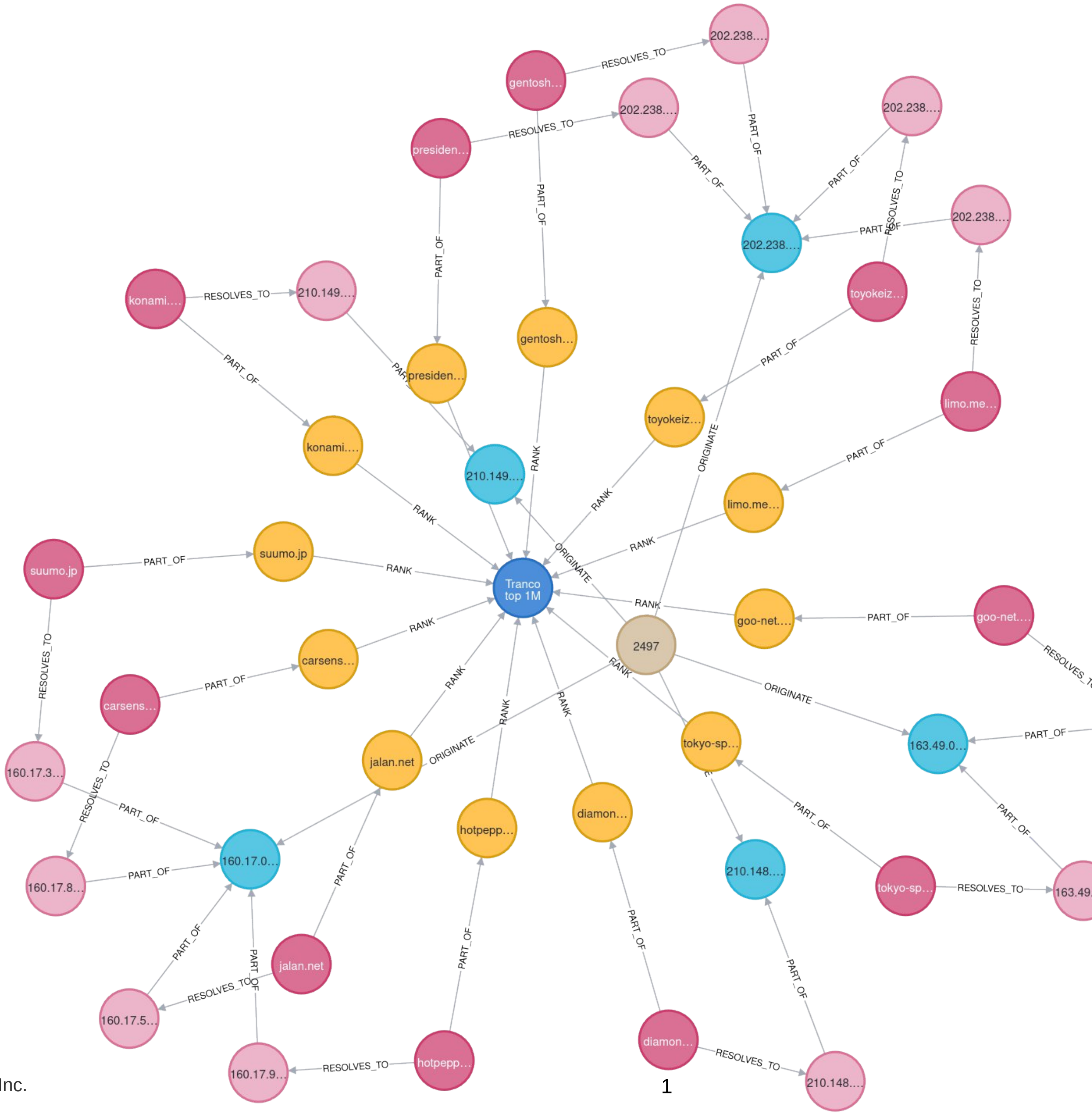
IYP embedding

Romain Fontugne

Internet Initiative Japan

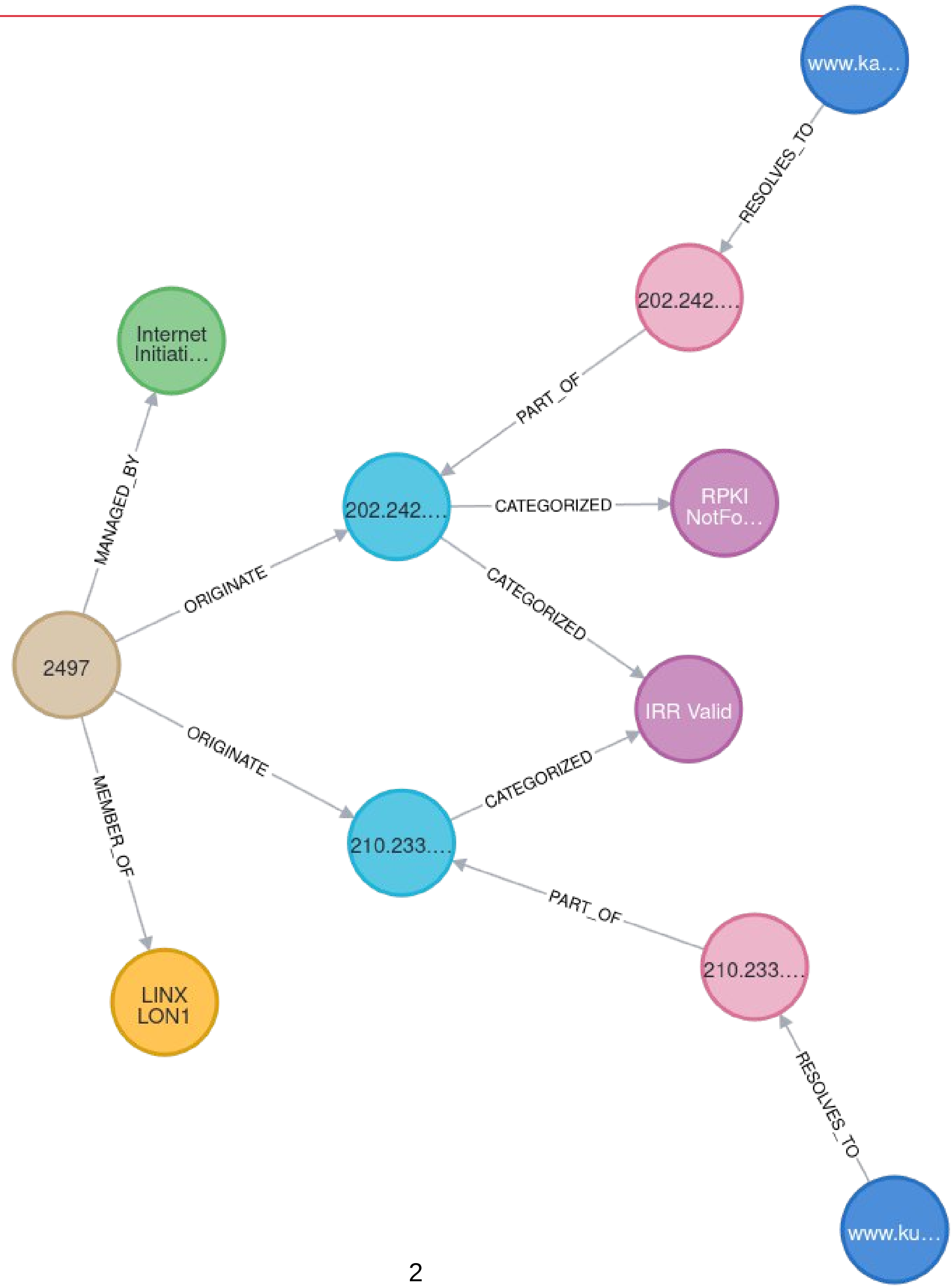
February 2026
AIMS



Internet Yellow Pages?

- **Knowledge graph of Internet resources**
