# IV-Applications
# Applications of tnreason

## Foundations of Neuro-Symbolic AI

Alex Goessmann

University of Applied Science Würzburg-Schweinfurt

Summer Term 2026

# Contraction Calculus

Tensors appear naturally when representing factored systems, such as in

- Probabilistic Representation: Probability tensors
- Logical Representation: Semantic tensors

Reasoning amounts to calculate contractions of the representations

- Probabilistic Inference: Marginal distributions
- Locial Inference: Model counts (to decide entailment)

**`tnreason` is a tool for contraction calculus**

`representation` represents semantic and probability tensors and `engine` executes their contractions.

# Efficient Contraction Calculus

Efficient tensor representations by contractions of tensor networks

- Representation of empirical distributions $\mathbb{P}^D$ by datacores $\beta^{D_k}$
- Representations of complex formulas $f$ by connective encodings $\beta^\circ$

Localized algorithms demand only contractions with small number of open variables

- Gibbs Sampling
- Alternating Weight Optimization

Advanced: Algorithms with small number of tensor cores based on *Message Passing Approaches*.

# Approaching Use Cases with tnreason

**Buildtime** (Generation of a model):
tnreason.knowledge.HybridKnowledgeBase

- Elicitation from experts: Generation of formulas
- Calibration on Data: Optimization of weights to probabilistic formulas given data
- *Advanced Approach: Neuro-Symbolic AI*

**Runtime** (Usage of the model):
tnreason.knowledge.InferenceProvider

- Analysis of distributions by queries
- Infering missing information based on evidence
- Sampling random data from the model