Do’s and Dont’s in Scientific Talks

Bogdan Savchynskyy

12/12/2022

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Computer Vision and Learning Lab

Motivation

Talk material

Slides

+ story

- Get credit points
- Learn how to give talks
- Learn about the topic

Motivation

Talk material

Slides

\+ story

- Get credit points
- Learn how to give talks
- Learn about the topic

- Get higher grade for your PhD/Master/Bachelor
- Get feedback to you work

Motivation

Talk material

Slides + story

- Get credit points
- Learn how to give talks
- Learn about the topic

- Get higher grade for your PhD/Master/Bachelor
- Get feedback to you work

You are judged by your talk:

Each talk is an application talk

Outline

- **Slides content** (What is allowed on the slides and what isn‘t)
- **Slides order** (General structure of a talk)

Outline

- **Slides content** (What is allowed on the slides and what isn‘t)
- **Slides order** (General structure of a talk)

- **Slides order** (General structure of a talk)
- **Slides content** (What is allowed on the slides and what isn‘t)

Slides Order

General structure of a talk

Typical talk outline?

1. Title
2. Outline
3. Problem description
4. Method/Solution
5. Experiments
6. Conclusions
7. Future work

DSAC – Differentiable RANSAC for Camera Localization

Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, Carsten Rother

TU Dresden, Microsoft

Outline

1. RANSAC
2. Camera localization problem
3. Learning camera localization
4. Our end-to-end learning approach, DSAC
5. Experiments
6. Conclusions and Outlook

How much do you understand out of it?
Is this outline helpful for you?

Typical talk outline

1. Title
2. **Outline**
3. **Problem description**
4. Method/Solution
5. Experiments
6. Conclusions
7. Future work

!

1. Title
2. **Problem description**
3. **Outline**
4. Method/Solution
5. Experiments
6. Conclusions
7. Future work

Problem description explains the outline.

Typical talk outline

1. Title
2. **Outline**
3. **Problem description**
4. Method/Solution
5. Experiments
6. Conclusions
7. Future work

!

1. Title
2. **Problem decription**
3. ~~**Outline**~~
4. Method/Solution
5. Experiments
6. Conclusions
7. Future work

Problem description explains the outline.
You do not need an outline for most short (< 30 mins) talks.

Video: Provide your timing

1. Title
2. Problem description
3. Method/Solution
4. Experiments
5. Conclusions
6. Future work

Example of a talk

YouTube Video: DSAC – Differentiable RANSAC for Camera Localization

Typical talk timing

	Time from the beginning of the talk:
1. Title	< 1 min: Thanks, coauthors
2. Problem description	**< 5 min**
3. Method/Solution	> 50 % of the talk
4. Experiments	
5. Conclusions	20-30 % of the talk
6. Future work	

