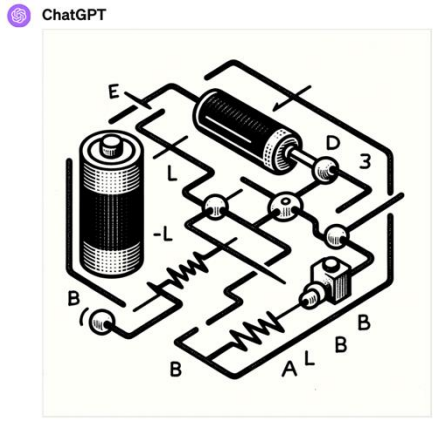
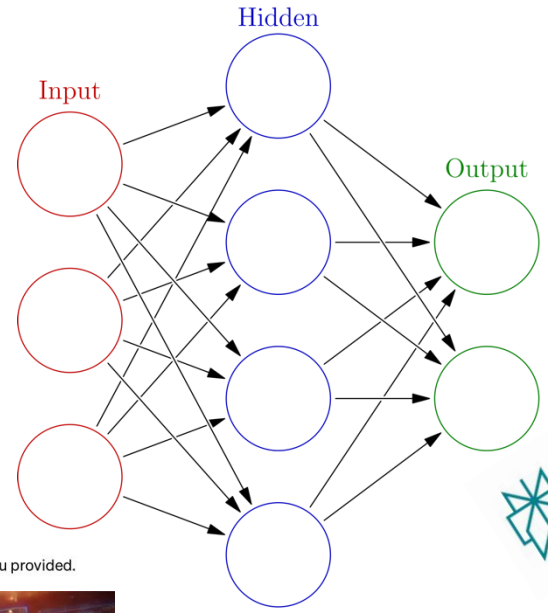
Teaching Digital Topics: A Practical Report with Dos and Don’ts

ErUM Data Hub
Deep Learning Train-the-Trainer

September 19, 2025





perplexity

ALEX BRITZ
FACILITATING SCIENCE

python™

Disclaimer

These slides primarily reflect my experiences and opinions from about five years of teaching digital topics.

What I’ve compiled here is likely incomplete and includes many personal perspectives, so please view this as a trigger for personal reflection rather than strict guidance.

If you have any doubts or questions, please contact me under mail@alexbritz.de.

Let's Play!

Myth or Fact

Myth or Fact I:
It Requires 10000 Hours to Reach Excellence

MYTH!

The psychologist Anders Ericsson published this number in 1993 about music students in Berlin.

But: You can‘t reach excellence just by practicing brute force for 10000 hours. Other factors are also important, for example, goal-oriented practice.

Image generated with ChatGPT

Yvonne Konstanze Behnke „Lernmythen aufgedeckt“

Myth or Fact II:
Videos are the Most Effective Way to Learn.

MYTH!

Pure video courses and long learning videos have a large abortion rate.

Consumption is passive.

Content often lacks fit and is inflexible.

Lack of interaction.

Distraction occurs due to multitasking.

Image generated with ChatGPT

Yvonne Konstanze Behnke „Lernmythen aufgedeckt“

Myth or Fact III: **Digital Natives Navigate The Digital Sphere Naturally and Easily.**

MYTH!

Digital literacy ≠ Digital familiarity.

We need to distinguish usage and competence – there‘s no clear correlation.

Digital skills need to be developed and are not inherited.

Image generated with ChatGPT

Yvonne Konstanze Behnke „Lernmythen aufgedeckt“

Who am I?

And what am I going to talk about.