  - 80+ datasets in one database!
  - https://iyp.iijlab.net

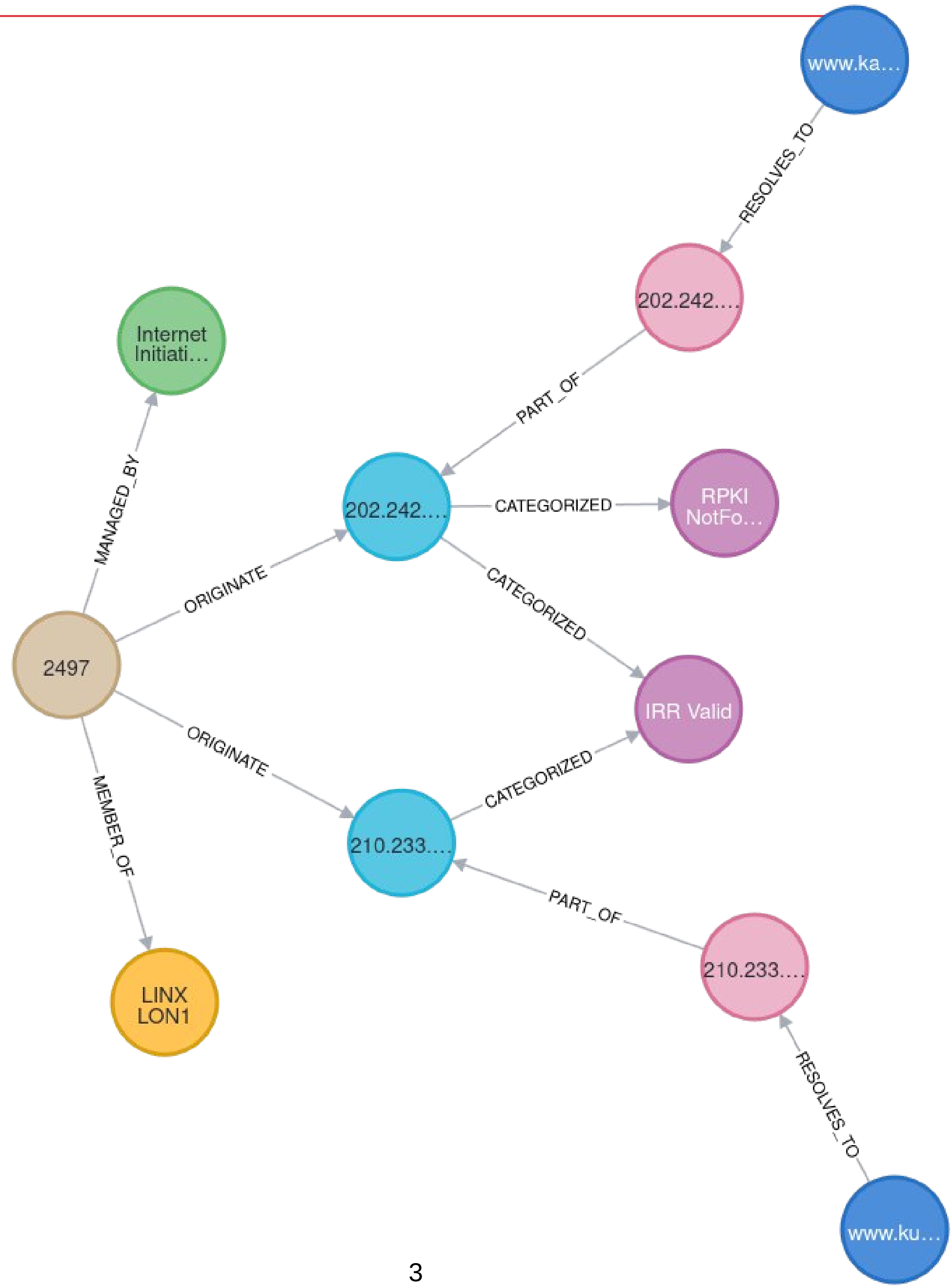
- **Usual usages**
  - IHR website
  - Dataset extraction
  - Q&A / brainstorm
  - Data cleaning