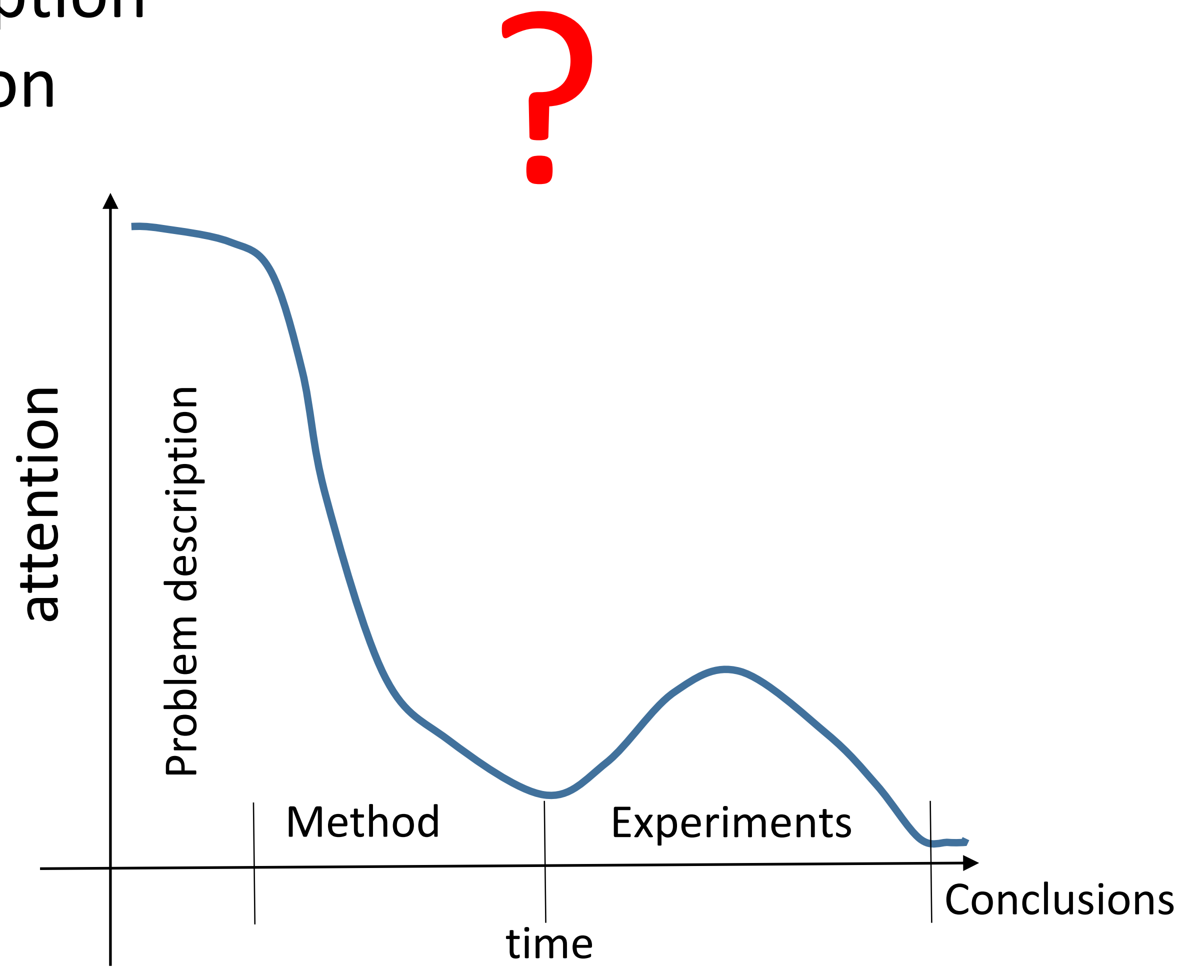
The most important part of the talk

1. Title
2. Problem description
3. Method/Solution
4. Experiments
5. Conclusions
6. Future work

?

The most important part of the talk

1. Title
2. Problem description
3. Method/Solution
4. Experiments
5. Conclusions
6. Future work



Video: The most important part of the talk

Problem description:

- simplified description (top view, clear for broad audience)
- more details (necessary to pose your actual problem)
- existing solutions
- deficiencies you address (problem statement)
- advertisement (short description of your results)

Example of a talk

Rolling Outline

1. RANSAC
2. Camera localization problem
3. **Learning camera localization**
4. Our end-to-end learning approach, DSAC
5. Experiments
6. Conclusions and Outlook

When:

- long talk
- involved method

Outline

1. **RANSAC**
2. Camera localization problem
3. Learning camera localization
4. Our end-to-end learning approach, DSAC
5. Experiments
6. Conclusions and Outlook

RANSAC

…

Outline

1. RANSAC
2. **Camera localization problem**
3. Learning camera localization
4. Our end-to-end learning approach, DSAC
5. Experiments
6. Conclusions and Outlook

Camera Localization Problem

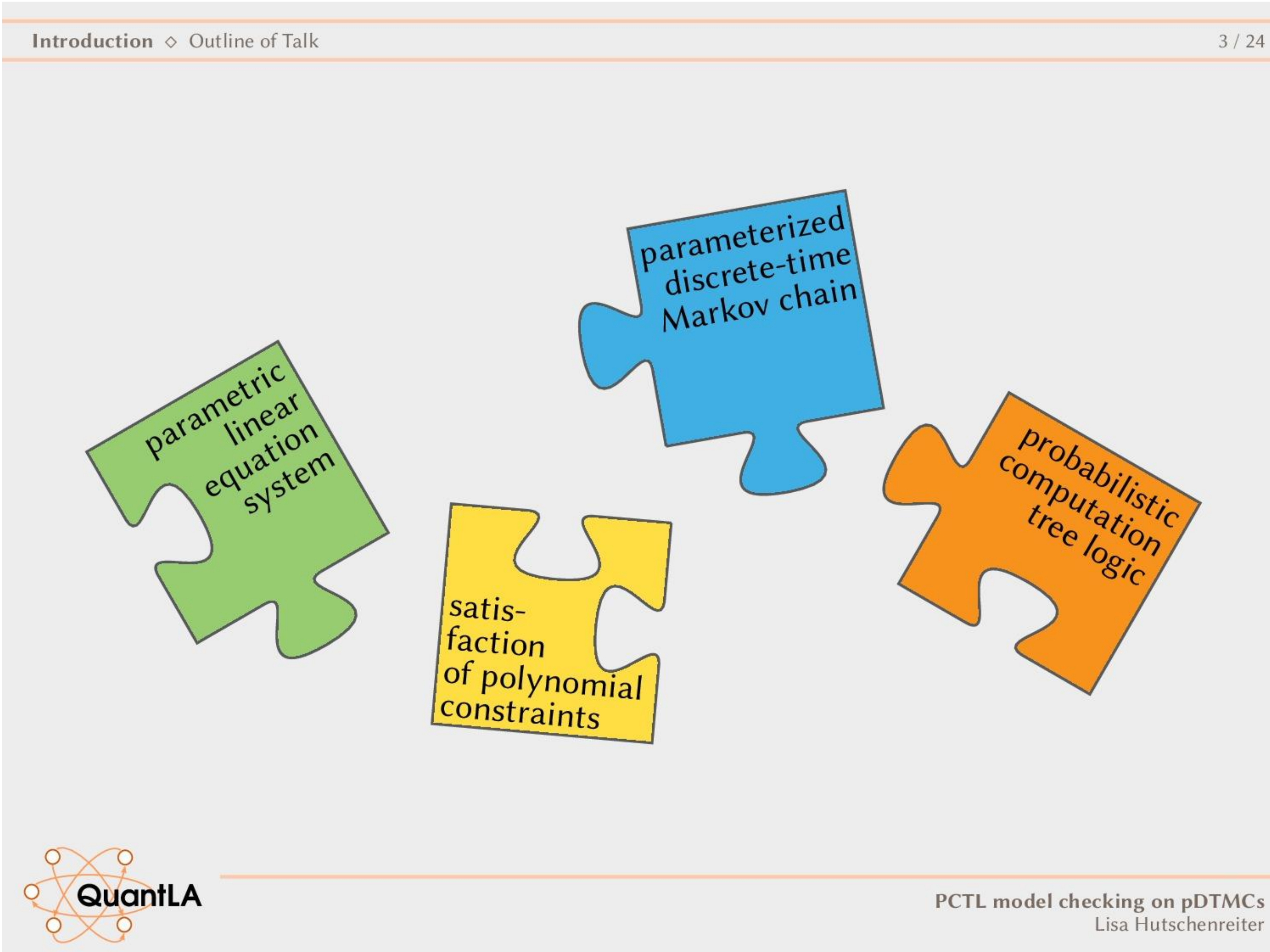
…

Hint: For complex talks consider conclusions to each part

Outline

1. RANSAC
2. Camera localization problem
3. **Learning camera localization**
4. Our end-to-end learning approach, DSAC
5. Experiments
6. Conclusions and Outlook