Introduction of Myself

- By education: Experimental Physicist
- Research Career: Physical Chemistry and Materials Science at Synchrotrons, XFELs, and UED.
- Now: Trainer and Facilitator
- Taught about
  350+ PhD candidates, postdocs and group leaders in Python programming and
  150+ in Generative AI use in science and academia
  (and many more in communication skills)

What Are Typical Digital Topics?
Examples of Courses Which I Teach & Taught

- Generative AI in Science and Academia
- Python Programming
- Data Analysis and Visualization
- Machine Learning with Python
- But also: Project Management
- VLAB – a virtual XFEL Laboratory (lead by C. Bressler at Uni HH)

Digital Learning and Teaching (in Natural Sciences)

Moodle

Blink.it

Online Learning Platforms

Padlet.com

Blended Learning

Miro

Mural

Zoom

Communikation and Interaction

MS Teams

Digital Tools

Immersive, AR & VR

Virtual Labs

Scientific Programming and Data Science

SPSS

R

Python

(generative) AI

Midjourney

Synthesia

ChatGPT

Perplexity

Anthropic Claude

Github Copilot

Cursor

Teaching Digital Topics:
10 Practical Advices

1. **Adapt to different levels and personalities** → If possible, work with small groups (max 15 students).
2. **Assume no prior knowledge and challenge in baby steps** → Start with basics, accelerate challenges incrementally, finish with advanced concepts. Include optional extensions (bonus challenges).
3. **Avoid distractions and promote commitment** → Lead by example and be committed yourself. Find creative ways to avoid distractions and enhance commitment.
4. **Don't assume motivation** → Make relevance explicit (e.g., *"this method is used in real phsysics research"*).
5. **Balance practice and theory (≈50/50)** → Live coding, teamwork, and projects are as important as lectures.
6. **Distinguish familiarity from literacy** → Not just *"can use"*, but *"understands what it does and why"*.
7. **Promote collaboration** → Teamwork, pair programming, and peer review mirror real scientific practice.
8. **Encourage responsible tool use** → Rely on few, well-understood tools and techniques. Prefer basic over complex tools, promote critical awareness when using GenAI tools.
9. **Assure a fit of the training / workshop content** → Use pre-surveys to calibrate. Consider saying "no".
10. **Prioritize reflection, transfer & application** → Ask *"why does this matter?"*, work on projects with real datasets, emphasize reproducible workflows, and plan for application beyond the teaching format.

1. Adapt to Different Skills/Knowledge Levels and Personalities

Python Pre-Questionnaire

When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.

1. What's your name (optional)?

Enter your answer

2. And your email address (optional)?

Enter your answer

3. What's your research topic (optional)?

Enter your answer

4. Please describe your programming knowledge. Which programming languages do you already know? And to which level (beginner, intermediate, expert)?

Enter your answer

5. Did you already use Python in the past? If so, please name packages that you typically use and feel comfortable with.

Enter your answer

6. Please describe what you would like to learn in this course!

Enter your answer

7. Please describe how you would like to use Python in the future!

Enter your answer

8. Do you have data from experiments or simulations which you could work with during this course? If so, please describe the data in a few words (type of data, file formats, size, etc.)

Enter your answer

9. What should NOT happen during the course?

Enter your answer

10. Please fast forward in time. What should happen in this course in order for it to be successful to you?

Enter your answer

Submit

Challenges with Self-Assessment and Prior Knowledge

Scenario 1: A student signs up for the course „Machine Learning with Python” but has never written any Python code before.

Scenario 2: A student in the “Fundamentals of Python” course has already participated in a similar program half a year ago. They believe they didn’t learn anything but can complete all exercises quickly (and even help me teach the class).

