

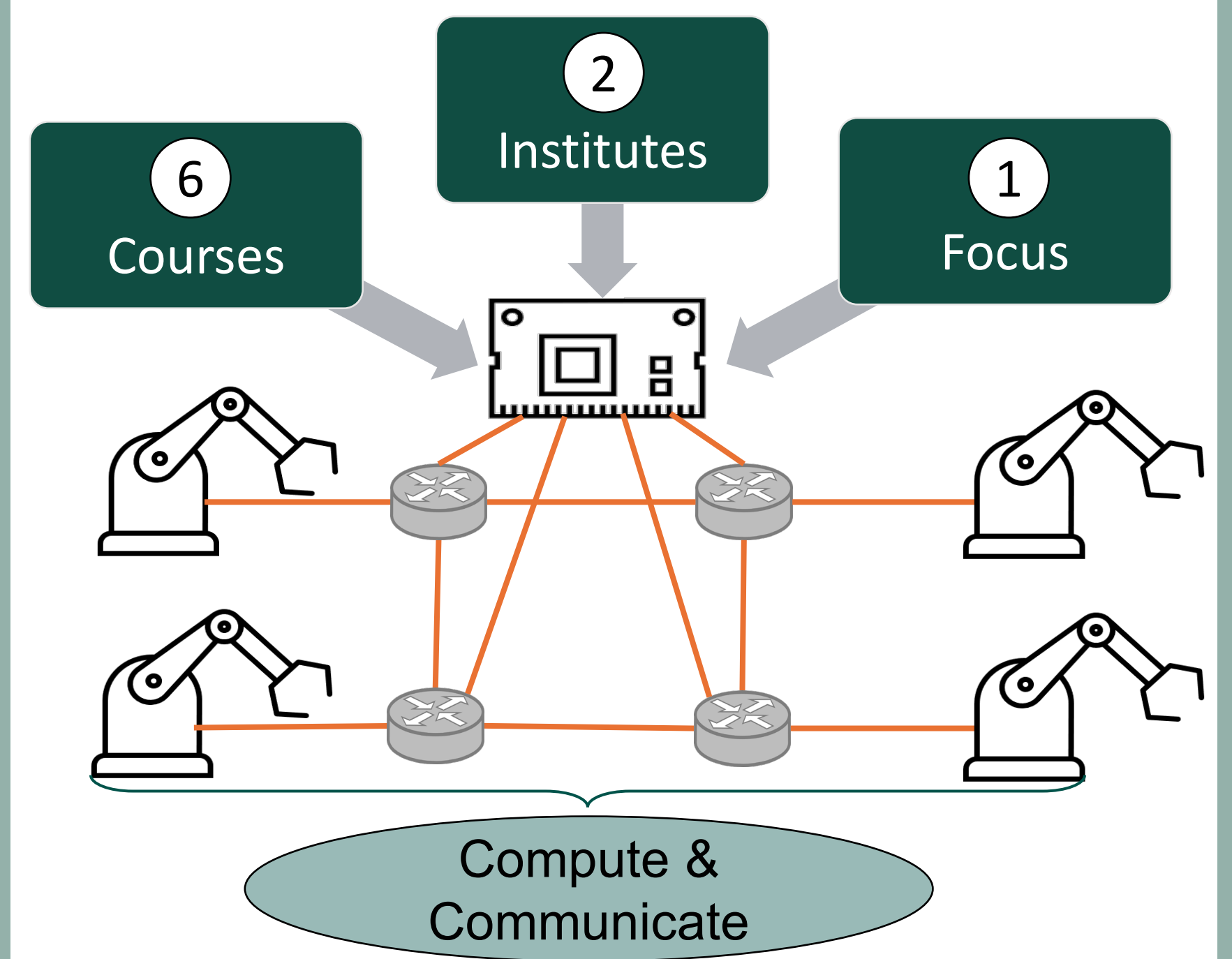
Platforms for Information Technology

Prof. Dr. Selma Saidi – Dr.-Ing. Lisa Maile, Dr.-Ing. Björn Fiethe, Omar Laimona (M.Sc.), Sanjida Reza Rafa (B.Sc.)
Prof. Dr.-Ing. Eduard Jorswieck – Dr. math. Christian Deppe, Jyun-Sian Wu (M.Sc.), Marcel Mross (M.Sc.)
Institute of Computer Engineering & Institute for Communications Technology
Technische Universität Braunschweig
lisa.maile@tu-braunschweig.de | Telefon +49 (0) 531 391-9663

Project Goals

An interdisciplinary approach to education.

