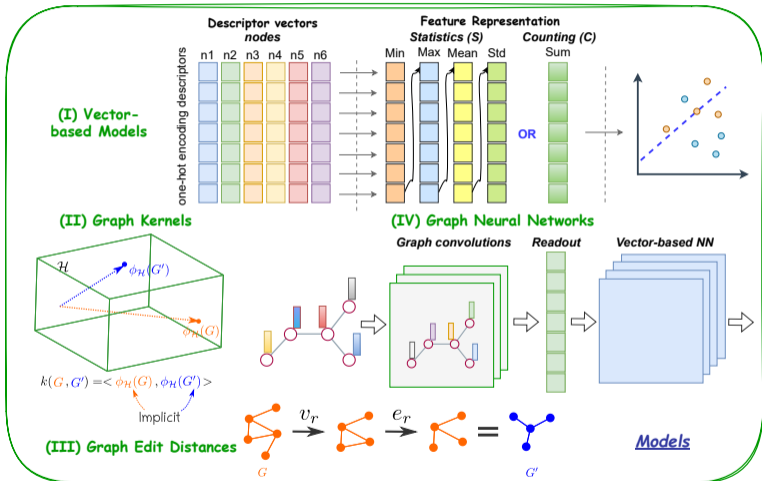
# 分子描述符



两大类描述符：向量描述符 (7种基线方法) vs 图描述符 (12种图方法)

Predicting redox potentials by graph-based machine learning methods, J. Comput. Chem., 2025

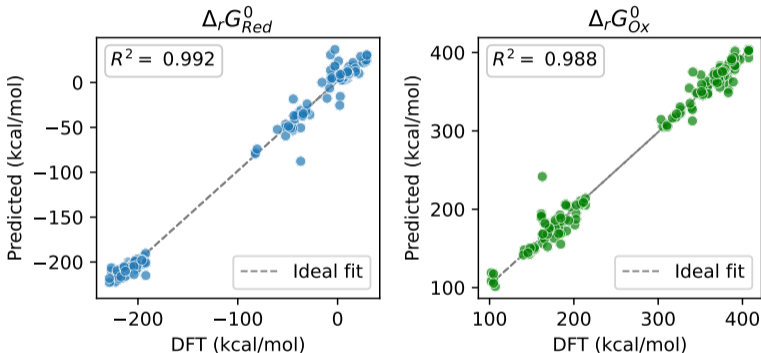
# 图方法详解



四大类模型：(I) 向量基线 · (II) 图核 · (III) 图编辑距离 · (IV) 图神经网络

共 19 种方法：7 种向量基线 + 5 种图核 + 1 种 GED + 6 种 GNN

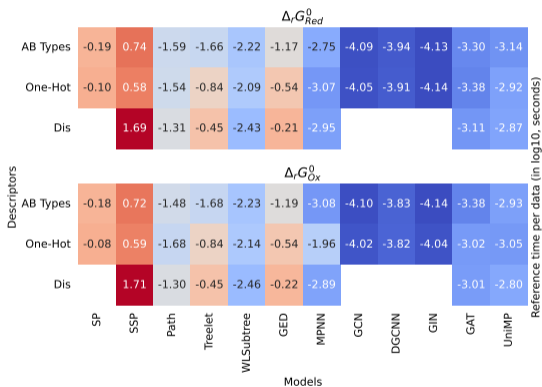
# 预测结果



DFT计算值 vs ML预测值的相关性

最佳 MAE: 还原电位 5.6 kcal/mol, 氧化电位 7.2 kcal/mol

# 效率与精度的权衡



各方法计算时间对比 (对数尺度)

- **图核方法**: 速度最快  
适合小数据集快速验证
- **GED方法**: 可解释性强  
提供图间对应关系
- **GNN方法**: 精度最高  
端到端学习, 数据越多越好
- **图表示**捕获了向量方法遗漏的结构信息

## 案例三：人文

# 为什么要从手写数据中提取图结构?

- **优势:**

- 捕获**拓扑结构信息**, 超越像素层面
- 对变形、旋转、书写风格**鲁棒**
- 提供**可解释**的笔画和关系特征
- 支持**无标注**设置

- **应用:**

- 古籍手稿数字化与检索
- 少样本/跨作者字符识别
- 公式和文档结构理解
- 工程图纸符号提取



George Washington 手稿 (histograph.ch)

# 现有方法的局限

$u^b$

<sup>b</sup>  
UNIVERSITÄT  
BERN

1) For each connected component...



3) ...invert junction points



2) ...find junction points



4) For each connected subcomponent...



