At the Intersection of Neuro-Symbolic AI and Federated Learning

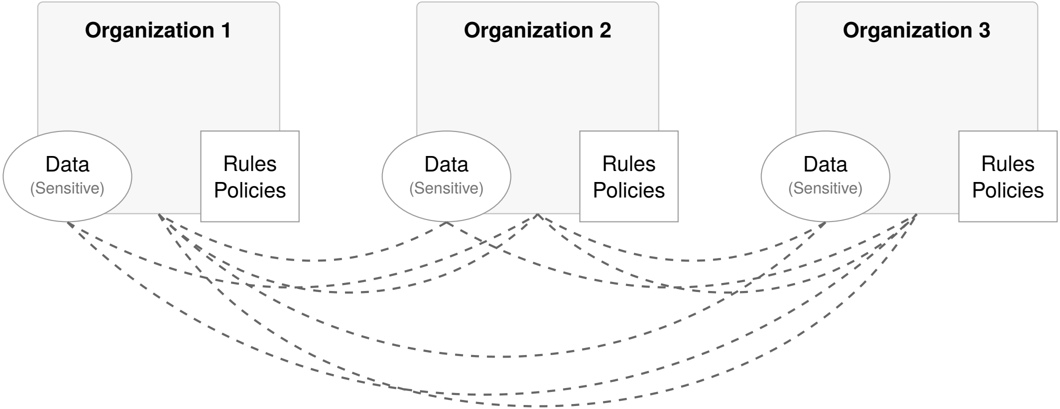
Lou Elah Süsslin

Seminar in AI
Winter term 2025/26
TU Wien

Overview

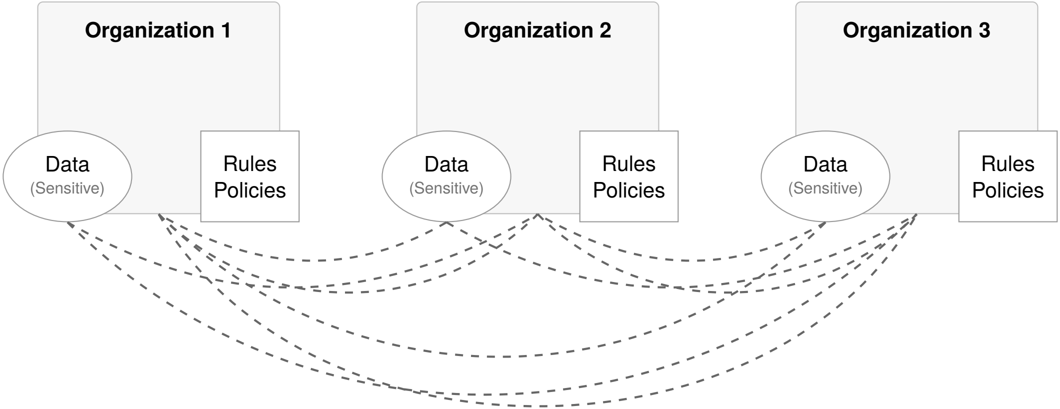
A real scenario, a real problem...

A real scenario, a real problem...



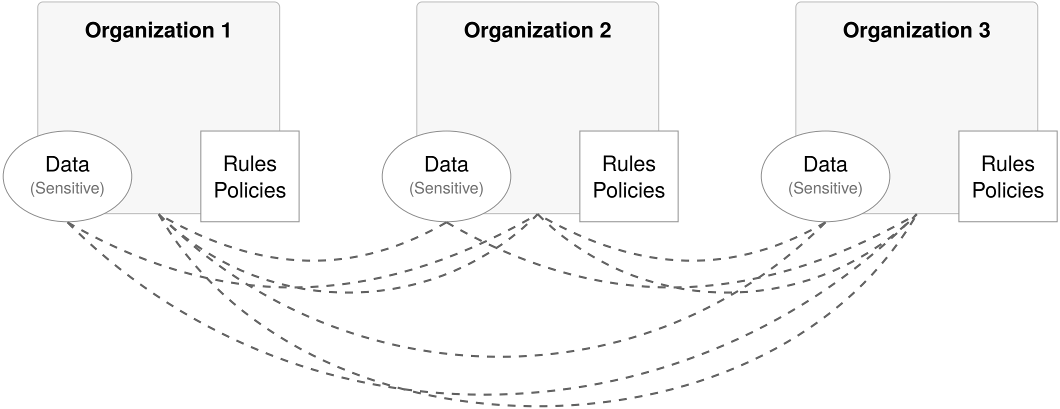
There would be so much to learn when collaborating and sharing data!

A real scenario, a real problem...



There **would be** so much to learn when collaborating and sharing data!

A real scenario, a real problem...



There **would be** so much to learn when collaborating and sharing data!

Problem:
Cannot share sensible data

Conflict of wants

want to benefit from what others know

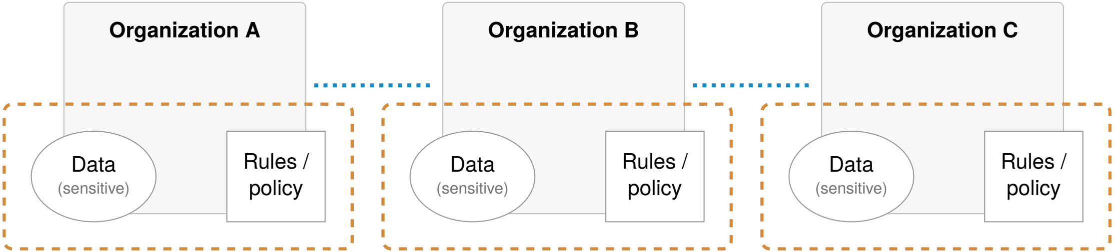
don’t want to hand over own raw data

want powerful models

but I want to adhere to certain rules (or have to)