2. Don’t Count on Prior Knowledge and Challenge in Baby Steps

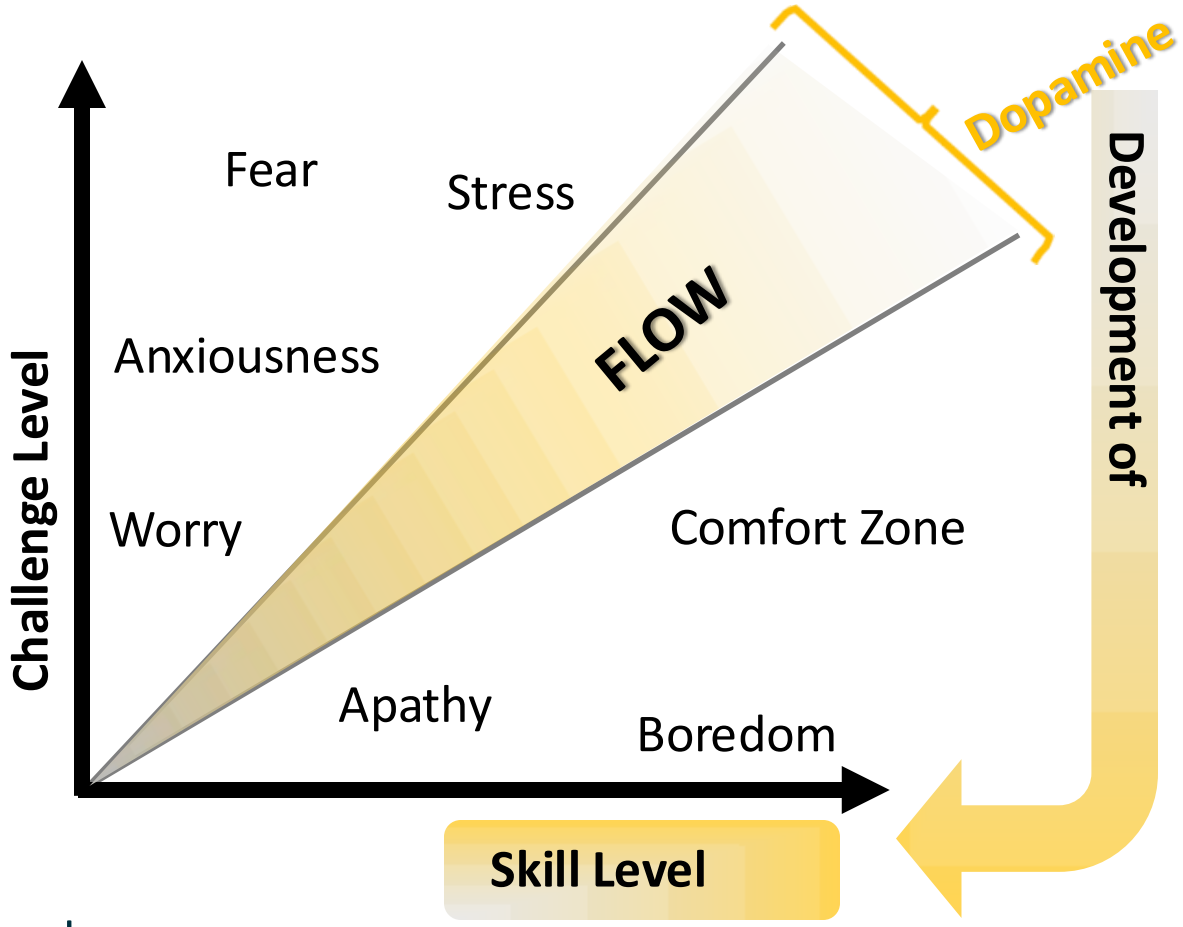
I Believe I Can Learn It!

The student says: *„I believe I can do It!“*

- Self-efficacy plays a major role in learning: If I can do it, I will make it.
- How can we enhance self-efficacy:
  - Find role models.
  - Celebrate small successes.

- How can we implement this? Take baby steps and challenge incrementally – one step at a time.

Enhance „Flow“ (Montessori & Csíkszentmihályi)



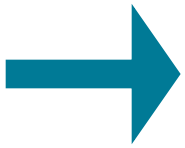
Slide: Max Happel

According to Steven Kottler the Challenges should be 4% above the Skills

https://www.flowresearchcollective.com/blog/flow-trigger-the-challenge-skill-ratio

Baby Steps in a Deep Learning Exercise

Build and train a neural network with PyTorch to classify handwritten digits from the MNIST dataset.