Creative Rolling Outline: Puzzles

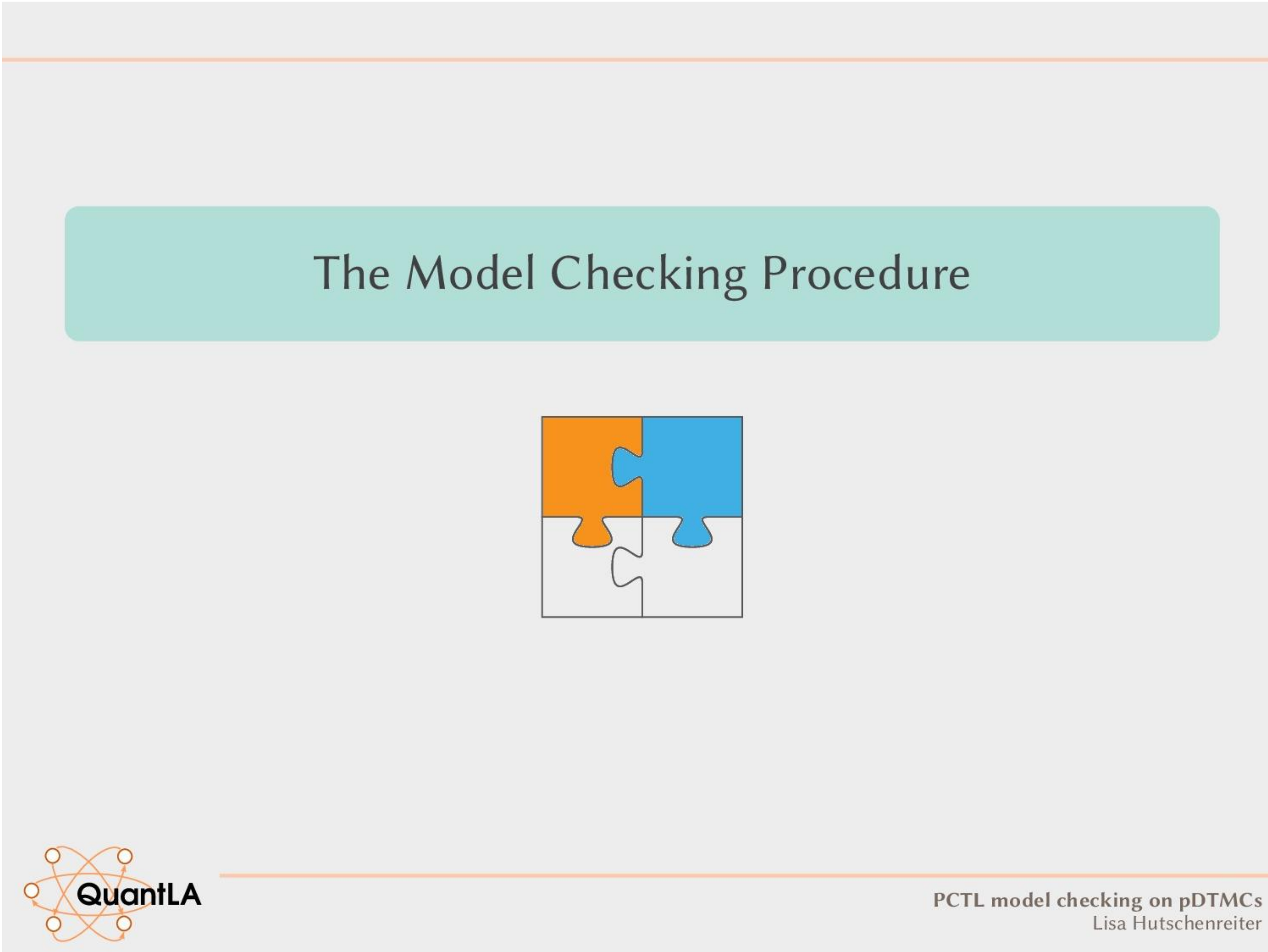


Creative Rolling Outline: Puzzles

Creative Rolling Outline: Puzzles

A **discrete-time Markov chain** $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathbf{P}, \mathrm{AP}, \mathcal{L})$.