Better IYP analysis

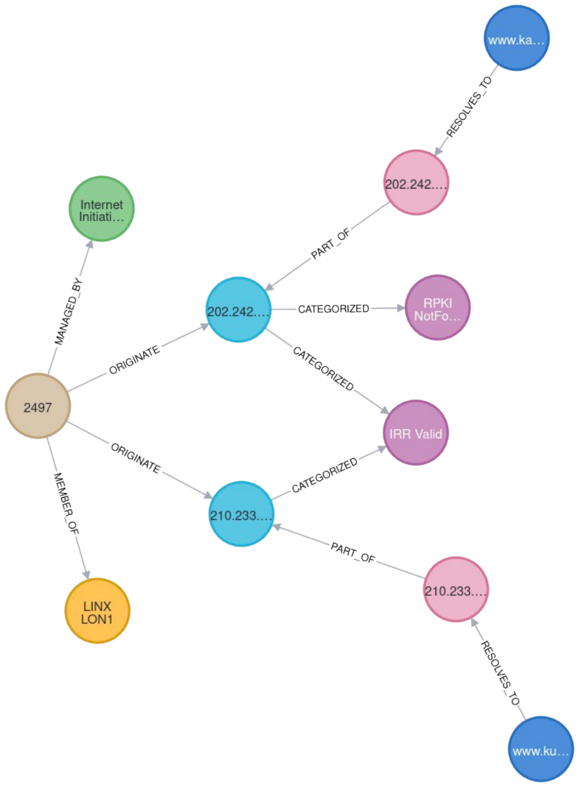
- **Find common patterns? Outliers?**
- **Cluster ASes? prefixes? IXPs?**
- **Semantic similarity?**

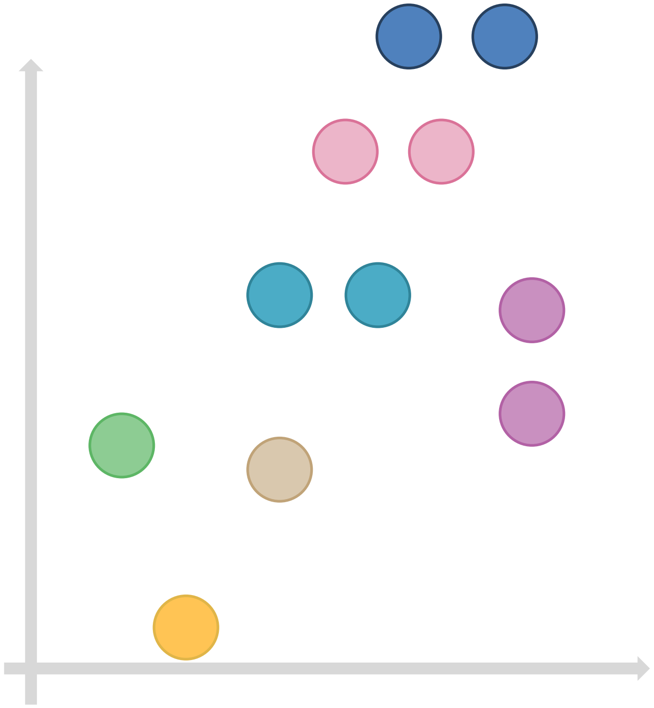
→ Need better tools to leverage information stored in IYP



