Development of Accessible, Aesthetically-Pleasing Color Sequences

SciPy 2022

Matthew A. Petroff

2022-07-15

Center for Astrophysics, Harvard & Smithsonian

Outline

1. Terminology and overview of color vision and color-vision deficiencies
2. Issues with existing color sequences
3. Generating accessible color sets
4. Aesthetic-preference survey and modeling
5. Results and best practices

Terminology

Colormap Continuous, e.g., Viridis 

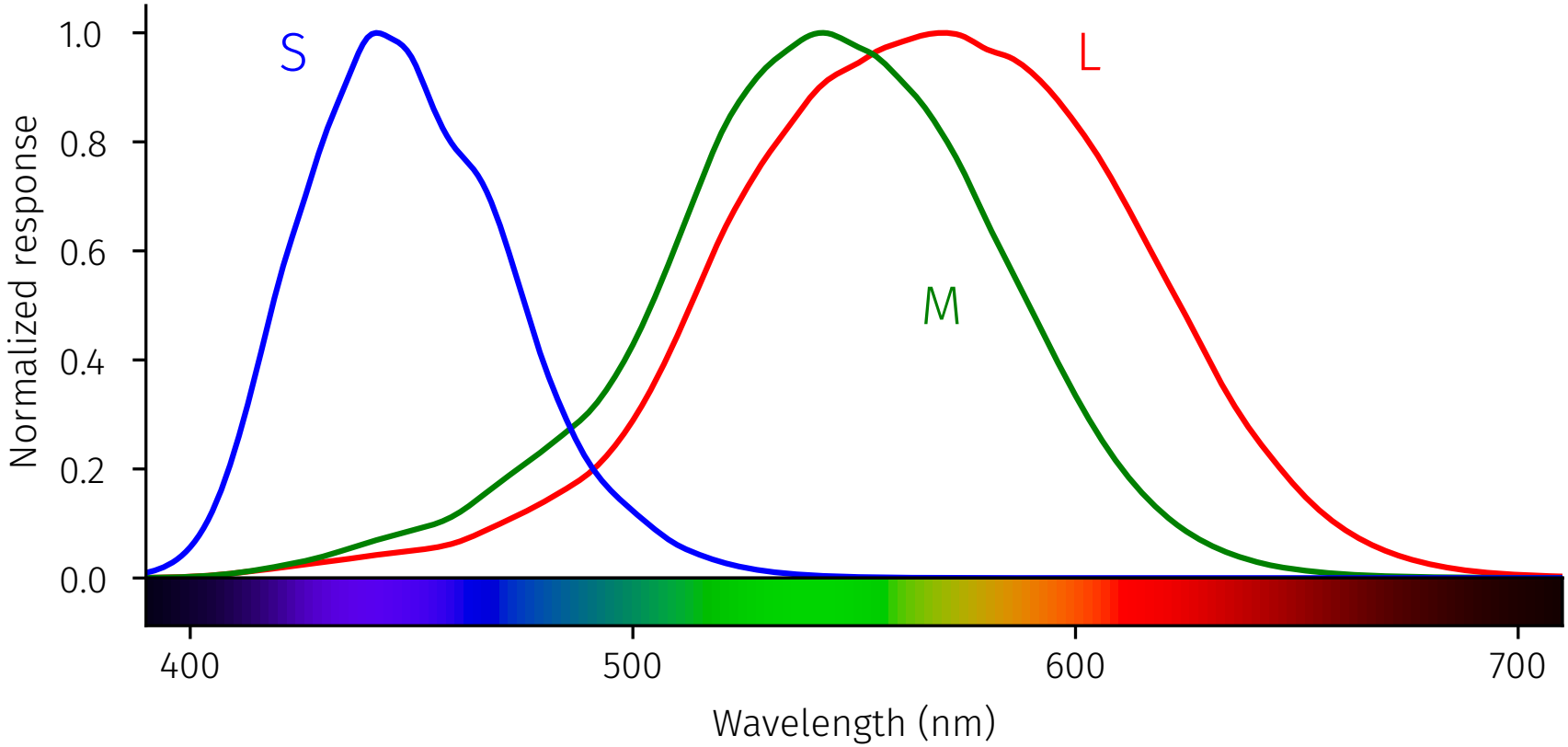
Color sequence Discrete, e.g., Category 10 