What would be helpful

share which rules seem to work where



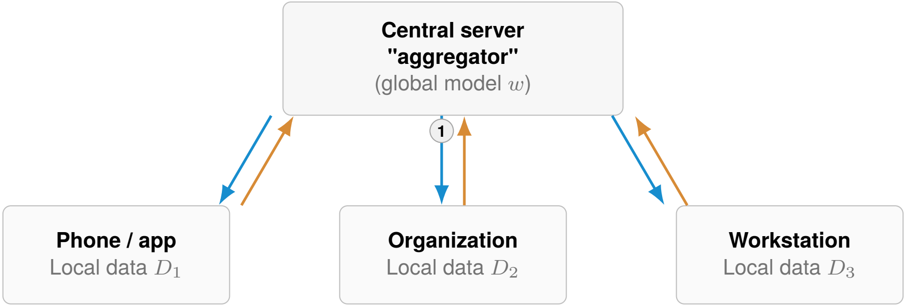
keeps data and rules local

Federated Learning Essentials

Architecture

One round:
1 Broadcast w^t
2 Local train (SGD on D_i)
3 Upload update
4 Aggregate (FedAvg) $\rightarrow w^{t+1}$

One shared model, trained collaboratively; raw data stays local.

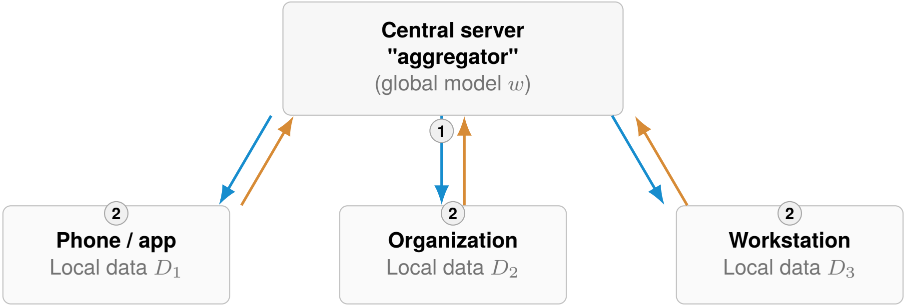


[Bharati et al. 2022]

Architecture

One round:
1 Broadcast w^t
2 Local train (SGD on D_i)
$w \leftarrow w - \eta \nabla L_i(w)$
3 Upload update
4 Aggregate (FedAvg) $\rightarrow w^{t+1}$

One shared model, trained collaboratively; raw data stays local.

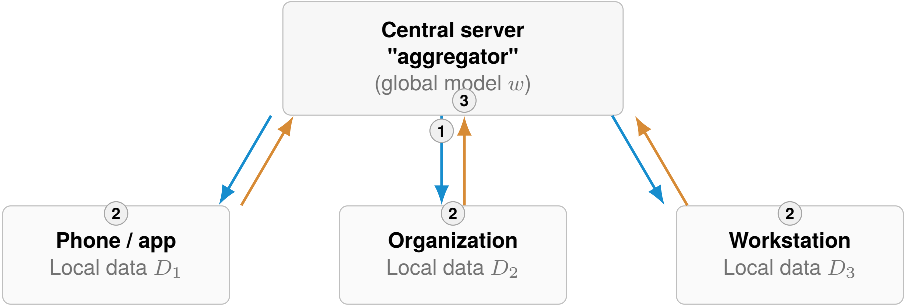


[Bharati et al. 2022]

Architecture

One round:
1 Broadcast w^t
2 Local train (SGD on D_i)
$w \leftarrow w - \eta \nabla L_i(w)$
3 Upload update
4 Aggregate (FedAvg) $\rightarrow w^{t+1}$

One shared model, trained collaboratively; raw data stays local.

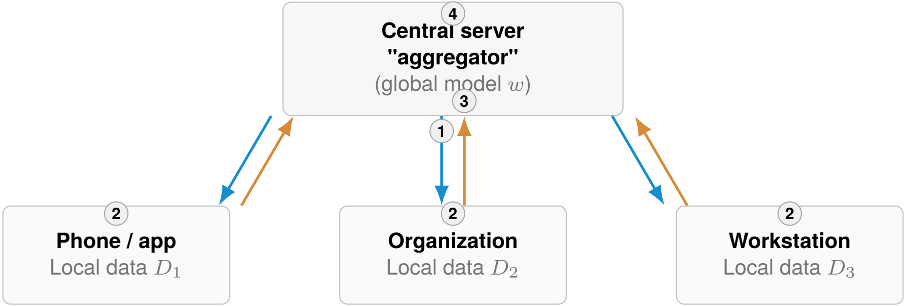


[Bharati et al. 2022]

Architecture

One shared model, trained collaboratively; raw data stays local.

One round:
1 Broadcast w^t
2 Local train (SGD on D_i)
$w \leftarrow w - \eta \nabla L_i(w)$
3 Upload update
4 Aggregate (FedAvg) $\rightarrow w^{t+1}$



[Bharati et al. 2022]

Objective and Aggregation

Global objective (one shared model)

$$w^\star = \arg\min_w \sum_{i=1}^{N} f_i \, L_i(w) \qquad \text{(often } f_i \propto |D_i|\text{)}$$

(Think: choose w to *minimize* the federation's weighted average loss.)

One round (what the loop implements)

Clients (local SGD, τ steps): $w \leftarrow w - \eta \nabla \ell(w; b), \;\; b \subset D_i$ (repeat) $\Rightarrow \; w_i$

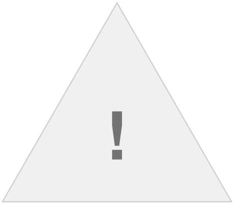
Server (FedAvg): $$w^{t+1} = \sum_{i=1}^{N} f_i \, w_i$$

(Subtracting the gradient = "step downhill" to reduce loss; $\nabla \ell(w; b)$ is a mini-batch estimate of $\nabla L_i(w)$.)