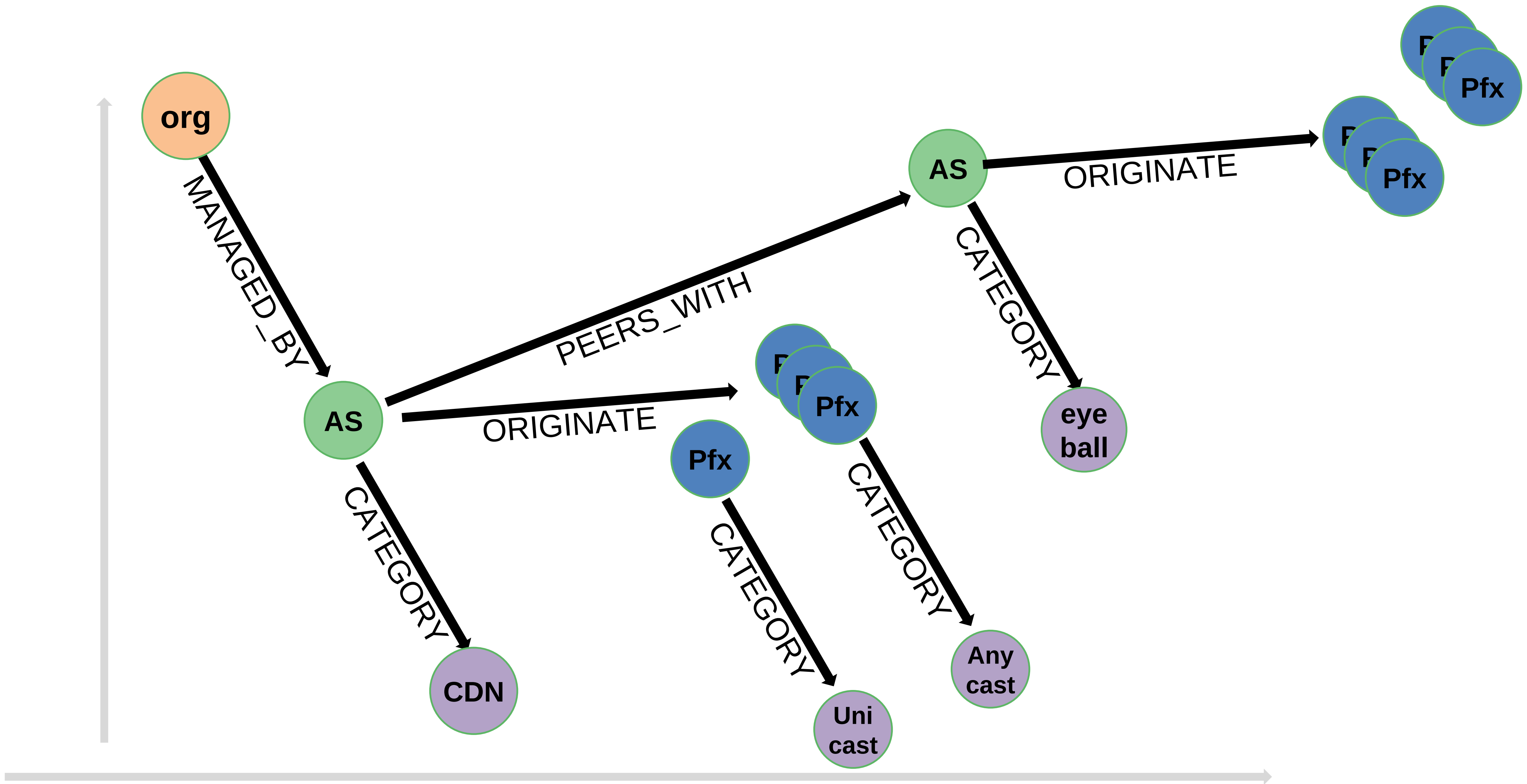
Knowledge Graph Embedding

- Project nodes and relationships in **low dimensional space** and **conserve "data meaning"**

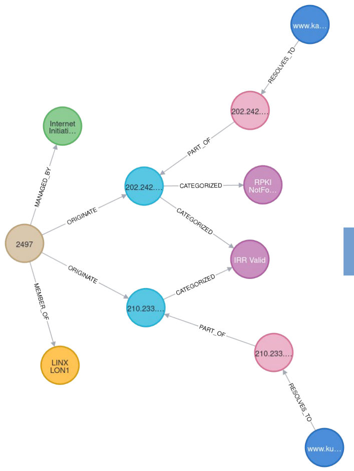




Knowledge Graph Embedding



Knowledge Graph Embedding: Framework



Facts
(head, relation, tail)

AS2497, MANAGED_BY, IIJ
AS2497, MEMBER_OF, LINX
AS2497, ORIGINATE, 202.242.16.0/23
....

Negative Generation

Scoring Layer

$f(h,r,t) \rightarrow R$

Loss Function

Optimizer
(e.g. SGD, Adam)

AS2497	IIJ	LINX	202.242.16.0/23	MANAGED_BY	...
0.173	0.294	0.731	0.438	0.284	...
0.624	0.817	0.284	0.127	0.678	...
0.957	0.463	0.619	0.982	0.566	...
0.382	0.052	0.407	0.365	0.811	...
0.741	0.678	0.852	0.704	0.624	...
0.059	0.156	0.095	0.519	0.095	...
0.811	0.901	0.566	0.876	0.982	...

Knowledge Graph Embedding: Scoring function

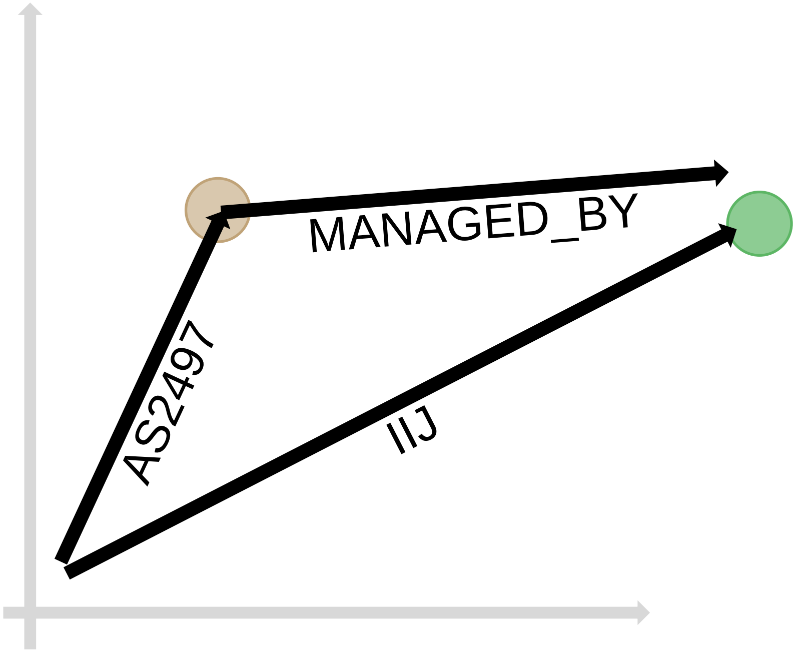
Facts
(head, relation, tail)

AS2497, MANAGED_BY, IIJ
AS2497, MEMBER_OF, LINX
....