Goal 1: Coherent Topics	Goal 2: Enhanced Remote Support
<ul style="list-style-type: none"> Integrated module cluster (30 ECTS, 1 semester) Focus on interfaces between platform components Designed for future engineers – with Big Picture No cyclic course dependencies 	<ul style="list-style-type: none"> Flexible learning for international students Live streaming of courses Interactive Jupyter Notebooks StudyBuddy: LLM for self-study assistance



Work Packages & Milestones

Research

- ✓ Identify needs of international students
- ✓ Contacted existing TU BS StudyBuddy solutions
- ✓ Get feedback from other modules which use Jupyter

Initial Setup

- ✓ Aquired and installed streaming tools
- ✓ Researched & implemented Demo-StudyBuddy with open-source tools
- ✓ Extended Jupyter Notebooks

Presentation & Deployment

- ✓ Some delays due to challenges (see on the right), but preparation for Winter Semester 2025
- ✓ Outreach to ERASMUS partners (France)
- ✓ Announced on websites

SCAN ME

Challenges

StudyBuddy

Privacy & Copyright Concerns

- Public LLMs unsuitable
- Alternatives: Llama 3, GWDG, OpenGPT-x, GITZ
- Currently: Discussion with IT User Board

Behavioral Complexity

- Required Inference Parameter Tuning
- Generic solution wanted

Jupyter

- Which type of task is suitable for Jupyter Notebooks?
- How to integrate it in the current setup of Lecture/Exercise?
- How to enable the students to easily access it?

Reflection & Evaluation

The module fosters interdisciplinary learning with a coherent topic for future engineers.

- Remote learning through streaming and Jupyter Notebooks
- StudyBuddy – Teaching resources for Retrieval-Augmented Generation
- Future Steps:
 - Further development & deployment of StudyBuddy with partners*
 - Further extending of materials
 - Continuing exchange of institutes

*Bijan Khosrawi-Rad & Patrick Hisce (Service Information Systems), Fabian Runge (Media Lab)

Weakly typical sequences

Definition: ϵ -weakly typical sequences

Let $X^{(n)} = (X_1, X_2, \dots, X_n)$ be a discrete, i.i.d. random vector with probability mass function $P_{X^n}(\cdot)$ and alphabet \mathcal{X}^n . Then all vectors with the same entropy $H(X) := H(X_1)$. Furthermore, let $\epsilon > 0$.

Then the set of all ϵ -weakly typical sequences is defined as follows:

$$\mathcal{A}^{(n)} = \left\{ x^n = (x_1, \dots, x_n) \in \mathcal{X}^n : 2^{-n(H(X) + \epsilon)} \leq P_{X^n}(x^n) \leq 2^{-n(H(X) - \epsilon)} \right\}$$

In plain words, the above definition means the following:

ϵ -weakly typical sequences are all sequences whose probability is roughly $2^{-nH(X)}$.

ϵ -weakly typical sequences can be expressed equivalently as follows:

$$\mathcal{A}^{(n)} = \left\{ x^n = (x_1, \dots, x_n) \in \mathcal{X}^n : H(X) - \epsilon \leq -\frac{1}{n} \log_2(P_{X^n}(x^n)) \leq H(X) + \epsilon \right\}$$

With this notation, we can interpret weakly typical sequences as follows:

A sequence of length n is called ϵ -weakly typical if the empirical entropy $-\frac{1}{n} \log_2(p^{(n)}(x^n))$ is close to the true entropy of the random variable X .

Example

The properties of typical sequences are most important for large n (i.e., long sequences).

In the following example, we consider i.i.d. Bernoulli random variables $X_i \sim B(q)$, i.e. $p(1) = q$. The alphabet is $\mathcal{X} = \{0, 1\}$.

- We generate random sequences $X^{(n)} = (X_1, X_2, \dots, X_n)$ of length n .
- We check whether the sequences are ϵ -weakly typical according to the definition.

```
def example_ratio_weakly_typical(num_seq=10):
    def update(n, q=0.1, eps=0.1):
        rv = stats.bernoulli(q)
        sequences = rv.rvs(size=(num_seq, n))
        prob_seq = rv.pmf(sequences)
        emp_ent = -np.log2(prob_seq, axis=1)
        real_ent = -np.log2(q)
        weak_typ = np.logical_and(real_ent - eps < emp_ent, emp_ent < real_ent + eps)
        for i, seq in zip(sequences, weak_typ):
            print(f"i: {i}, format_seq: {seq}")
            print(f"emp: {emp_ent[i]}, real: {real_ent}, format_real: {real_ent}")
    interact(update, n=(5, 50, 1), q=(0, .5, .95), eps=(0, 1, .95))
```