Legend: i client; N clients; t round; D_i local dataset; b mini-batch; w model parameters; w^t global params at round t; w_i client params after local updates; $L_i(w)$ = loss averaged over D_i; $\ell(w; b)$ = loss on mini-batch b (SGD estimate); η step size; τ local steps; f_i aggregation weight.

[Bharati et al. 2022]

Caveat about privacy

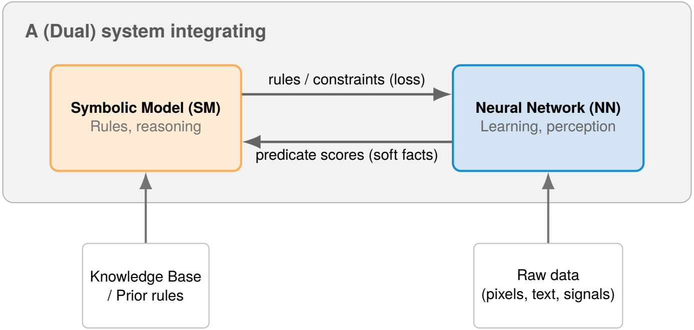


- Federated Learning is **privacy-enhancing by design**
- It is **not a formal privacy guarantee**

Neuro-Symbolic AI Essentials

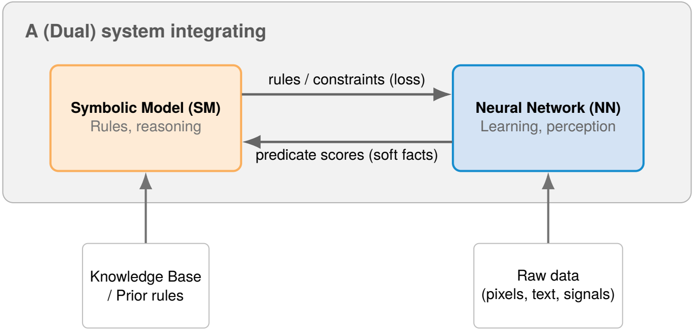
Neural + symbolic parts in one system

[Garcez & Lamb 2023; Wang et al. 2024]



Neural + symbolic parts in one system

[Garcez & Lamb 2023; Wang et al. 2024]



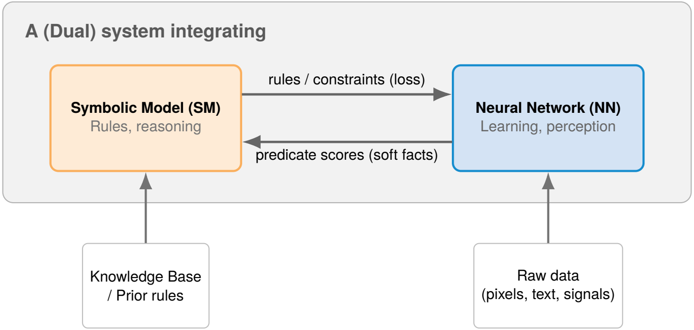
Medical Micro-example p=patient, d=drug

NN output: $P(\text{FluidRetention}(p)) = 0.8,\ P(\text{PoorKidney}(p)) = 0.7$

SM rule: $\text{FluidRetention}(p) \wedge \text{PoorKidney}(p) \to \text{DontPrescribe}(p, d)$

Neural + symbolic parts in one system

[Garcez & Lamb 2023; Wang et al. 2024]



Medical Micro-example p=patient, d=drug

NN output: $P(\text{FluidRetention}(p)) = 0.8,\; P(\text{PoorKidney}(p)) = 0.7$

SM rule: $\text{FluidRetention}(p) \wedge \text{PoorKidney}(p) \rightarrow \text{DontPrescribe}(p, d)$

Tight integration: add rule-violation penalty to NN loss.

Two interaction architectures (loose vs. tight)

Architecture 1: Modular pipeline (loose integration)

- NN: perception / scoring on raw data
- SM: discrete reasoning / planning with rules

Micro-example:

- ▷ NN outputs: $P(\text{FluidRetention}(p))$, $P(\text{PoorKidney}(p))$
- ▷ SM applies: FluidRetention $\land$ PoorKidney $\rightarrow$ DontPrescribe(p, d)

$\Rightarrow$ discrete decision + explanation

+ reasoning is transparent
− discrete step $\rightarrow$ hard to train end-to-end

[Garcez & Lamb 2023]

Two interaction architectures (loose vs. tight)

Architecture 1: Modular pipeline (loose integration)

- NN: perception / scoring on raw data
- SM: discrete reasoning / planning with rules

Micro-example:

- ▷ NN outputs: $P(\text{FluidRetention}(p))$, $P(\text{PoorKidney}(p))$
- ▷ SM applies: FluidRetention $\wedge$ PoorKidney $\rightarrow$ DontPrescribe(p, d)

$\Rightarrow$ discrete decision + explanation

$+$ reasoning is transparent

$-$ discrete step $\rightarrow$ hard to train end-to-end

Architecture 2: Differentiable loss (tight integration)

- NN predicts *soft* predicate scores (truth degrees)
- Rule template $\Rightarrow$ fuzzy/Gödel relaxation $\Rightarrow$ penalty $L_{\text{rule}}(w)$ computed from NN scores

$$\min_{w} \left(L_{\text{task}}(w) + \lambda L_{\text{rule}}(w)\right)$$

Micro-example (rule penalty):

- ▷ NN scores: $P(\text{FluidRetention})$, $P(\text{PoorKidney})$, $P(\text{DontPrescribe})$

$\Rightarrow$ antecedent high + consequent low $\Rightarrow$ L_{rule} increases

Legend: L_{task} supervised data loss (labels); L_{rule} rule-violation penalty (from rule template + NN scores); λ trade-off weight.