5) ...find start- and end points



7) Replace neighbours of junction point



6) ...add nodes along path



8) Add edges based on skeleton



Michael Stauffer, Andreas Fischer and Kaspar Riesen

30.11.16

8

Michael Stauffer, Andreas Fischer and Kaspar Riesen

30.11.16

9

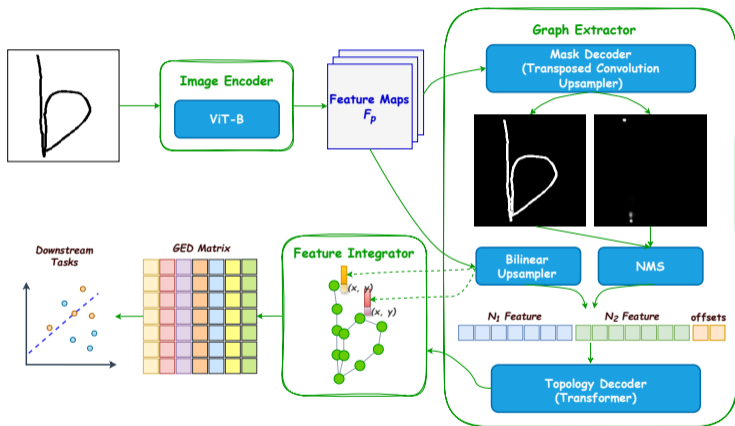
基于启发式图像处理方法（二值化、骨架化、关键点检测）

→ 针对特定手稿类型设计

→ 难以跨类型泛化

Stauffer et al., A novel graph database for handwritten word images, S+SSPR 2016

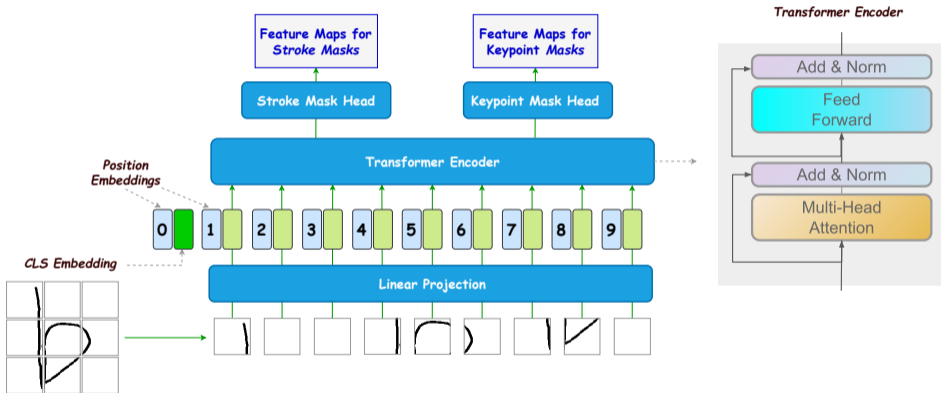
# GraphInk 框架



## 基于 Segment Anything Model (SAM) 和 Transformer 的端到端图提取

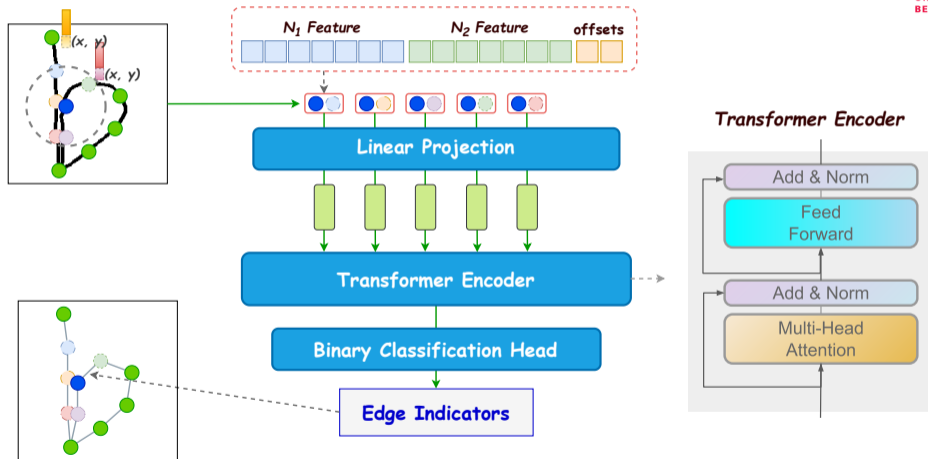
GraphInk: Graph Structure Extraction from Offline Handwriting via SAM and Transformers, submitted 2025