Creative Rolling Outline: Puzzles



Creative Rolling Outline: Puzzles

Input

- PCTL state formula φ

$$\varphi ::= \text{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_{\bowtie c}(\Phi) \qquad \text{state formula}$$

$$\Phi ::= \Diamond\, \varphi \qquad \text{path formula}$$

Slides Content

What is allowed on the slides and what not

Information Representation

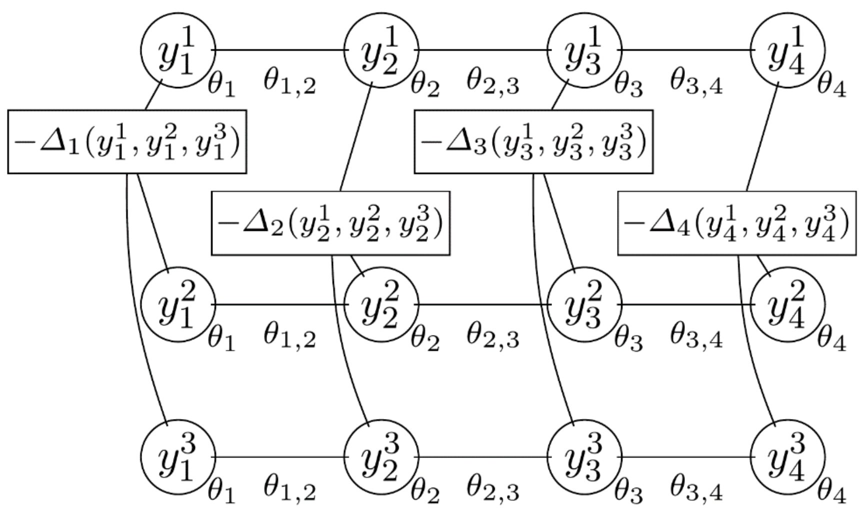
Text

A combinatorial optimization problem can be characterized by a mapping $f : R^m \rightarrow \{0,1\}^n$, where the problem description is mapped to a binary vector representing an optimal problem solution. In case of ILPs the description consists of the coefficients of the objective function and the constraint matrix. Computation of the mapping *f* has in general a complexity that grows at least as an exponent of *m*.

Formulas

$$\mathbf{K} = \mathrm{diag}(\mathrm{vec}(\mathbf{K}_p)) + (\mathbf{G}_2 \otimes \mathbf{G}_1)\,\mathrm{diag}(\mathrm{vec}(\mathbf{K}_q))(\mathbf{H}_2 \otimes \mathbf{H}_1)^\top$$

Images



Information Representation: Text

Text

A combinatorial optimization problem can be characterized by a mapping $f : R^m \rightarrow \{0,1\}^n$, where the problem description is mapped to a binary vector representing an optimal problem solution. In case of ILPs the description consists of the coefficients of the objective function and the constraint matrix. Computation of the mapping f has in general a complexity that grows at least as an exponent of m.

Information Representation: Text

Text

A combinatorial optimization problem can be characterized by a mapping $f : R^m \rightarrow \{0,1\}^n$, where the problem description is mapped to a binary vector representing an optimal problem solution. In case of ILPs the description consists of the coefficients of the objective function and the constraint matrix. Computation of the mapping f has in general a complexity that grows at least as an exponent of m.

$f : R^m \rightarrow \{0,1\}^n$ *- optimization problem, NP-hard*

Text

A combinatorial optimization problem can be characterized by a mapping $f : R^m \rightarrow \{0,1\}^n$, where the problem description is mapped to a binary vector representing an optimal problem solution. In case of ILPs the description consists of the coefficients of the objective function and the constraint matrix. Computation of the mapping f has in general a complexity that grows at least as an exponent of m.

$f : R^m \rightarrow \{0,1\}^n$ - optimization problem, NP-hard

Example:

$$f(A,c) = \underset{\substack{Ax \leq b \\ x \in \{0,1\}^m}}{\arg\min} \langle c, x \rangle$$

Text

A combinatorial optimization problem can be characterized by a mapping $f : R^m \rightarrow \{0,1\}^n$, where the problem description is mapped to a binary vector representing an optimal problem solution. In case of ILPs the description consists of the coefficients of the objective function and the constraint matrix. Computation of the mapping f has in general a complexity that grows at least as an exponent of m.

$f : R^m \rightarrow \{0,1\}^n$ *- optimization problem, NP-hard*

Example:
$$f(A,c) = \arg\min_{\substack{Ax \leq b \\ x \in \{0,1\}^m}} \langle c, x \rangle$$