[Garcez & Lamb 2023]

Two interaction architectures (loose vs. tight)

Architecture 1: Modular pipeline (loose integration)

- NN: perception / scoring on raw data
- SM: discrete reasoning / planning with rules

Micro-example:

- ▷ NN outputs: $P(\text{FluidRetention}(p))$, $P(\text{PoorKidney}(p))$
- ▷ SM applies: FluidRetention $\wedge$ PoorKidney $\rightarrow$ DontPrescribe(p, d)
- $\Rightarrow$ discrete decision + explanation

- $+$ reasoning is transparent
- $-$ discrete step $\rightarrow$ hard to train end-to-end

Architecture 2: Differentiable loss (tight integration)

- NN predicts *soft* predicate scores (truth degrees)
- Rule template $\Rightarrow$ fuzzy/Gödel relaxation $\Rightarrow$ penalty $L_{\text{rule}}(w)$ computed from NN scores

$$\min_{w} \left(L_{\text{task}}(w) + \lambda L_{\text{rule}}(w) \right)$$

Micro-example (rule penalty):

- ▷ NN scores: $P(\text{FluidRetention})$, $P(\text{PoorKidney})$, $P(\text{DontPrescribe})$
- $\Rightarrow$ antecedent high + consequent low $\Rightarrow$ L_{rule} increases

Legend: L_{task} supervised data loss (labels); L_{rule} rule-violation penalty (from rule template + NN scores); λ trade-off weight.

Why we care: FedNSL follows Architecture 2 (tight coupling) and federates *rule-level beliefs*.

[Garcez & Lamb 2023]

Returning to the intersection

[Xing et al. 2024]

What is still missing?

Until now: two partial solutions

- **Federated learning:** black-box; no raw data shared (but rules are implicit)
- **Pure symbolic rules:** transparent/auditable (but brittle on messy data)

[Xing et al. 2024]

What is still missing?

Until now: two partial solutions

- **Federated learning:** black-box; no raw data shared (but rules are implicit)
- **Pure symbolic rules:** transparent/auditable (but brittle on messy data)

The gap (for our original scenario)

- We need **privacy-preserving collaboration** *and* **rule-level structure**
- $\Rightarrow$ not just exchanging weights, and not only hand-written rules

[Xing et al. 2024]

What is still missing?

Until now: two partial solutions

- **Federated learning:** black-box; no raw data shared (but rules are implicit)
- **Pure symbolic rules:** transparent/auditable (but brittle on messy data)

The gap (for our original scenario)

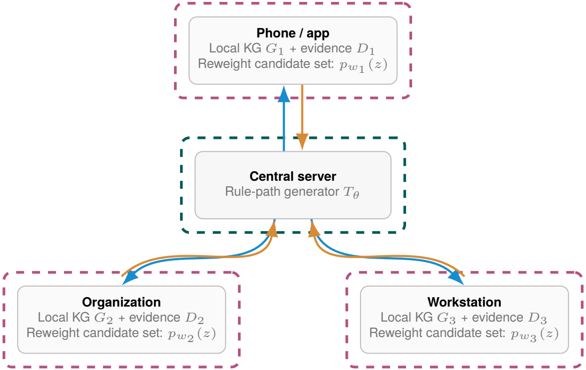
- We need **privacy-preserving collaboration** *and* **rule-level structure**
- $\Rightarrow$ not just exchanging weights, and not only hand-written rules

Transition: change the shared payload

FL: Δw (parameter updates) $\longrightarrow$ **FedNSL:** $q(r)$ (beliefs over candidate rules)

[Xing et al. 2024]

Same Federated Learning loop, new shared object



(1) Broadcast: **bounded candidate set** $b_{1:J}$
(3) Upload: posterior

Conventions: G_i = client's private KG (entities + relations + triples); D_i = local evidence from G_i (observed triples + local link-prediction queries).

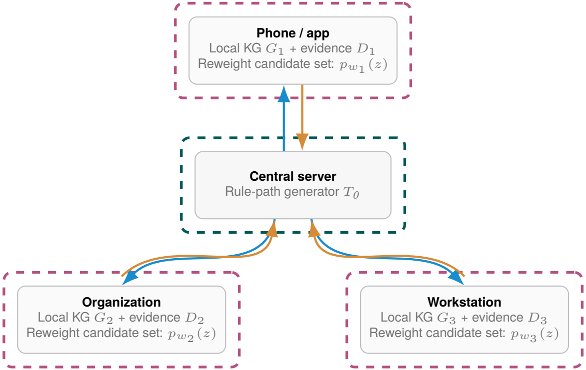
SERVER/AGGREGATOR

- **(1) Broadcast:** propose bounded path-bodies $b_{1:J}$ via T_θ and send prior $p_\theta(z)$
  - $z \in \{1, \dots, J\}$ indexes one b_z.
- **(4) Aggregate:** combine client posteriors and update θ

CLIENTS

- Raw G_i and D_i stay local (privacy)
- **(2) Local update:** reweight the candidate set on local evidence $\rightarrow p_{w_i}(z)$
- **(3) Upload:** send back posterior over candidates $p_{w_i}(z)$ (or a compact summary)

Same Federated Learning loop, new shared object



(1) Broadcast: **bounded candidate set** $b_{1:J}$ + prior $p_\theta(z)$
(3) Upload: posterior $p_{w_i}(z)$

Conventions: G_i = client's private KG (entities + relations + triples);
D_i = local evidence from G_i (observed triples + local link-prediction queries).

SERVER/AGGREGATOR

- **(1) Broadcast:** propose bounded path-bodies $b_{1:J}$ via T_θ and send prior $p_\theta(z)$
  - $z \in \{1, \ldots, J\}$ indexes one b_z.
- **(4) Aggregate:** combine client posteriors and update θ

CLIENTS

- Raw G_i and D_i stay local (privacy)
- **(2) Local update:** reweight the candidate set on local evidence $\to p_{w_i}(z)$
- **(3) Upload:** send back posterior over candidates $p_{w_i}(z)$ (or a compact summary)

FedNSL core objects (Knowledge Graph (KG) completion)

Local symbolic data

Client i holds a **private** KG G_i with entities + typed relations, stored as triples $\langle h, r, t \rangle$.