# 模块 1: 图像编码器 (Vision Transformer)



将手写图像分块 (patch) 编码为特征向量序列

# 模块 2: 拓扑网络



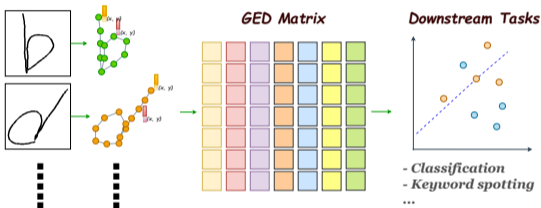
预测笔画概率图和关键点概率图, 通过 NMS 精简节点

# 模块 3: 图编辑距离与下游任务



$$\text{ged}(G_1, G_2) = \min_{\pi \in \Pi} \sum_i C(\pi_i)$$

字符分类 (KNN + GED) · 关键词检索 (GED ranking)



# 结果：性能对比

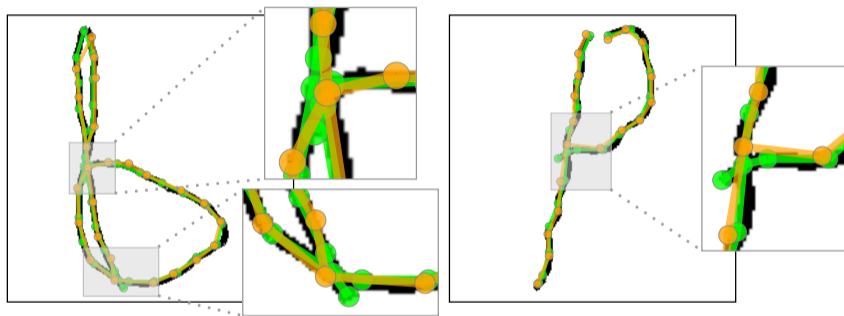
**Table:** 模型在下游任务上的性能

Methods	Unipen (accuracy)	Washington (mAP)
Heuristic (baseline)	0.634 $\pm$ 0.007	0.613 $\pm$ 0.038
GraphInk	<b>0.701</b> $\pm$ 0.007	<b>0.630</b> $\pm$ 0.034
GraphInk w.o. embeddings	0.680 $\pm$ 0.008	0.622 $\pm$ 0.035

**Table:** 笔画和关键点掩码的 IOU

Masks	Unipen	Washington
Stroke masks	0.963 $\pm$ 0.0004	0.918 $\pm$ 0.005
Keypoint masks	0.477 $\pm$ 0.003	0.528 $\pm$ 0.005

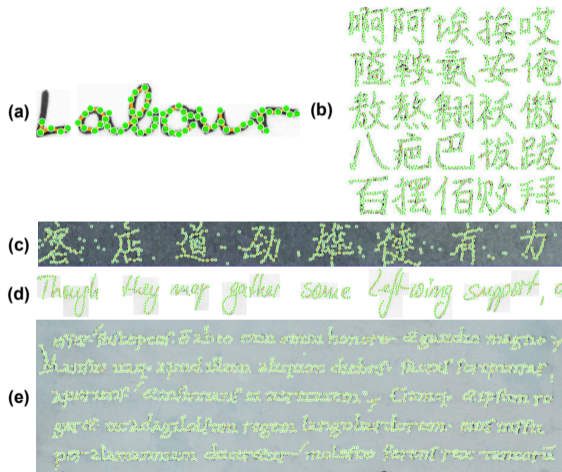
# 学习的图 vs 启发式的图



绿色：启发式方法    橙色：GraphInk

在交叉点/连接处，启发式方法产生大量冗余重叠节点，  
而 GraphInk 在训练中学习将其精炼为更精确的关键点位置

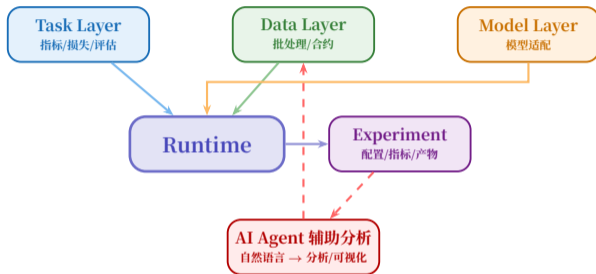
# 泛化能力：未见数据集



- 仅在 Unipen 上训练
- 在多种未见数据集上直接提取图结构
- 涵盖英文、中文、阿拉伯文等多种手写体数据集
- **无需重新训练!**

# 通用平台

# liulian: 时空图实验平台



## Support

- Optimizer (超参调优)
- Loggers (指标记录)
- Viz (可视化)
- Plugins (领域适配)
- Manifests (数据合约)