1. Set up a fresh Python environment and install torch, torchvision, and matplotlib.
2. Import PyTorch/torchvision; configure device detection (cuda if available) and set random seeds.
3. Download and load MNIST with torchvision.datasets.MNIST, applying ToTensor() and normalization; create train/test DataLoaders.
4. Visualize a small batch (e.g., 25 images) with labels to sanity-check the input pipeline.
5. Define a simple model (start with a CNN: Conv→ReLU→Pool→Conv→ReLU→Pool→FC→Dropout→FC); print a parameter count summary.
6. Choose CrossEntropyLoss and an optimizer (e.g., Adam), set reasonable hyperparameters (batch size, learning rate, epochs).
7. Implement the training loop: forward pass, compute loss, backprop, optimizer step, and track running loss/accuracy per epoch.
8. Implement an evaluation loop on the test set (no grad), reporting overall accuracy and a confusion matrix.
9. Run at least two experiments (e.g., change learning rate or add BatchNorm/Dropout) and compare metrics; plot loss/accuracy curves.
10. Save the best model with torch.save, reload it to verify inference on a few samples, and write a short paragraph summarizing results and next steps.

Simpley and quick example from ChatGPT

3. Avoid Distractions and Promote Commitment

Challenges with Blended Learning

We once designed a blended learning format about data analysis with python for PhD candidates in physics. It was a flipped classroom format with asynchronous video lectures, and we filmed a total of 10 hours of video material, ranging from fundamentals to advanced and expert levels. Although participation in the live session was great and the feedback positive, the video material was not as effective as expected. Of the 20 videos produced, students watched only the initial/fundamental ones; few students watched the later ones.

I have had similar experiences with online content of communication skills workshops.

→ Blended learning is challenging. Learning from asynchronous content requires dedication and commitment.

Distractions due to Emails

Instructor („Me“): „This is how you can do xyz…“

Student (looks up from their screen. „Can we also do xyz? That would be really interesting to know…“

Distractions, even if happening in a few seconds, kill our concentration.

Success Story: Forcing Commitment.

This year, I designed a new asynchronous Python Kickstarter course (with a total workload of about 8 hours).

I offered this course for free in exchange for

a) Feedback after completion and
b) Weekly updates from the students about their progress.

The course's completion rate was about 50%, which is extremely high for a fully asynchronous format.