Allowed: keywords, names, terminology

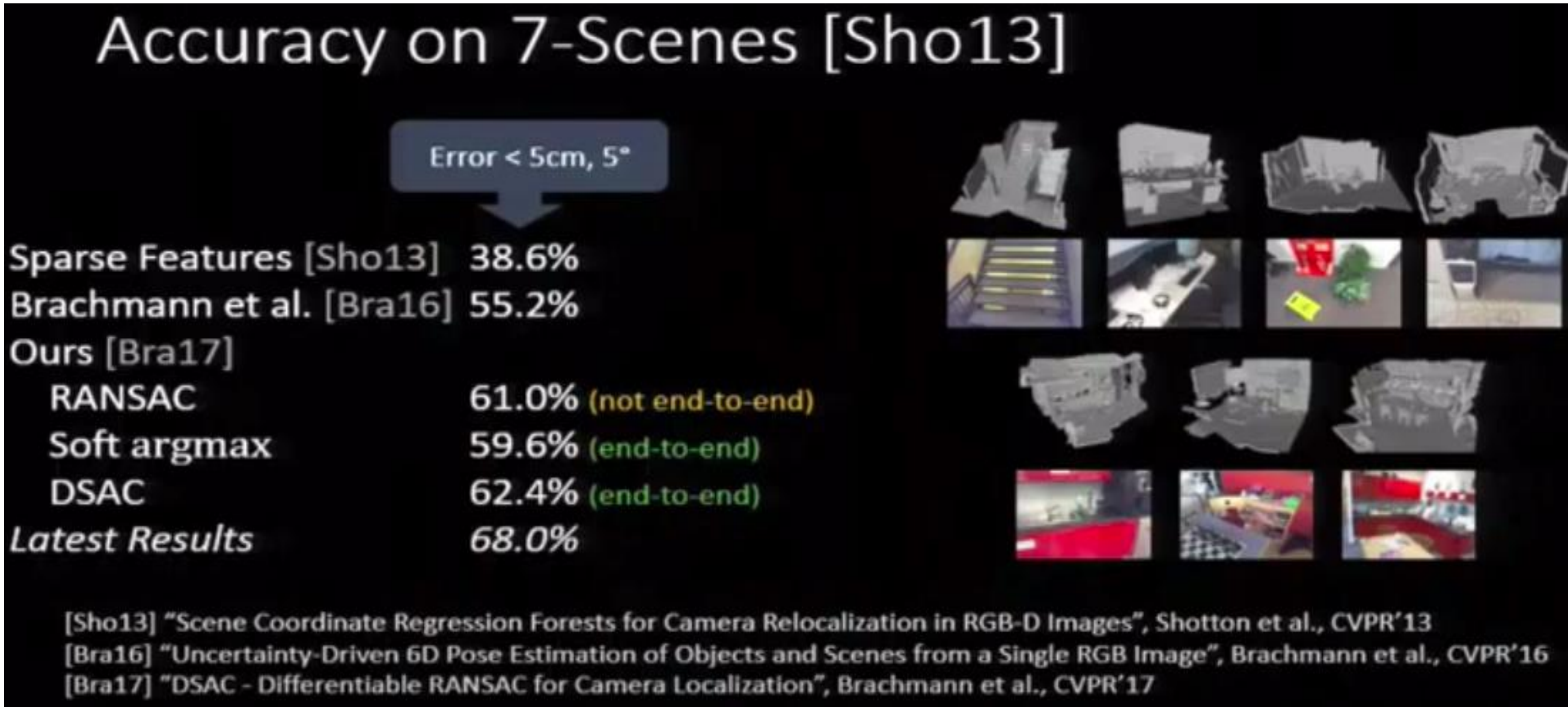
Avoid non-standard acronyms! SVM, VAE, ICA, CINN, LAP, PCA, CNN, MRF, CRF, MAP?

Information Representation: Text in Tables

Table 1. Accuracy measured as the percentage of test images where the pose error is below 5cm and 5°. *Complete* denotes the combined set of frames (17000) of all scenes. Numbers in green denote improved accuracy after end-to-end training for SoftAM resp. DSAC compared to componentwise training. Similarly, red numbers denote decreased accuracy. **Bold** numbers indicate the best result for each scene.

	Sparse Features [36]	Brachmann *et al.* [5]	Ours: Trained Componentwise			Ours: Trained End-To-End	
			RANSAC	SoftAM	DSAC	SoftAM	DSAC
Chess	70.7%	**94.9%**	**94.9%**	94.8%	94.7%	94.2% -0.6%	94.6% -0.1%
Fire	49.9%	73.5%	75.1%	75.6%	75.3%	**76.9%** +1.3%	74.3% -1.0%
Heads	67.6%	48.1%	72.5%	**74.5%**	71.9%	74.0% -0.5%	71.7% -0.2%
Office	36.6%	53.2%	70.4%	**71.3%**	69.2%	56.6% -14.7%	71.2% +2.0%
Pumpkin	21.3%	**54.5%**	50.7%	50.6%	50.3%	51.9% +1.3%	53.6% +3.3%
Kitchen	29.8%	42.2%	47.1%	47.8%	46.2%	46.2% -1.6%	**51.2%** +5.0%
Stairs	9.2%	**20.1%**	6.2%	6.5%	5.3%	5.5% -1.0%	4.5% -0.8%
Average	40.7%	55.2%	59.5%	**60.1%**	59.0%	57.9% -2.2%	**60.1%** +1.1%
Complete	38.6%	55.2%	61.0%	61.6%	60.3%	57.8% -3.8%	**62.5%** +2.2%

Paper-style table.
Make it short enough to explain everything!



Presentation-style table.

Information Representation: Formulas

$$\mathbf{K} = \mathrm{diag}(\mathrm{vec}(\mathbf{K}_p)) + (\mathbf{G}_2 \otimes \mathbf{G}_1)\, \mathrm{diag}(\mathrm{vec}(\mathbf{K}_q))(\mathbf{H}_2 \otimes \mathbf{H}_1)^\top$$

Factorization I

$$\mathbf{K} = \text{diag}(\text{vec}(\mathbf{K}_p)) + (\mathbf{G}_2 \otimes \mathbf{G}_1)\text{diag}(\text{vec}(\mathbf{K}_q))(\mathbf{H}_2 \otimes \mathbf{H}_1)^T.$$

$$\mathbf{K}_p = \begin{pmatrix} 1 & 1 & 4 \\ 10 & 3 & 2 \\ 2 & 9 & 3 \\ 1 & 2 & 8 \end{pmatrix}$$

$$\cdots = \left(\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8
\end{array}\right)$$

Factorization II

$$\mathbf{K} = \text{diag}(\text{vec}(\mathbf{K}_p)) + (\mathbf{G}_2 \otimes \mathbf{G}_1)\text{diag}(\text{vec}((\mathbf{K}_q))(\mathbf{H}_2 \otimes \mathbf{H}_1)^{\mathsf{T}}.$$

$$\mathbf{G}_2 \otimes \mathbf{G}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \mathbf{G}_1 = \begin{pmatrix} \mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 & \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_1 \end{pmatrix}$$