Task-driven — 任务语义与模型逻辑分离 · AI Agent — 非编程科学家也能轻松分析数据 · Unified API — 适配 PyTorch / sklearn / 自定义模型

**liulian-python:** 面向时空/图数据的任务驱动实验框架 + AI Agent 辅助分析  
[github.com/jajupmochi/liulian-python](https://github.com/jajupmochi/liulian-python)

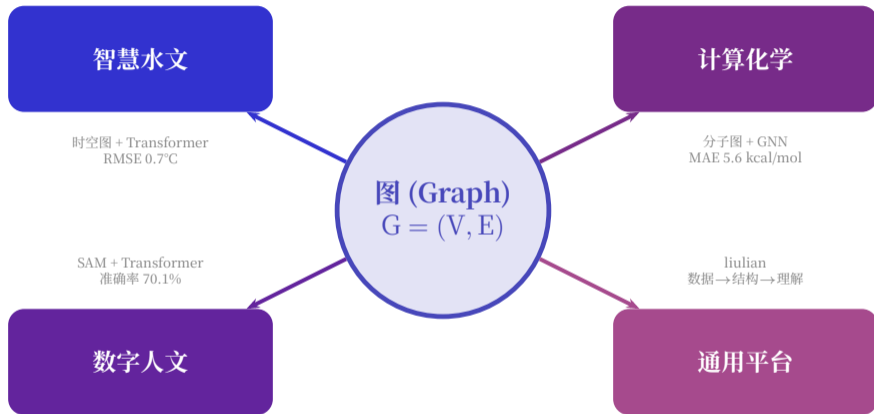


点击播放 / Click to play · 若无法播放：点此外部打开视频

**liulian-python**: 面向时空/图数据的任务驱动实验框架 [github.com/jajupmochi/liulian-python](https://github.com/jajupmochi/liulian-python)

# 总结与展望

# 总结



一种结构，多种语言 —— 图是理解复杂关系的通用框架

- **图基础模型 (Graph Foundation Models)**
  - 预训练大规模图模型，跨领域迁移
  - 利用 LLM 增强图推理能力
- **跨域迁移学习**
  - 水文中的时空图模型 → 交通、能源网络
  - 分子图方法 → 材料科学、药物设计
- **多模态图学习**
  - 结合视觉、文本、图结构的统一学习
  - Vision-Language-Graph 大模型
- **平台生态**
  - liulian 平台持续扩展，欢迎合作!

# 感谢聆听!

## 欢迎提问与交流



微信



GitHub

[linlin.jia@unibe.ch](mailto:linlin.jia@unibe.ch)

**Open to positions!**



## NMS 节点精简算法

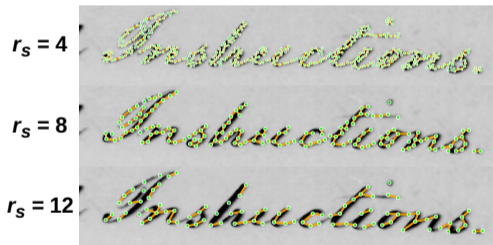
---

### Algorithm Non-Maximum Suppression (NMS) for Probability Maps

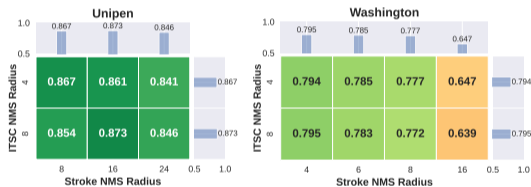
---

- 1: **Input:** 概率图  $P$ , 距离阈值  $r_{\text{nms}}$ , 概率阈值  $\tau$
  - 2: **Output:** 精简后的节点集  $V'$
  - 3: 按概率  $P(n)$  降序排列所有节点
  - 4: 移除  $P(n) < \tau$  的节点
  - 5: 初始化  $V' \leftarrow \emptyset$
  - 6: **for** 排序列表中的每个节点  $n$  **do**
  - 7:     **if**  $V'$  中没有节点  $n'$  满足  $\text{dist}(n, n') < r_{\text{nms}}$  **then**
  - 8:         将  $n$  加入  $V'$
  - 9:     **end if**
  - 10: **end for**
  - 11: **return**  $V'$
-

# NMS 半径参数研究



不同 NMS 半径下的图提取效果



不同 NMS 半径下的下游任务性能