4. Don’t Assume Motivation

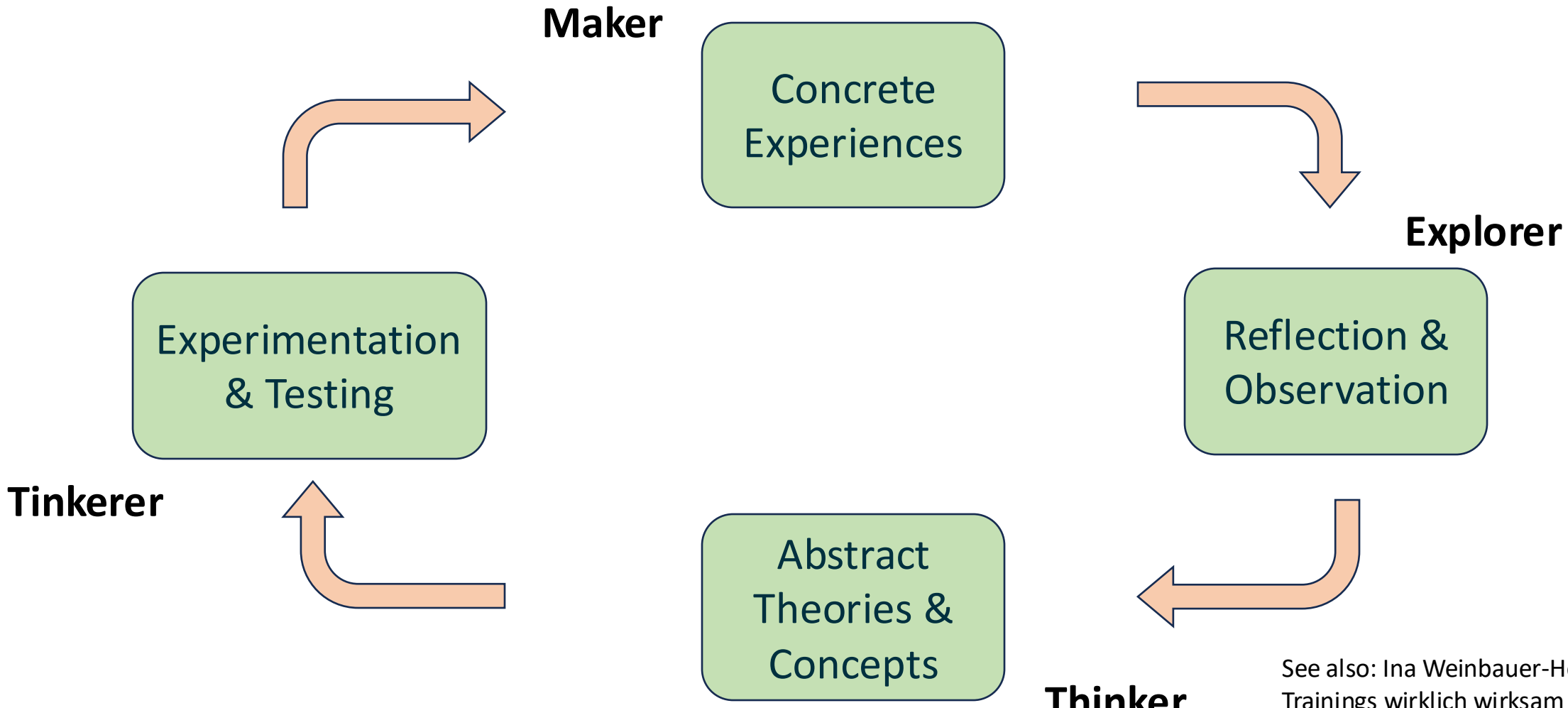
A Motivated Participant?!

„Machine Learning is a great skill to have on my CV. I saw this in the course catalogue of the university and signed up. It‘s free. Cool, right?“

Participant in the intro round.

5. Balance Theory and Practice

Learning by Doing



See also: Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“

Balance between Theory Input and Exercise / Application

Active Practice should make up

30% – 50%

of any Training or Workshop

6. Don’t Mistake Familiarity with Literacy

Familiarity or literacy?

Instructor: „How often do you use AI tools at work?“

Student: „Daily. It‘s fully incorporated into my workflows“

Instructor: „Amazing. What is your formula to prompt effectively?“

Student: „No formula, I just ask questions. Sometimes ChatGPT makes up citations. I don‘t know why…“

7. Promote Collaboration

Promote Collaboration and Team Work

Because it‘s fun.

And effective.

8. Responsible Tool Use

Battle of the Tools

Should I Teach Seaborn or Matplotlib?

- Low-level: Full control for custom plots.
- Requires more code for complex visuals.
- Extremely flexible and customizable.

- High-level: Quick creation of statistical and aesthetic plots.
- Simplified syntax, efficient for complex plots.
- Limited custom options, focused on good defaults.

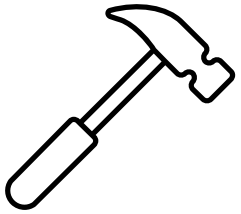
My Answer: Teach Matplotlib – because it‘s the foundation of Seaborn. If you know Matplotlib, learning Seaborn is trivial. The other way around difficult.

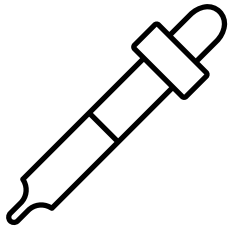
By Alex & Perplexity

9. Does the Training fit the Participant’s Needs?

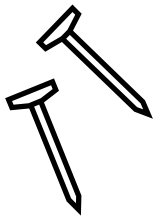
If I have a hammer, everything looks like a nail

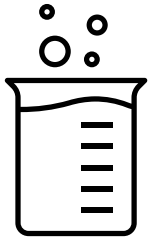
Solution
(=Tool)





Problem





But what is the implication for teaching Deep Learning?