Example: $\langle \text{TU Wien},\ \text{locatedIn},\ \text{Vienna} \rangle$

FedNSL core objects (Knowledge Graph (KG) completion)

Local symbolic data	Client i holds a **private** KG G_i with entities + typed relations, stored as triples $\langle h, r, t \rangle$. Example: $\langle$TU Wien, locatedIn, Vienna$\rangle$
Prediction task	**KG completion / link prediction:** query $q_i = \langle h, ?, t \rangle \Rightarrow$ predict missing relation $a_i = r_{\text{head}} \in \mathcal{R}$. (multi-class classification over relation labels $\mathcal{R}$)

FedNSL core objects (Knowledge Graph (KG) completion)

Local symbolic data

Client i holds a **private** KG G_i with entities + typed relations, stored as triples $\langle h, r, t \rangle$.

Example: $\langle$TU Wien, locatedIn, Vienna$\rangle$

Prediction task

KG completion / link prediction:
query $q_i = \langle h, ?, t \rangle \Rightarrow$ predict missing relation $a_i = r_{\text{head}} \in \mathcal{R}$.

(multi-class classification over relation labels $\mathcal{R}$)

Rule (path template)

A rule = **(path body $\Rightarrow$ head relation)**: a multi-hop path supports $r_{\text{head}}(h, t)$.

Body: $b = (r_1, \ldots, r_\ell)$ with $r_j \in \mathcal{R}$ along $h \to x_1 \to \cdots \to t$.

Form: $r_1(h, x_1) \wedge r_2(x_1, x_2) \wedge \cdots \wedge r_\ell(x_{\ell-1}, t) \Rightarrow r_{\text{head}}(h, t)$.

Variable sharing: shared $x_1, \ldots, x_{\ell-1}$ link the atoms into one chain

Example (2-hop chain): worksAt$(x, o) \wedge$ locatedIn$(o, y) \Rightarrow$ livesIn(x, y).

FedNSL core objects (Knowledge Graph (KG) completion)

Local symbolic data	Client i holds a **private** KG G_i with entities + typed relations, stored as triples $\langle h, r, t\rangle$. Example: $\langle$TU Wien, locatedIn, Vienna$\rangle$
Prediction task	**KG completion / link prediction:** query $q_i = \langle h, ?, t\rangle \Rightarrow$ predict missing relation $a_i = r_{\text{head}} \in \mathcal{R}$. (multi-class classification over relation labels $\mathcal{R}$)
Rule (path template)	A rule = **(path body $\Rightarrow$ head relation)**: a multi-hop path supports $r_{\text{head}}(h, t)$. **Body:** $b = (r_1, \ldots, r_\ell)$ with $r_j \in \mathcal{R}$ along $h \to x_1 \to \cdots \to t$. **Form:** $r_1(h, x_1) \wedge r_2(x_1, x_2) \wedge \cdots \wedge r_\ell(x_{\ell-1}, t) \Rightarrow r_{\text{head}}(h, t)$. **Variable sharing:** shared $x_1, \ldots, x_{\ell-1}$ link the atoms into one chain Example (2-hop chain): worksAt$(x, o) \wedge$ locatedIn$(o, y) \Rightarrow$ livesIn(x, y).
What "weights" mean here	**Server:** θ for generator $T_\theta(b \mid r_{\text{head}})$ (proposes a bounded candidate set $b_{1:J}$). **Client:** w_{ij} = belief weight for candidate j under private G_i. (communicate beliefs over candidate path-bodies, not raw triples)

Legend: $G_i = (\mathcal{E}_i, \mathcal{R}, \mathcal{T}_i)$ local KG (entities, relations, triples); $\langle h, r, t\rangle \in \mathcal{T}_i$ triple; q_i query; r_{head} predicted relation; $b = (r_1, \ldots, r_\ell)$ body/path; θ generator params; w_{ij} client belief.

Client objective (local fit + global alignment)

$$L(w_i, \theta; z_i, \bar{z}) = \underbrace{\ell_i(w_i, \theta; z_i)}_{\substack{\text{Local fit: expected task loss} \\ \text{on client evidence } D_i \text{ (derived from } G_i)}}$$

Details

- **Local fit term ℓ_i:** expected task loss under the client's posterior over candidate indices:

$$\ell_i(w_i, \theta) = \mathbb{E}_{z_i \sim p_{w_i}}\left[\mathcal{L}_{\text{task}}(\text{pred}(\theta, z_i), \text{ labels in } D_i)\right]$$

where D_i = triples + link-prediction queries constructed from G_i

Client objective (local fit + global alignment)

$$L(w_i,\,\theta;\,z_i,\,\bar{z}) = \underbrace{\ell_i(w_i,\,\theta;\,z_i)}_{\substack{\text{Local fit: expected task loss} \\ \text{on client evidence } D_i \text{ (derived from } G_i)}} + \underbrace{\lambda}_{\substack{\text{trade-off weight} \\ \lambda \geq 0}} \underbrace{D_{KL}\Big(\underbrace{p_{w_i}(z_i)}_{\substack{\text{client} \\ \text{posterior}}} \,\Big\|\, \underbrace{p_\theta(\bar{z})}_{\substack{\text{global} \\ \text{prior}}}\Big)}_{\text{alignment / mismatch penalty}}$$

Details

- **Local fit term ℓ_i:** expected task loss under the client's posterior over candidate indices:

$$\ell_i(w_i, \theta) = \mathbb{E}_{z_i \sim p_{w_i}}\Big[\mathcal{L}_{\text{task}}(\text{pred}(\theta, z_i),\ \text{labels in } D_i)\Big]$$

where D_i = triples + link-prediction queries constructed from G_i

- **Alignment term (*Kullback–Leibler divergence* (KL) + λ):**

$$D_{KL}(p\|q) = \sum_z p(z) \log \frac{p(z)}{q(z)} \geq 0, \qquad \lambda \text{ controls how strongly we prefer global coherence}$$

* "||" means "relative to", not a norm.