Scoring Layer

f(h,r,t) → R

- High score = fact (h, r, t) is probably true
- Example: TransE



Knowledge Graph Embedding: Scoring function

Table 1: A comprehensive summary of models that address complex mapping characteristics of relations within the context of KGE

Model		Embedding Space	Scoring Function	Characteristics	Pros and Cons
Models Based on Relation-Aware Mapping	TransH	Real Vector Space	$\lVert \boldsymbol{h}_r + \boldsymbol{r} - \boldsymbol{t}_r \rVert_{L_1/L_2}$	Relation hyperplane projection	Pros: robust geometric interpretability. Cons: require additional parameters for each relation, which can lead to increased computational costs and potential overfitting.
	TransR		$\lVert \boldsymbol{M}_r \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{M}_r \boldsymbol{t} \rVert_{L_1/L_2}$	Relation space projection	
	STransE		$\lVert \boldsymbol{M}_{r1} \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{M}_{r2} \boldsymbol{t} \rVert_{L_1/L_2}$	Projection of Head and Tail Entities into Different Relation Spaces	
	TransD		$\lVert \theta_h \boldsymbol{M}_{rh} \boldsymbol{h} + \boldsymbol{r} - \theta_h \boldsymbol{M}_{rt} \boldsymbol{t} \rVert_{L_1/L_2}$	Adaptive Sparse Projection Matrix	
	TransF		$(\boldsymbol{h} + \boldsymbol{r})^T \boldsymbol{t} + \boldsymbol{h}^T (\boldsymbol{t} - \boldsymbol{r})$	"Head Entity Vector + Relation Vector" aligned with Tail Entity Vector	
	TransA		$(\lvert \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t} \rvert)^T \boldsymbol{M}_r (\lvert \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t} \rvert)$	Euclidean Distance replaced by Weighted Mahalanobis Distance	
	TransM		$w_r \lVert \boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t} \rVert_{L_1/L_2}$	Reducing Weights for Relations with Complex Mapping Characteristics	
Models Based on Specific Representation Spaces	KG2E	Real Vector Space	$\int_{\boldsymbol{x} \in \mathbb{R}^k} N(\boldsymbol{x}, \boldsymbol{u}_e, \Sigma_e) N(\boldsymbol{x}, \boldsymbol{u}_r, \Sigma_r) d\boldsymbol{x}$	Representing Uncertainty of Entities and Relations in Gaussian Space	Pros: directly modeling complex relation mappings. Cons: a more complex representation space compared to real vector spaces.
	ManifoldE	Real Vector Space	$\lVert MF(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) - D_r^2 \rVert$	Head Entity and Relation as Center of Sphere, Tail Entity within the Sphere	
	TorusE	Real Vector Space	$E_{L1}(h, r, t) = 2 d_{L1}([\boldsymbol{h}] + [\boldsymbol{r}], [\boldsymbol{t}])$	Consistent Difference of Embeddings on Compact Lie Group Torus for Different Tail Entities	
Tensor Decomposition-Based Models	RESCAL	Real Vector Space	$\boldsymbol{h}^T \boldsymbol{M}_r \boldsymbol{t}$	Decomposing third-order tensors into low-dimensional matrices and tensor multiplication	Pros: directly capture complex relation mappings. Cons: come with a higher parameter count.
	DistMult		$\boldsymbol{h}^T diag(\boldsymbol{r}) \boldsymbol{t}$	Representing Each Relation as a Diagonal Matrix to Reduce Parameters	
	TuckER		$w \times_1 \boldsymbol{h} \times_2 \boldsymbol{r} \times_3 \boldsymbol{t}$	Reducing Parameters through Shared Weight Mechanism	
Neural Network-Based Models	SME	Real Vector Space	$g_u(h, r)^T g_v(t, r)$	Using Fully Connected Neural Networks to Model Interactions between Head, Tail Entities, and Relations for Triple Scoring	Pros: directly modeling complex relation mappings. Cons: high computational complexity.
	NTN		$\boldsymbol{r}^T \tanh(\boldsymbol{h}^T \boldsymbol{M}_r \boldsymbol{t} + \boldsymbol{M}_r^1 \boldsymbol{h} + \boldsymbol{M}_r^2 \boldsymbol{t} + \boldsymbol{b}_r)$	Encoding Interactions between Head, Tail Entities, and Relations in Fully Connected Neural Networks	
	ConvE		$\sigma(vec(\sigma(([\boldsymbol{M}_h; \boldsymbol{M}_r] * \omega))\boldsymbol{W}))\boldsymbol{t}$	Reconstructing head entity and relation embeddings into 2D matrices for convolution operations	
	ConvKB		$\mathrm{concat}(Relu([\boldsymbol{h}; \boldsymbol{r}; \boldsymbol{t}] * \omega))\boldsymbol{w}$	Directly stacking head, tail entity, and relation vectors to form 2D matrices for convolution operations	
	CapsE		$\lVert capsnet(Relu([\boldsymbol{h}; \boldsymbol{r}; \boldsymbol{t}] * \omega)) \rVert$	Using capsule networks post-convolution to capture entries along the same dimension of feature vectors	
	InteractE		$\sigma(vec(Relu(([\boldsymbol{M}_h; \boldsymbol{M}_r] * \omega))\boldsymbol{W}))\boldsymbol{t}$	Reconstructing 2D matrices for convolution operations using direct stacking, row-wise circular convolution, and element-wise interaction	