→ Teach also other methods (e.g. statistical analysis, classification and regression)

Examples of Poor Fit of Training

Life scientists participate in „Fundamentals of Python“ course, although in their field and their environment everyone works with R.

Participants learn Deep Learning (PyTorch), but not simple classification and regression algorithms (e.g. with Scikit-Learn) – now they treat every problem with the most complex solution (the sledge hammer).

Python Pre-Questionnaire

When you submit this form, it will not automatically collect your details like name and email address unless you provide it yourself.

1. What's your name (optional)?

Enter your answer

2. And your email address (optional)?

Enter your answer

3. What's your research topic (optional)?

Enter your answer

4. Please describe your programming knowledge. Which programming languages do you already know? And to which level (beginner, intermediate, expert)?

Enter your answer

5. Did you already use Python in the past? If so, please name packages that you typically use and feel comfortable with.

Enter your answer

6. Please describe what you would like to learn in this course!

Enter your answer

7. Please describe how you would like to use Python in the future!

Enter your answer

8. Do you have data from experiments or simulations which you could work with during this course? If so, please describe the data in a few words (type of data, file formats, size, etc.)

Enter your answer

9. What should NOT happen during the course?

Enter your answer

10. Please fast forward in time. What should happen in this course in order for it to be successful to you?

Enter your answer

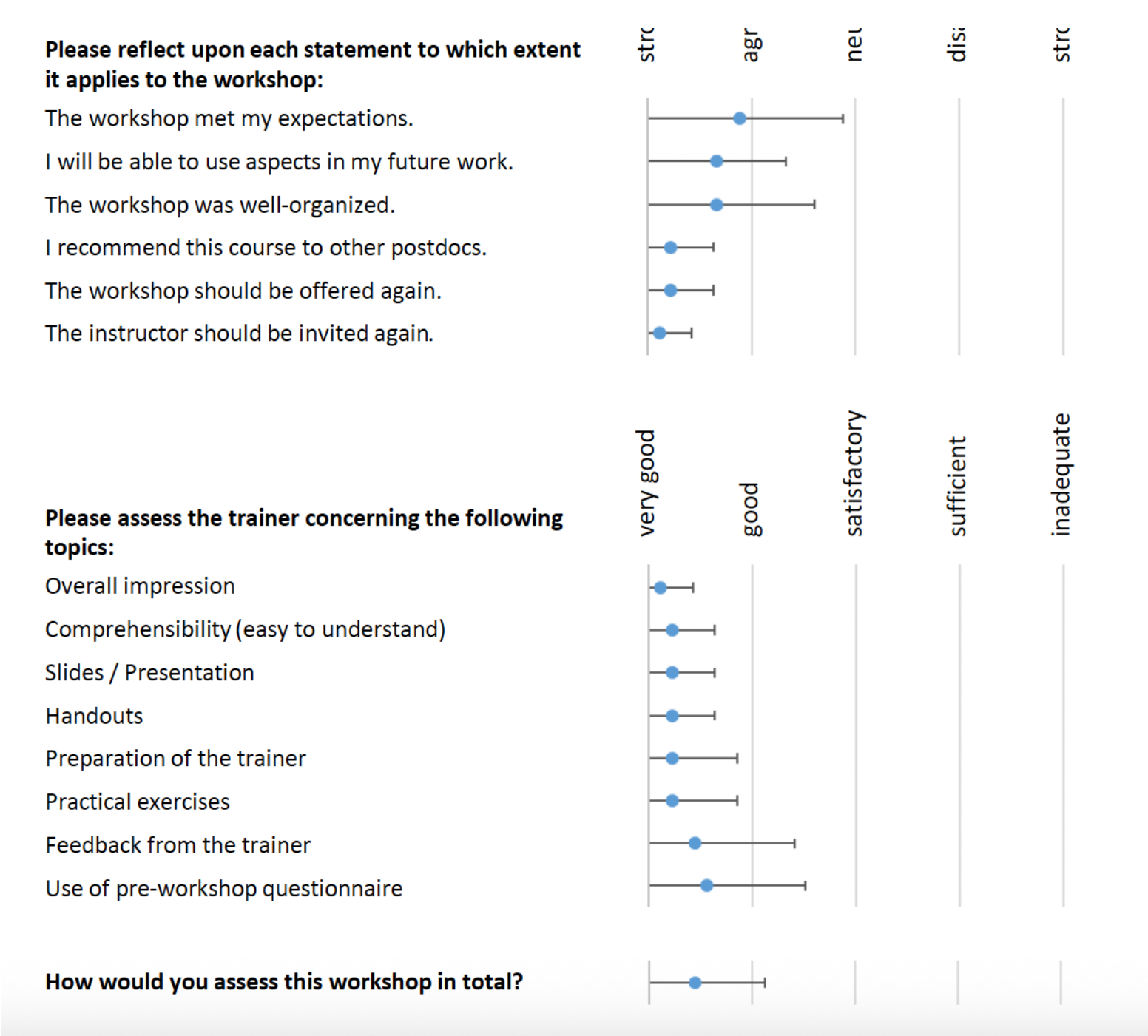
Submit

What Makes Learning Effective?