Client objective (local fit + global alignment)

$$L(w_i,\, \theta;\, z_i,\, \bar{z}) = \underbrace{\ell_i(w_i,\, \theta;\, z_i)}_{\substack{\text{Local fit: expected task loss} \\ \text{on client evidence } D_i \text{ (derived from } G_i)}} + \underbrace{\lambda}_{\substack{\text{trade-off weight} \\ \lambda \geq 0}} \underbrace{D_{KL}\Big(\underbrace{p_{w_i}(z_i)}_{\substack{\text{client} \\ \text{posterior}}} \,\Big\|\, \underbrace{p_\theta(\bar{z})}_{\substack{\text{global} \\ \text{prior}}} \Big)}_{\text{alignment / mismatch penalty}}$$

Used in (2) local update (client updates w_i) and (4) aggregation (server updates θ).

Details

- **Local fit term ℓ_i:** expected task loss under the client's posterior over candidate indices:

$$\ell_i(w_i, \theta) = \mathbb{E}_{z_i \sim p_{w_i}} \left[\mathcal{L}_{\text{task}}(\text{pred}(\theta, z_i), \text{ labels in } D_i) \right]$$

where D_i = triples + link-prediction queries constructed from G_i

- **Alignment term (*Kullback–Leibler divergence* (KL) + λ):**

$$D_{KL}(p \| q) = \sum_z p(z) \log \frac{p(z)}{q(z)} \geq 0, \qquad \lambda \text{ controls how strongly we prefer global coherence}$$

* "||" means "relative to", not a norm.

Server update (refining the generator T_θ)

Input
(from (3) client uploads, many clients in round k)

$\{p_{w_i}(z)\}_{i \in C_k}$ (where C_k = clients in round k)
(posterior belief over candidate indices z, i.e., over proposed path-bodies b_z)

1. Aggregate

$\tilde{p}_k(z) = \text{Agg}_{i \in C_k} \, p_{w_i}(z)$ (e.g., mean or data-weighted mean)

2. Build training pairs

1. Choose path-bodies with high support
   - Sampling: $z \sim \tilde{p}_k(z)$
   - (Alternative: Top-J: take J indices with largest $\tilde{p}_k(z)$)
2. Form training batch
   - $S_k = \{(r_{\text{head}}, b = b_z)\}$

3. Train generator (update θ)

$$\theta_{k+1} \leftarrow \arg\max_\theta \sum_{(r,b) \in S_k} \log T_\theta(b \mid r) \quad \text{[max log-prob.]}$$
$$= \arg\min_\theta \sum_{(r,b) \in S_k} - \log T_\theta(b \mid r) \quad \text{[cross-entropy]}$$

4. Output next-round prior

Update prior for the next round: $T_{\theta_{k+1}}(\cdot \mid r_{\text{head}})$

Empirical setup — cross-visible unseen-type test

Stage 1 (toy proxy): not KG completion. Replace explicit path-bodies b by a **hidden type label** $z \in \{A, B, C\}$, standing in for different **families of rule-patterns**.

Empirical setup — cross-visible unseen-type test

Stage 1 (toy proxy): not KG completion. Replace explicit path-bodies b by a **hidden type label** $z \in \{A, B, C\}$, standing in for different **families of rule-patterns**.

Cross-visible split (simplified):

- 3 hidden types: $z \in \{A, B, C\}$
- Client 1 trains on **A & B** only
- Client 2 trains on **B & C** only
- **Test:** Client 1 is tested on **C** (**unseen locally**)

Key question (unseen-type transfer)

Can Client 1 succeed on unseen C **via federation** (only belief summaries), without local C-data?

Cross-visible anchor: C is present at Client 2; overlap on B keeps the type labels aligned.

Empirical setup — cross-visible unseen-type test

Stage 1 (toy proxy): not KG completion. Replace explicit path-bodies b by a **hidden type label** $z \in \{A, B, C\}$, standing in for different **families of rule-patterns**.

Cross-visible split (simplified):

- 3 hidden types: $z \in \{A, B, C\}$
- Client 1 trains on **A & B** only
- Client 2 trains on **B & C** only
- **Test:** Client 1 is tested on **C** (**unseen locally**)

Key question (unseen-type transfer)

Can Client 1 succeed on unseen C **via federation** (only belief summaries), without local C-data?

Cross-visible anchor: C is present at Client 2; overlap on B keeps the type labels aligned.

What the proxy isolates (mechanism)

- **Shared object:** common z-space across clients
- **Belief exchange:** upload $p_{w_i}(z)$, not data
- **KL tether:** keeps client beliefs compatible with a global prior

FedNSL mapping in the proxy

Server: maintains/updates global belief $p_\theta(z)$ (prior)
Clients: infer local belief $p_{w_i}(z)$ from partial data
KL: discourages purely-local solutions that conflict with the global belief

Empirical setup — cross-visible unseen-type test

Stage 1 (toy proxy): not KG completion. Replace explicit path-bodies b by a **hidden type label** $z \in \{A, B, C\}$, standing in for different **families of rule-patterns**.

Cross-visible split (simplified):

- 3 hidden types: $z \in \{A, B, C\}$
- Client 1 trains on **A & B** only
- Client 2 trains on **B & C** only
- **Test:** Client 1 is tested on **C** (**unseen locally**)

Key question (unseen-type transfer)

Can Client 1 succeed on unseen C **via federation** (only belief summaries), without local C-data?

Cross-visible anchor: C is present at Client 2; overlap on B keeps the type labels aligned.

Paper detail: synthetic global distribution modeled as a 3-component mixture model.