→ Many choices! with different properties

IYP embeddings in practice

- **Computing IYP embeddings:**
  1. Extract sub-graph of interest
  2. Add explicit node types (type-aware KG)
  3. Train a model

- **Downstream tasks:**
  - Peering recommendation (similar to Loqman IMC'24 metAScritic)
  - IXP membership detection
    (Peeringdb embeddings, ground truth PCH and AliceLG)
  - AS classification (e.g personnal AS)
  - VP placement and selection (cluster AS or VPs)

Preliminary experiments

- **Training data:**

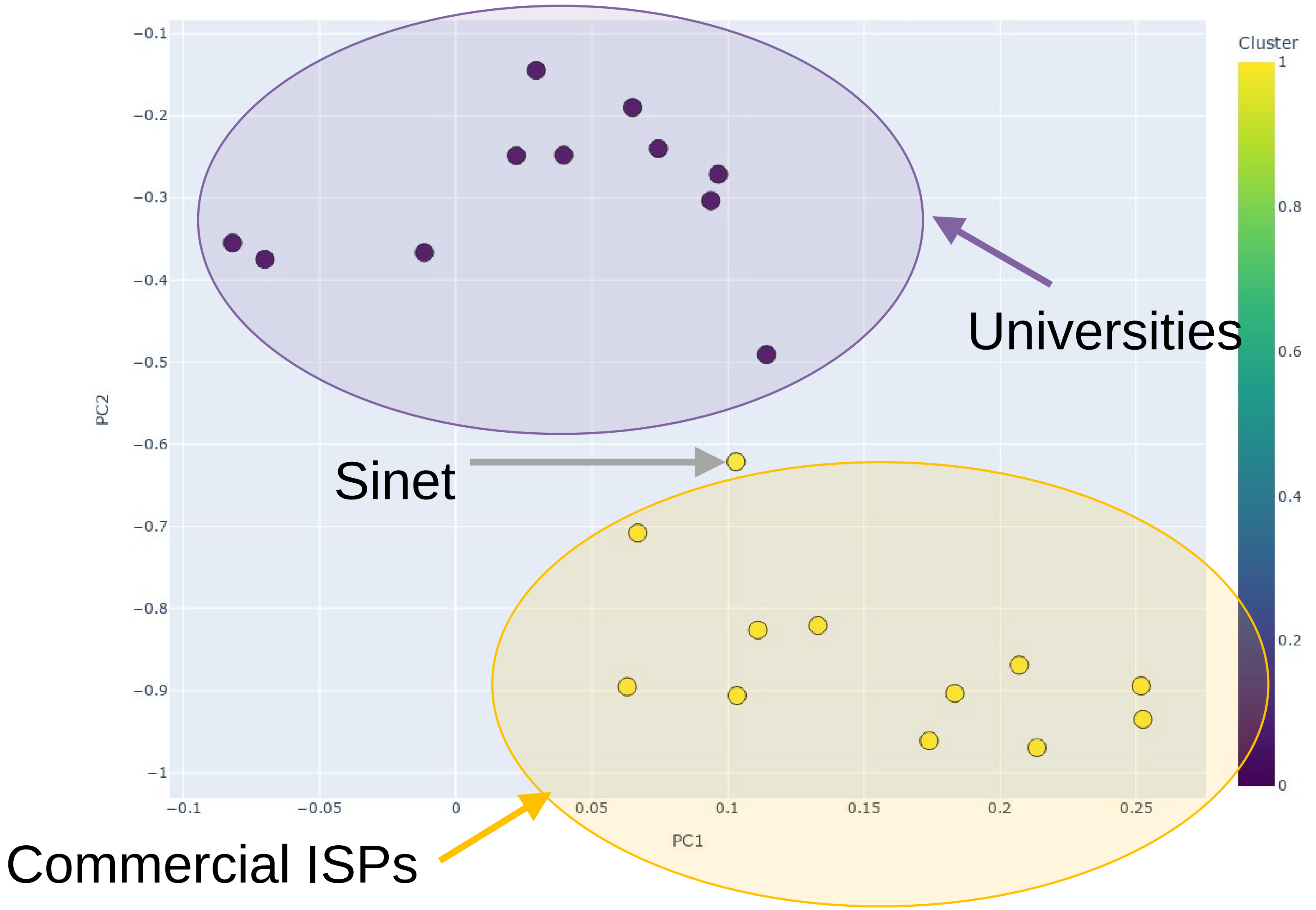
  AS links (BGP) + PeeringDB + Delegated stats

- **Models:**

  TransH, RotatE, AutoSF (20 and 200 dimensions)