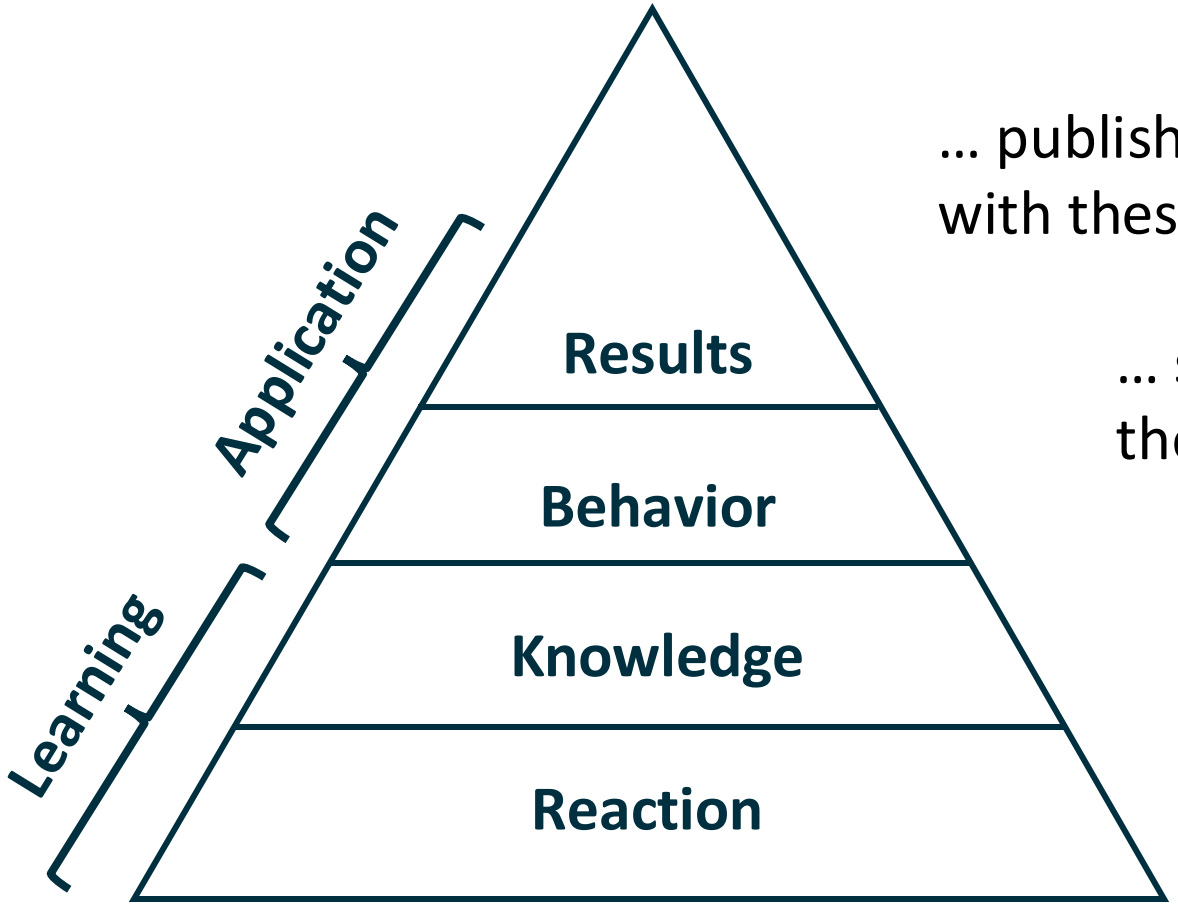
10. Prioritize Transfer

Evaluation with Smiley-Sheets

→ These only assess the perception/reaction of the students.



4 Levels of Learning Transfer (Kirckpatrick Model)



… publishes a paper with these results.

… starts to use nn to model their own data sets.

… knows how to model with nn.

The student liked the dl workshop.

What Makes Learning Effective?
5 Checks for Guidance.

See also: Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“

Teaching Digital Topics: 10 Practical Advices

1. **Adapt to different levels and personalities** → If possible, work with small groups (max 15 students).
2. **Assume no prior knowledge and introduce challenges gradually →** Begin with fundamentals, increase difficulty step by step, and conclude with advanced topics. Include optional extensions (bonus challenges).
3. **Avoid distractions and promote commitment →** Lead by example and stay committed yourself. Find creative ways to avoid distractions and strengthen commitment.
4. **Don't assume motivation** → Make relevance explicit (e.g., *"this method is used in real physics research"*).
5. **Balance practice and theory (≈50/50)** → Live coding, teamwork, and projects are as important as lectures.
6. **Distinguish familiarity from literacy** → Not just *"can use"*, but *"understands what it does and why"*.
7. **Promote collaboration** → Teamwork, pair programming, and peer review mirror real scientific practice.