What the proxy isolates (mechanism)

- **Shared object:** common z-space across clients
- **Belief exchange:** upload $p_{w_i}(z)$, not data
- **KL tether:** keeps client beliefs compatible with a global prior

FedNSL mapping in the proxy

Server: maintains/updates global belief $p_\theta(z)$ (prior)
Clients: infer local belief $p_{w_i}(z)$ from partial data
KL: discourages purely-local solutions that conflict with the global belief

Stage 2 (real task): news-document KG benchmark

Input from documents: entities (typed) + relations between them $\Rightarrow$ triples $\langle h, r, t \rangle$.
Task: link prediction (as earlier). **Federation:** 4 clients, cross-visible seen/unseen split.
Report: F1 across rounds; **KL on/off check**.

Empirical results — transfer + stability

Two-stage evidence

Stage 1 (proxy): unseen-type transfer (accuracy)
Stage 2 (real KG): KG completion performance + KL on/off comparison (F1)

Empirical results — transfer + stability

Two-stage evidence

Stage 1 (proxy): unseen-type transfer (accuracy)
Stage 2 (real KG): KG completion performance + KL on/off comparison (F1)

Stage 1 result: proxy transfer

+29% avg **unseen-test** accuracy (**+17%** train, unbalanced)
→ direct evidence of **unseen-type transfer** in the cross-visible split

Empirical results — transfer + stability

Two-stage evidence

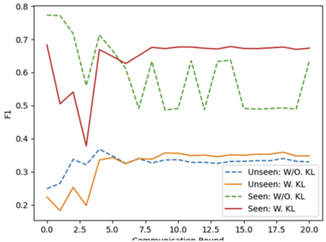
Stage 1 (proxy): unseen-type transfer (accuracy)
Stage 2 (real KG): KG completion performance + KL on/off comparison (F1)

Stage 1 result: proxy transfer

+29% avg **unseen-test** accuracy (**+17%** train, unbalanced)
→ direct evidence of **unseen-type transfer** in the cross-visible split

Stage 2 result: real KG + KL on/off

With KL: stable convergence, higher F1
Without KL: oscillations / lower convergence
→ **KL stabilizes**



Real KG: F1 across rounds (seen/unseen; with vs without KL)
With KL: stable convergence, higher F1
Without KL: oscillations, lower / no convergence

Empirical results — transfer + stability

Two-stage evidence

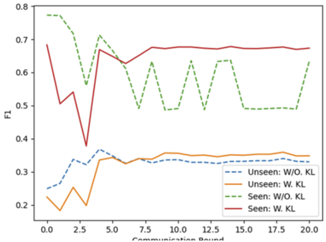
Stage 1 (proxy): unseen-type transfer (accuracy)
Stage 2 (real KG): KG completion performance + KL on/off comparison (F1)

Stage 1 result: proxy transfer

+29% avg **unseen-test** accuracy (**+17%** train, unbalanced)
→ direct evidence of **unseen-type transfer** in the cross-visible split

Stage 2 result: real KG + KL on/off

With KL: stable convergence, higher F1
Without KL: oscillations / lower convergence
→ **KL stabilizes**



Real KG: F1 across rounds (seen/unseen; with vs without KL)
With KL: stable convergence, higher F1
Without KL: oscillations, lower / no convergence

Mechanism: why KL matters

KL is a **tether**: local adaptation stays compatible with the global prior
→ shared signal remains usable across clients

Also: extracted rules are checked against gold logic relations in the dataset.

Takeaway: proxy demonstrates unseen-type transfer; real KG shows KL is the stability mechanism.

Related Research

Logical Reasoning-based eXplainable Federated Learning (LR-XFL): explicit rule aggregation for explainability

[Zhang & Yu 2023]

What it targets

Make FL **interpretable**: aggregate **explicit rules** into a **global rule set** while handling **client conflicts**.

What is communicated (the logic object)

Clients send **symbolic rules** + lightweight **rule statistics**; the server outputs a **global rule set**.

Why it matters for Neuro-Symbolic AI × Federated Learning

Explicit symbolic exchange supports **auditability** and **human-readable governance**, but can be **brittle/costly** under shifting or conflicting client logic.

How it works (only the 2 levers)

- **Resolve conflicts:** choose how to combine rules (e.g. $\wedge$ vs. $\vee$)
- **Weight clients:** prefer higher-quality rules (accuracy / fidelity)

(Keep the AND/OR toy example for the spoken explanation, not the slide.)

FedSTL: personalized FL via *induced temporal properties*

[An et al. 2024]

Setup: same problem, same architecture

All districts predict traffic volume (same sequence NN, e.g. LSTM).

- Historical: many days observed
- Predicted $\hat{Y}_{0:T}$: next T steps

How it works (who does what)

1. Client (district): induce local rule ϕ_i

Mine a pattern from its own history, e.g. "rush hour in [800,1200]".

FedSTL: personalized FL via *induced temporal properties*

[An et al. 2024]

Setup: same problem, same architecture

All districts predict traffic volume (same sequence NN, e.g. LSTM).

- Historical: many days observed
- Predicted $\hat{Y}_{0:T}$: next T steps

Logic object: induced constraint

District induces ϕ_i from historical sequences: **constraint on predicted sequence** $\hat{Y}_{0:T}$ (each entry = volume per time bin).

Example: "Rush hour volume stays in [800, 1200]"

How it works (who does what)

1. Client (district): induce local rule ϕ_i

Mine a pattern from its own history, e.g. "rush hour in [800,1200]".

2. Client: train with a property penalty

$$L_i = L_{\text{pred}} + \lambda\, L_{\text{prop}}(\phi_i, \hat{Y}_{0:T})$$

L_{prop}: how far predictions are from satisfying the rule.

FedSTL: personalized FL via *induced temporal properties*

[An et al. 2024]

Setup: same problem, same architecture

All districts predict traffic volume (same sequence NN, e.g. LSTM).