Factorization III

$$\mathbf{K} = \text{diag}(\text{vec}(\mathbf{K}_p)) + (\mathbf{G}_2 \otimes \mathbf{G}_1)\text{diag}(\text{vec}((\mathbf{K}_q))(\mathbf{H}_2 \otimes \mathbf{H}_1)^T.$$

$$\begin{bmatrix} \mathbf{G}_1 & 0 & 0 & 0 \\ 0 & \mathbf{G}_1 & \mathbf{G}_1 & 0 \\ 0 & 0 & 0 & \mathbf{G}_1 \end{bmatrix} \cdot \begin{bmatrix} d(\mathbf{k}^q_{(a,b)}) & 0 & 0 & 0 \\ 0 & d(\mathbf{k}^q_{(b,c)}) & 0 & 0 \\ 0 & 0 & d(\mathbf{k}^q_{(b,a)}) & 0 \\ 0 & 0 & 0 & d(\mathbf{k}^q_{(c,b)}) \end{bmatrix} \cdot \begin{bmatrix} 0 & \mathbf{H}_1 & 0 \\ 0 & 0 & \mathbf{H}_1 \\ \mathbf{H}_1 & 0 & 0 \\ 0 & \mathbf{H}_1 & 0 \end{bmatrix}$$

=

	a	b	c
a	$\mathbf{0}$	$\mathbf{G}_1\text{diag}(\mathbf{k}^q_{(a,b)})\mathbf{H}_1^T$	$\mathbf{0}$
b	$\mathbf{G}_1\text{diag}(\mathbf{k}^q_{(b,a)})\mathbf{H}_1^T$	$\mathbf{0}$	$\mathbf{G}_1\text{diag}(\mathbf{k}^q_{(b,c)})\mathbf{H}_1^T$
c	$\mathbf{0}$	$\mathbf{G}_1\text{diag}(\mathbf{k}^q_{(c,b)})\mathbf{H}_1^T$	$\mathbf{0}$

Images vs. formulas

Linear Assignment Problem

Goal

$$\text{maximize} \sum_{(i,j)\in A} a_{ij} x_{ij}$$

Restrictions

$$\sum_{\{j|(i,j)\in A\}} x_{ij} = 1 \quad \forall i \in N$$

$$\sum_{\{i|(i,j)\in A\}} x_{ij} = 1 \quad \forall j \in N$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in A$$

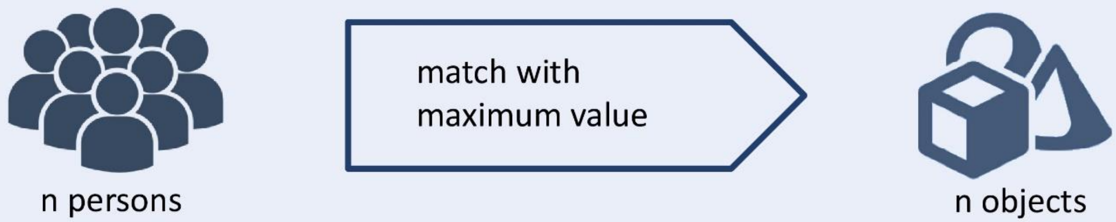
Is everything clear? Can you imagine the problem?

Images vs. formulas

What is the Linear Assignment Problem?

Heidelberg Collaboratory for Image Processing

Goal

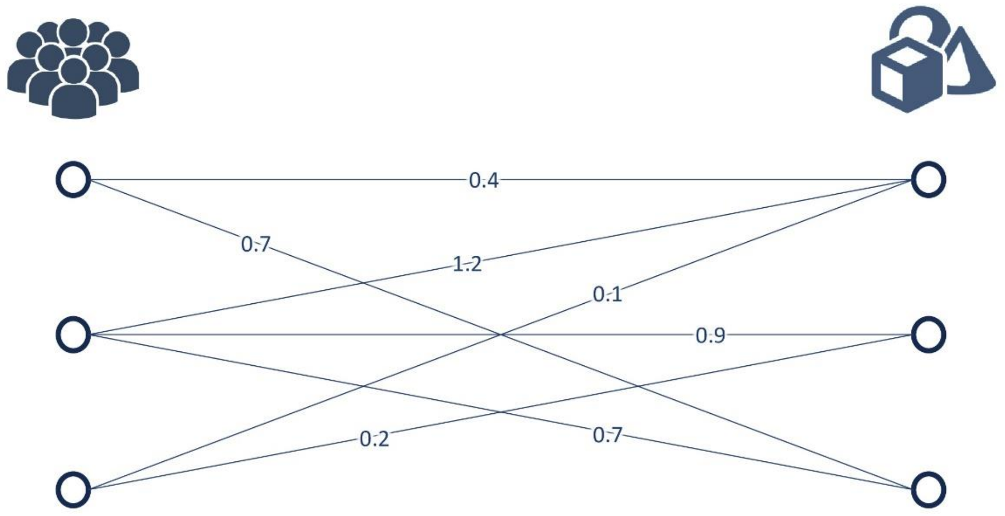


Restrictions

- can‘t match every person to every object
- matching happens on a one-to-one basis