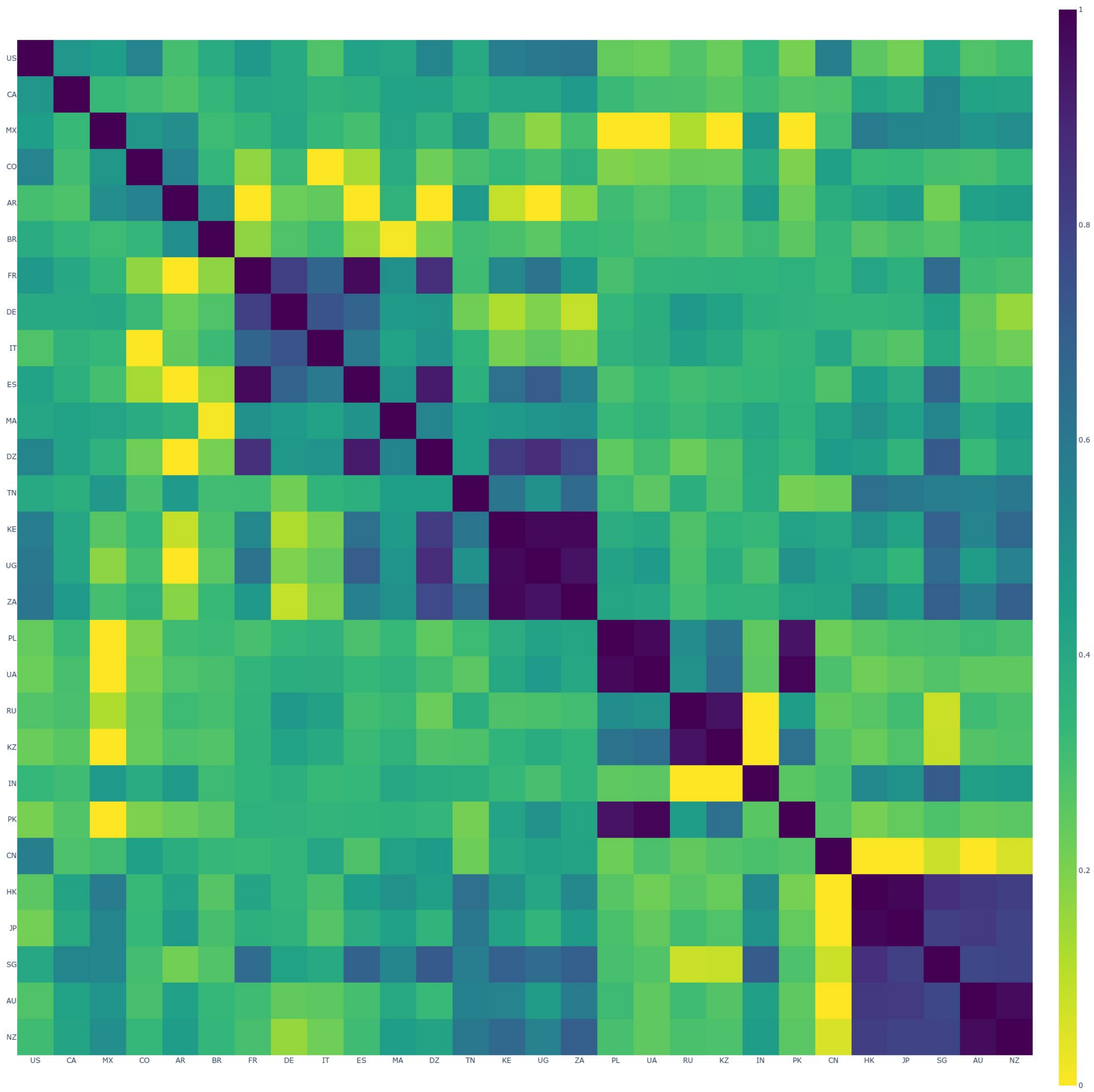
Example Embeddings

Japanese networks in 2D

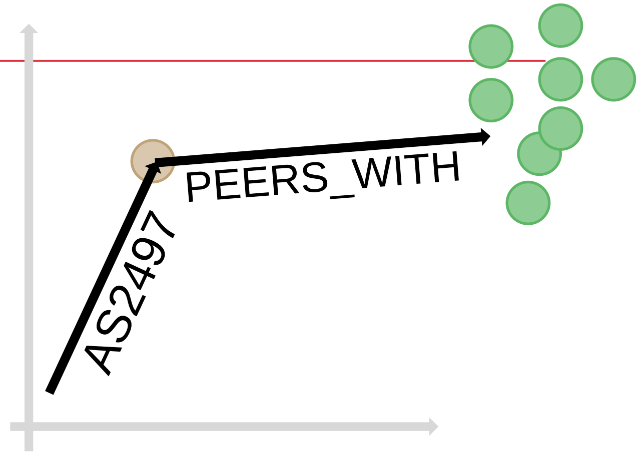


Example: Clustering

- Country distances



Example: Prediction



Peering recommendations for IIJ (AS2497)

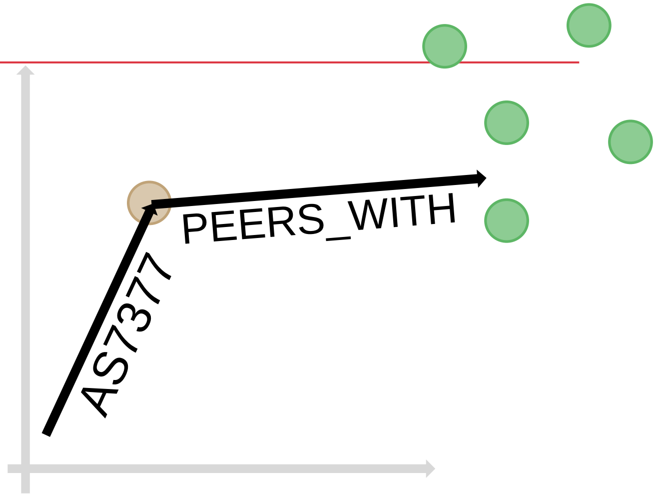
score	in_training	node_label	
-1.369303	True	2518	Biglobe
-1.523403	True	17676	Softbank
-1.562442	True	2516	KDDI
-1.585151	False	2497	IIJ (!)
-1.606323	True	4777	APNIC
-1.608234	True	9370	Sakura
-1.652315	True	8966	Etisalat
-1.653842	False	150369	TelHi
-1.675913	True	2500	WIDE
-1.686394	False	59105	Home NOC
-1.701167	True	4637	Telstra
-1.717189	False	138997	Eons Data
-1.717401	True	15412	Flagtel
-1.736476	True	7473	Singtel
-1.739194	True	55900	GLBB
-1.753962	True	17961	Mitene
-1.759668	False	17534	NSK
-1.768300	True	25152	K-root
-1.771381	False	5580	GTT Asia
-1.779772	True	9505	TWgate
-1.782672	False	38195	Superloop
-1.785485	True	2519	Vectant
-1.804334	False	17686	Accelia
-1.808694	True	4685	Asahi net
-1.812088	True	7522	STNet
-1.827899	True	7679	QTnet
-1.837492	True	7524	i-TEC
-1.844971	True	41095	IPTP
-1.858876	True	2914	NTT GIN
-1.862129	False	38001	NewMedia