- Historical: many days observed
- Predicted $\hat{Y}_{0:T}$: next T steps

Logic object: induced constraint

District induces ϕ_i from historical sequences: **constraint on predicted sequence** $\hat{Y}_{0:T}$ (each entry = volume per time bin).

Example: "Rush hour volume stays in [800, 1200]"

What's shared?

No raw data leaves a district. Server keeps K **cluster backbones** (not one global model).
Shared: backbone weights (aggregation in cluster).
Private: local head/adaptor (never uploaded).

How it works (who does what)

1. Client (district): induce local rule ϕ_i

Mine a pattern from its own history, e.g. "rush hour in [800,1200]".

2. Client: train with a property penalty

$$L_i = L_{\text{pred}} + \lambda\, L_{\text{prop}}(\phi_i, \hat{Y}_{0:T})$$

L_{prop}: how far predictions are from satisfying the rule.

3. Server: routing by rule-fit

- Every m rounds: re-route clients (they may switch clusters).
- For each client, score all K backbones on tiny D_i^s using rule-violation $L_{\text{prop}}(\phi_i, \hat{Y})$.
- Route to the lowest score; then run clustered FedAvg until next re-route: local train $\rightarrow$ upload shared layers $\rightarrow$ average within cluster.

Repeated re-routing makes clusters specialize.

Related work: At a glance

One axis: what is the "logic object" in FL?

- **LR-XFL: explicit rules** → a global rule set (interpretable, auditable)
- **FedSTL: temporal constraints over sequences** → constraint-guided personalization
- **FedNSL: beliefs over rule candidates** → transfer when rule **types are missing**

Practical takeaway (when each is attractive)

- Need **human-readable rules / governance** → LR-XFL
- Need **time-series constraints + personalization** → FedSTL
- Need **robust transfer under missing rule families** → FedNSL

These choices expose the same open tension: explicit structure helps governance, but robustness under heterogeneity is hard.

Open challenges at the intersection

1) Discover (knowledge acquisition)

- Learn usable **predicates/rules** from heterogeneous local traces
- Decide rule **correctness** under partial evidence

2) Trust (governance)

- Shared logic can **leak client specifics**; merging can amplify noise
- Need **auditable** server behaviour in high-stakes settings

3) Adapt (dynamics)

- Drift + continual learning: update logic **without forgetting**
- Non-IID participation: clients appear/disappear; evidence is uneven

Takeaway: beyond coordinating logic, we need mechanisms to **discover**, **trust**, and **adapt** it.

Conclusion: key takeaways

1) The problem

Data islands + rules/constraints: learn jointly **without sharing raw data**, while keeping the model **structured** (not only opaque weights).

2) The FedNSL move

Communicate **beliefs over rule candidates** (not raw data; not hard rules). A KL term tethers clients to a global prior $\rightarrow$ transfer when rule types are missing locally.

3) What the evidence shows

Unseen-rule transfer becomes possible; the KL term stabilizes training compared to removing it.

If you remember one thing: in Neuro-Symbolic AI $\times$ Federated Learning, *what you communicate* matters.

Thank you!

Questions?

References

Federated Learning

- Bharati, S., et al. (2022). Federated learning: Applications, challenges and future directions. *arXiv preprint* arXiv:2205.09513v2.
- Daly, K., et al. (2024). Federated learning in practice: ReFederated Learningections and projections. *arXiv preprint* arXiv:2410.08892v1.
- Liu, B., et al. (2024). Recent advances on federated learning: A systematic survey. *Neurocomputing, 597*, 128019.
- Zhu, H., et al. (2021). Federated learning on non-IID data: A survey. *arXiv preprint* arXiv:2106.06843v1.

Neuro-Symbolic AI

- Garcez, A. d'A., & Lamb, L. C. (2023). Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review, 56*(11), 12387–12406.
- Michel–Delétie, C., & Sarker, M. K. (2024). Neuro-Symbolic methods for trustworthy AI: A systematic review. *Neurosymbolic Artificial Intelligence*.
- Nawaz, U., et al. (2025). A review of Neuro-Symbolic AI integrating reasoning and learning for advanced cognitive systems. *Intelligent Systems with Applications, 26*, 200541.
- Wang, W., et al. (2024). Towards data-and knowledge-driven AI: A survey on Neuro-Symbolic computing. *arXiv preprint* arXiv:2210.15889v5.

Neuro-Symbolic AI × Federated Learning frameworks

- An, Z., et al. (2024). Formal logic enabled personalized federated learning through property inference (FedSTL). *AAAI Conference on Artificial Intelligence*.
- Xing, P., et al. (2024). Federated Neuro-Symbolic learning (FedNSL). *Proceedings of the 41st International Conference on Machine Learning (ICML)*.
- Zhang, Y., & Yu, H. (2023). LR-XFederated Learning: Logical reasoning-based explainable federated learning. *arXiv preprint* arXiv:2308.12681.

Backup Slides

Federated Learning: Practical challenges

Client & data heterogeneity (non-IID)

- Each client has its own population / distribution
  $\rightarrow$ local updates can pull w in different directions

Communication & participation constraints

- Limited bandwidth + many rounds; clients may drop out / be unavailable
  $\rightarrow$ protocols trade accuracy vs. communication cost

Privacy is not automatic

- Model updates/gradients can still leak information
  $\rightarrow$ practice: secure aggregation, differential privacy (etc.)

One global model $\neq$ good for everyone

- Average-optimal w can be systematically worse for some clients
  $\rightarrow$ personalization / clustering / local heads

[Bharati et al. 2022]

Neuro-Symbolic AI: current challenges

Knowledge and rules

- Incomplete, noisy, or inconsistent
- Manual engineering time-intensive
- Automatic rule mining unreliable (as of now)

Integration and optimization

- Exact reasoning doesn't scale
- Differentiable relaxations of logic (e.g. fuzzy) blur semantic/guarantees
- Non-trivial optimization issues

Robustness and evaluation

- Brittleness if rules are (partially) wrong
- Hard to test perception + reasoning + Interpretability *together*