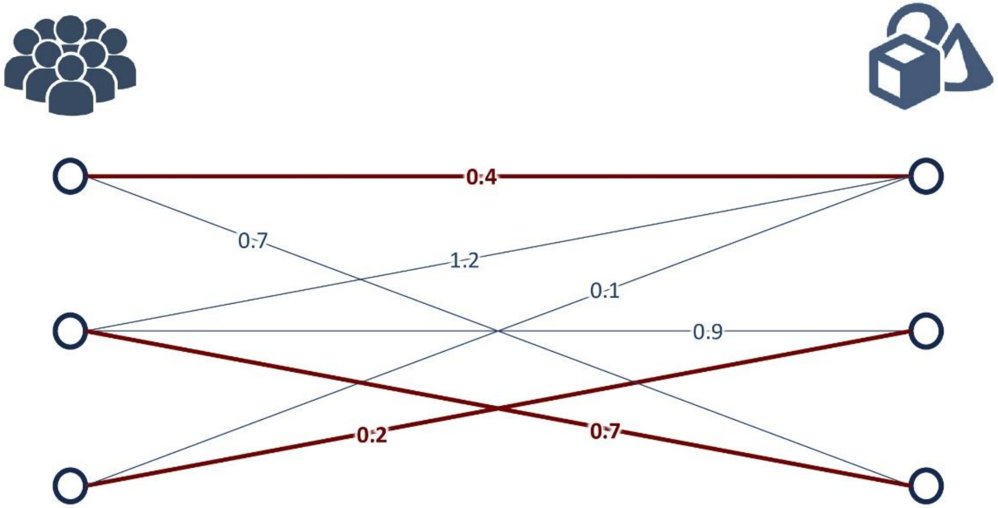
Images vs. formulas

Linear Assignment Problem



Images vs. formulas

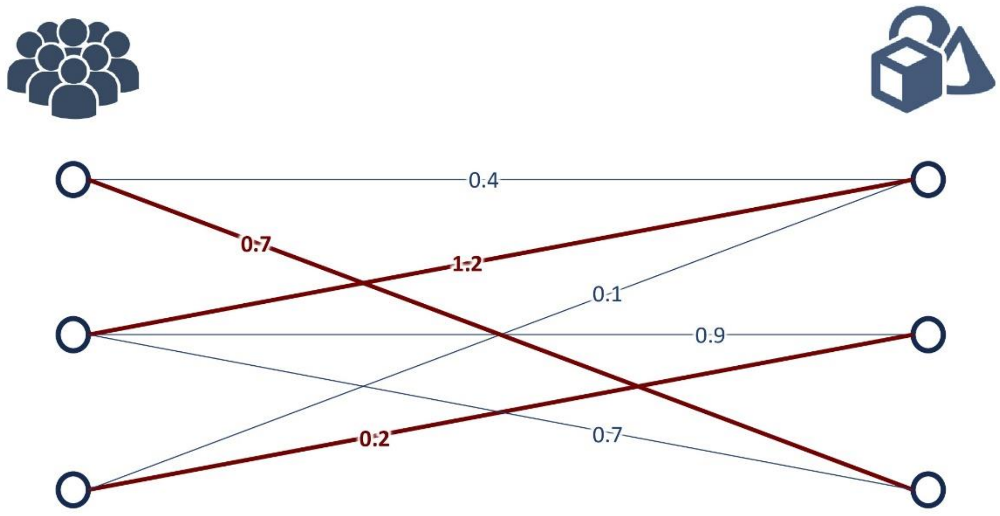
Linear Assignment Problem



total value: $0.4 + 0.7 + 0.2 = 1.3$

Images vs. formulas

Linear Assignment Problem



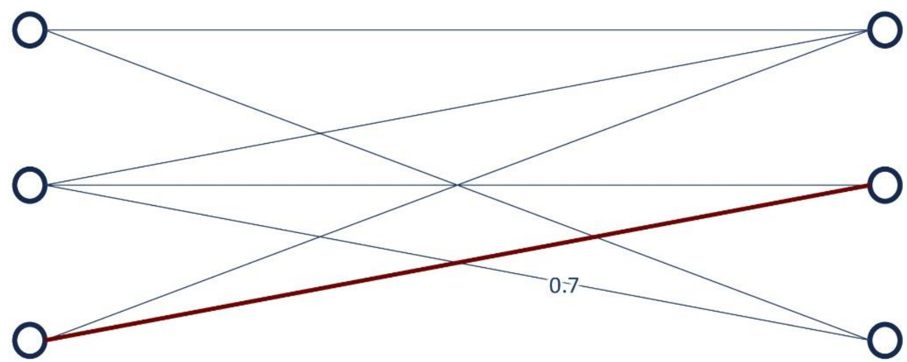
total value: $0.7 + 1.2 + 0.2 = 2.1$

Computer Vision and Learning Lab

Information Representation: Images

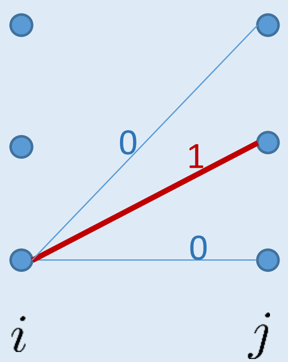
Linear Assignment Problem

persons $i \in \{1, \dots, n\}$



objects $j \in \{1, \dots, n\}$

assignment restrictions



$$i : \sum_j x_{ij} = 1$$

value

0.7

$a_{ij} \in \mathbb{R}^+ \quad \forall (i,j) \in A$

assignments

$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A$

Computer Vision and Learning Lab

Images vs. formulas

Linear Assignment Problem

Heidelberg Collaboratory for Image Processing

Goal

$$\text{maximize} \sum_{(i,j)\in A} a_{ij} x_{ij}$$

Restrictions

$$\sum_{\{j|(i,j)\in A\}} x_{ij} = 1 \quad \forall i \in N$$

$$\sum_{\{i|(i,j)\in A\}} x_{ij} = 1 \quad \forall j \in N$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in A$$

14

Better now?

Graph- or geometry-related problem? – Use images!