8. **Encourage responsible tool use** → Rely on a few, well-understood tools and techniques. Prefer basic over complex tools. Promote critical awareness when using GenAI tools.
9. **Assure a fit of the training/workshop content** → Use pre-surveys to calibrate. Consider saying "no".
10. **Prioritize reflection, transfer & application** → Ask *"why does this matter?"*, work on projects with real datasets, emphasize reproducible workflows, and plan for application beyond the teaching format.

Acknowledgements and References

Vlab Project lead by Prof. Dr. Christian Bressler (Uni Hamburg / European XFEL)

Yvonne Konstanze Behnke „Lernmythen aufgedeckt“

Images generated partially with ChatGPT and Midjourney

Ina Weinbauer-Heidel „Was Trainings wirklich wirksam macht: 12 Stellhebel der Transferwirksamkeit“

Additional Slides

Beyond Classroom Lectures and Exercises

Teaching in a Digital World

The Future of Teaching

4 Brainstorming Questions:

1. What novel teaching techniques (beyond lecture and exercise) do you know? How effective are they?
2. In a dream world - how would you teach? What tools would be available?
3. How do you think GenAI will change learning? Describe specific scenarios!
4. What challenges regarding teaching do you expect in the future?

Step 1: Split into 4 groups (5 min)

Step 2: Interview participants from other groups about your question. Take notes if needed (20 min)

Step 3: Summarize your group's findings and display on a flipchart (15 min)

Step 4: Let's discuss in the plenary (45 min)

Step 5: Closeout remarks