Large Japanese ISPs

Very well connected Japanese networks (300+ neighbours)

Large ISPs and cloud providers in Asia (especially Japan, Singapore, and Hong Kong)

Example: Prediction (cont.)



Peering recommendations for UCSD (AS7377)

```
     score  in_training node_label
-2.322950         True      26397 UCSD
-2.503095         True       2153 CENIC
-2.574491         True        195 SDSC
-2.612609         True       2152 CENIC
-3.007203        False         AS
-3.010085        False       7029 Windstream
-3.056936        False        701 Verizon
-3.072093        False       3356 Lumen
-3.081597        False      11537 Internet2
-3.108830        False      11164 Internet2
-3.136462        False        293 ESnet
-3.144664        False       6079 Astound
-3.168146        False      20115 Charter
-3.176275        False       7018 ATT
-3.186226        False       6461 Zayo
-3.187736        False        101
-3.188335        False      11404
-3.189052        False      22335
-3.190395        False       7922
-3.214052        False      53828
-3.222880        False      22773
-3.227499        False         24
-3.228077        False      30600
-3.231935        False      13536
-3.234946        False      36086
-3.235250        False      14041
-3.242578        False      33667
-3.242678        False       6325
-3.246642        False      11096
-3.2[illegible]7804        False       7377
```

Actual peers

Commercial or Educational ISPs in US

low scores => AS with small number of peerings

What’s next?

- **IYP embedding benchmark**

- **Downstream tasks**

- **Better understanding of the effect of training dataset**

  Different embeddings for different applications?

- ~~**Scaling to the whole graph**~~

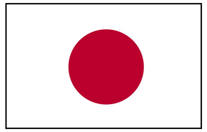
  ~~Try existing methods (e.g. NodePiece)~~

IYP benchmark

- **Motivation:** There is a many ways to make IYP embeddings
- **Goal:** Find the most meaningful embeddings
- **Intuition:** Loss acceptable for outliers, not for expected patterns

AS’s country make sense for stub AS but not for a hypergiant/Tier1

UTokyo



Google ??

Similarly:

Unicast vs. Anycast prefixes

Local vs. global IXPs

IYP benchmark: example

	TransH	RotatE	AutoSF
Dimension 20	Stub tests - hit@1: 0.68 - hit@10: 1.0 - mean entropy 1.95 Hypergiant tests - hit@1: 0.0 - hit@10: 0.08 - mean entropy 1.84	Stub tests - hit@1: 0.66 - hit@10: 0.90 - mean entropy 1.73 Hypergiant tests - hit@1: 0.0 - hit@10: 0.0 - mean entropy 1.80	Stub tests - hit@1: 0.53 - hit@10: 0.99 - mean entropy 1.57 Hypergiant tests - hit@1: 0.0 - hit@10: 0.08 - mean entropy 1.69
Dimension 200	Stub tests - hit@1: 0.89 - hit@10: 0.92 - mean entropy 1.29 Hypergiant tests - hit@1: 0.0 - hit@10: 0.16 - mean entropy 1.83	Stub tests - hit@1: 1.0 - hit@10: 1.0 - mean entropy 1.12 Hypergiant tests - hit@1: 0.0 - hit@10: 0.0 - mean entropy 1.30	Stub tests - hit@1: 0.0 - hit@10: 0.0 - mean entropy 1.84 Hypergiant tests - hit@1: 0.0 - hit@10: 0.0 - mean entropy 1.81

More dimensions not always better!

Perfect score!

IIJ

Internet Initiative Japan

日本のインターネットは1992年、IIJとともにはじまりました。以来、IIJグループはネットワーク社会の基盤をつくり、技術力でその発展を支えてきました。インターネットの未来を想い、新たなイノベーションに挑戦し続けていく。それは、つねに先駆者としてインターネットの可能性を切り拓いてきたIIJの、これからも変わることのない姿勢です。IIJの真ん中のIはイニシアティブ——IIJはいつもはじまりであり、未来です。

Content

- IYP quick overview (KG,
- Going beyond the graph (e.g. distance)

    Machine learning, clustering, prediction,
- KGE 101 (TransE)
- IYP KGE

    extract subgraph

    explicit semantics
- IYP KGE benchmark
- Example KGE with IYP

    Plot japanese ISPs

    Plot countries

Example: Prediction (cont. 2)

more:

- can do ‘reverse’ prediction: ?? COUNTRY JP

- different meaning with different training dataset (e.g. LINE)