Talk vs. paper or lecture notes

Table 1: **Characteristics of datasets.** For all datasets used for evaluation we state number of instances (*inst.*), number of nodes (n), number of labels ($|\mathcal{L}|$), and graph density in percent (*dens.*).

	inst.	n	$\lvert\mathcal{L}\rvert$	*dens. (%)*
`hotel`	105	30	$= n$	100
`house`	105	30	$= n$	100
`car`†	30	19 - 49	$= n$	11 - 27
`motor`†	20	15 - 52	$= n$	10 - 32
`flow`	6	48 - 126	$\approx n$	45 - 98
`opengm`	4	19 / 20	$= n$	66 / 100
`worms`	30	558	$\approx 2.4\,n$	≈ 1.5
`pairs`	16	511 - 565	$\approx n$	≈ 20

† Zero edges were removed. Prior to this, graph density was 100 %.

What is wrong with this slide?

Talk vs. paper or lecture notes

Algorithm 2: Iterative Pruning Arc Consistency

Input: Cost vector $f \in \mathbb{R}^{\mathcal{I}}$, test labeling $y \in \mathcal{X}$;

Output: Strictly improving substitution p;

$(\forall v \in \mathcal{V})\ \mathcal{Y}_v := \mathcal{X}_v \backslash \{y_v\}$;

while true

  Construct verification problem $g := (I - [p])^\top f$ with p defined by (7);

  Use dual solver to find φ such that g^φ is arc consistent;

  $\mathcal{O}_v(\varphi) := \{i \in \mathcal{X}_v \mid g_v^\varphi(i) = 0\}$;

  if $(\forall v \in \mathcal{V})\ \ \mathcal{O}_v(\varphi) \cap \mathcal{Y}_v = \varnothing$ **then return** p;

  for $v \in \mathcal{V}$ **do**

    Pruning of substitutions: $\mathcal{Y}_v := \mathcal{Y}_v \backslash \mathcal{O}_v(\varphi)$;

What is wrong with this slide?

Talk vs. paper or lecture notes

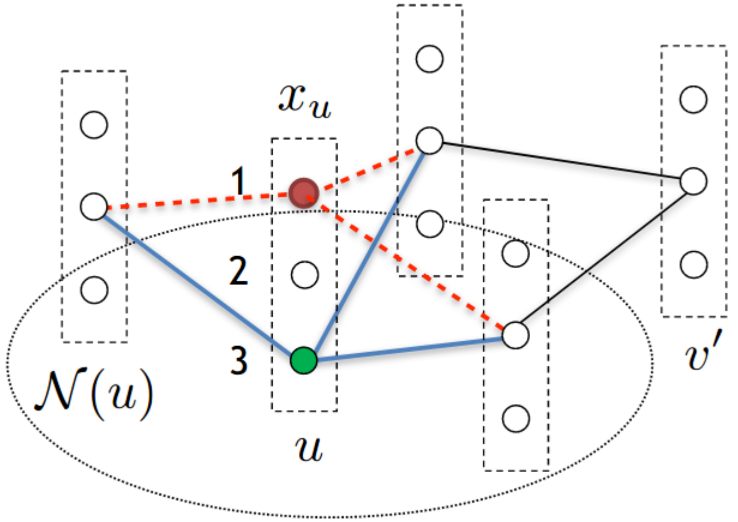


Fig. 2: Dead end elimination / dominance. Variables are shown as boxes and their possible labels as circles. Label $x_u = 1$ is substituted with label $x_u = 3$. If for any configuration of neighbors $x_{\mathcal{N}(u)}$ the energy does not increase (only the terms inside $\{u\} \cup \mathcal{N}(u)$ contribute to the difference), label $x_u = 1$ can be eliminated without loss of optimality.

What is wrong with this slide?

Title slide

The Best Method to Solve
the Universal Problem

Max Mustermann

Seminar on
„ Optimization in Machine Learning and Computer Vision“

Heidelberg 2021

What is missing here?

The Best Method to Solve the Universal Problem

Y. Zhang, J. Schmidt, A. Zimmersinger
University of Toronto, DeepResearch Ltd.

Presenter: Max Mustermann

Seminar on
„Optimization in Machine Learning and Computer Vision“

Heidelberg 2021

Authors and affiliations!

Do's of an entertaining talk

- **Joke** (if you feel comfortable with that)
- **Advertise** (e.g. after problem formulation)
- **Simplify** (give examples)
- **Intrigue** (e.g. seemingly correct conclusions)

Checklist/Feedback criteria

1. Talk structure:

- Is the general structure of the presentation correct, as in Sl. 14?
- Is the **Problem description** composed as in Sl. 19? Was the timing ok? (<5 min)
- Would you change anything in the structure of the **Method/Solution** part?
- Useless or missing/insufficient **Outline**?

2. Slide content:

- Too much text?
- Too few explanatory images?
- Difficult formulas with insufficient explanation (did you understand them)?
- Long image/table captions?
- Enumeration of formulas, descriptions starting with **Table**, **Figure**, **Algorithm** etc?
- Are there paper-style slides, tables, algorithms, theorems etc. ?
- Title page ok?

3. Presentation and question answering:

- Missing/incorrect/uncertain answers to the questions?
- Intonation/voice modulation: Emphasis on important things?
- Acoustical problems: Too fast/not loud enough/unclear pronunciation etc.?

A good practice applied math presentation

YouTube Video: An explicit analysis of the entropic penalty in linear programming

Thanks you for attention!

This is just a kind reminder to submit your slides 2 weeks in advance ☺