- Also known as color cycles or color palettes
- Will be used in this talk when the order of the colors matters

Color set Will be used in this talk to refer to a groups of discrete colors when order does not matter

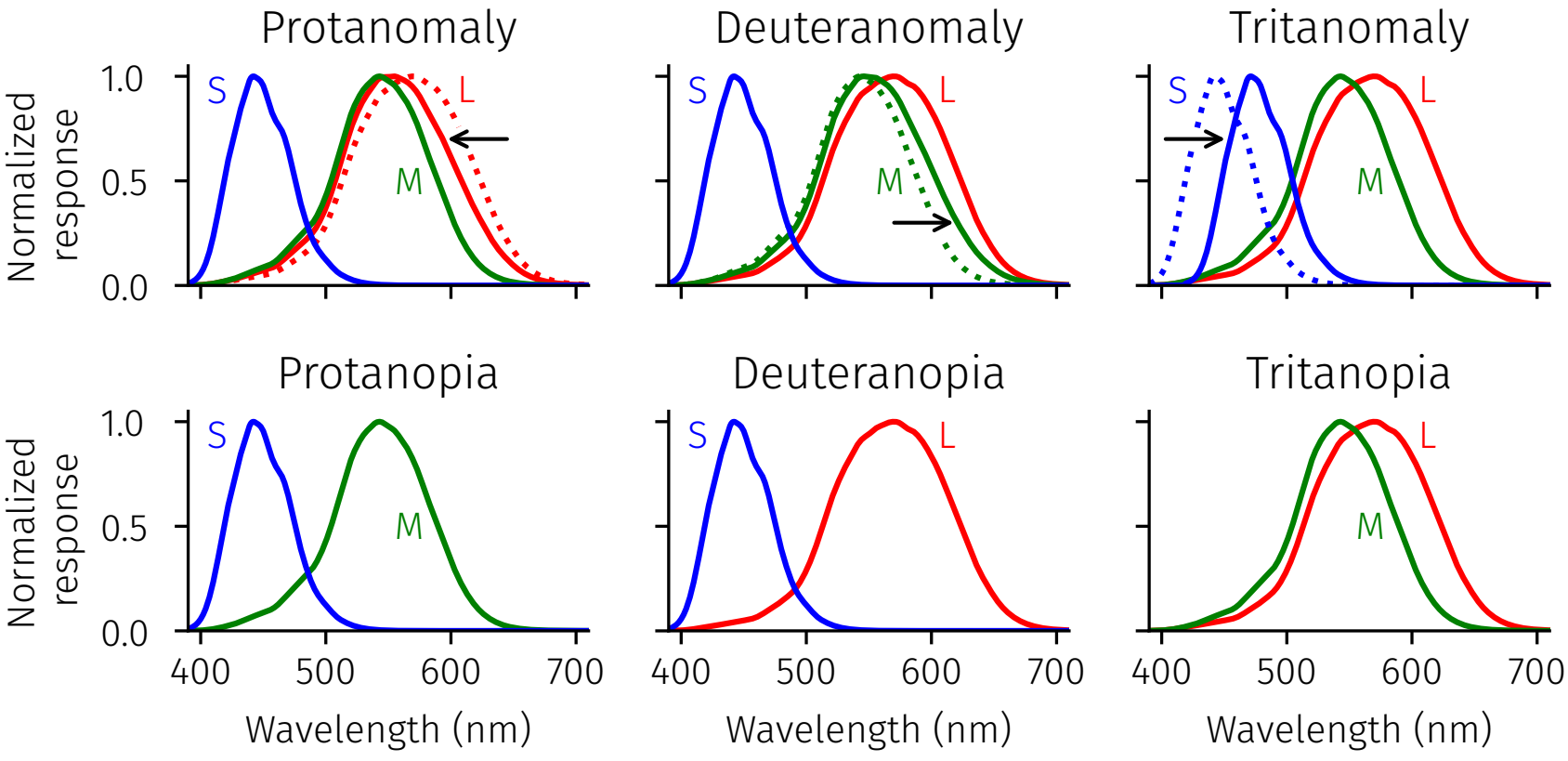
Overview of human color vision

- Photometry via cone cells sensitive to long (L), medium (M), and short (S) wavelengths
- Rod cells for low-light vision
- Higher-level functions: beyond the scope of this talk



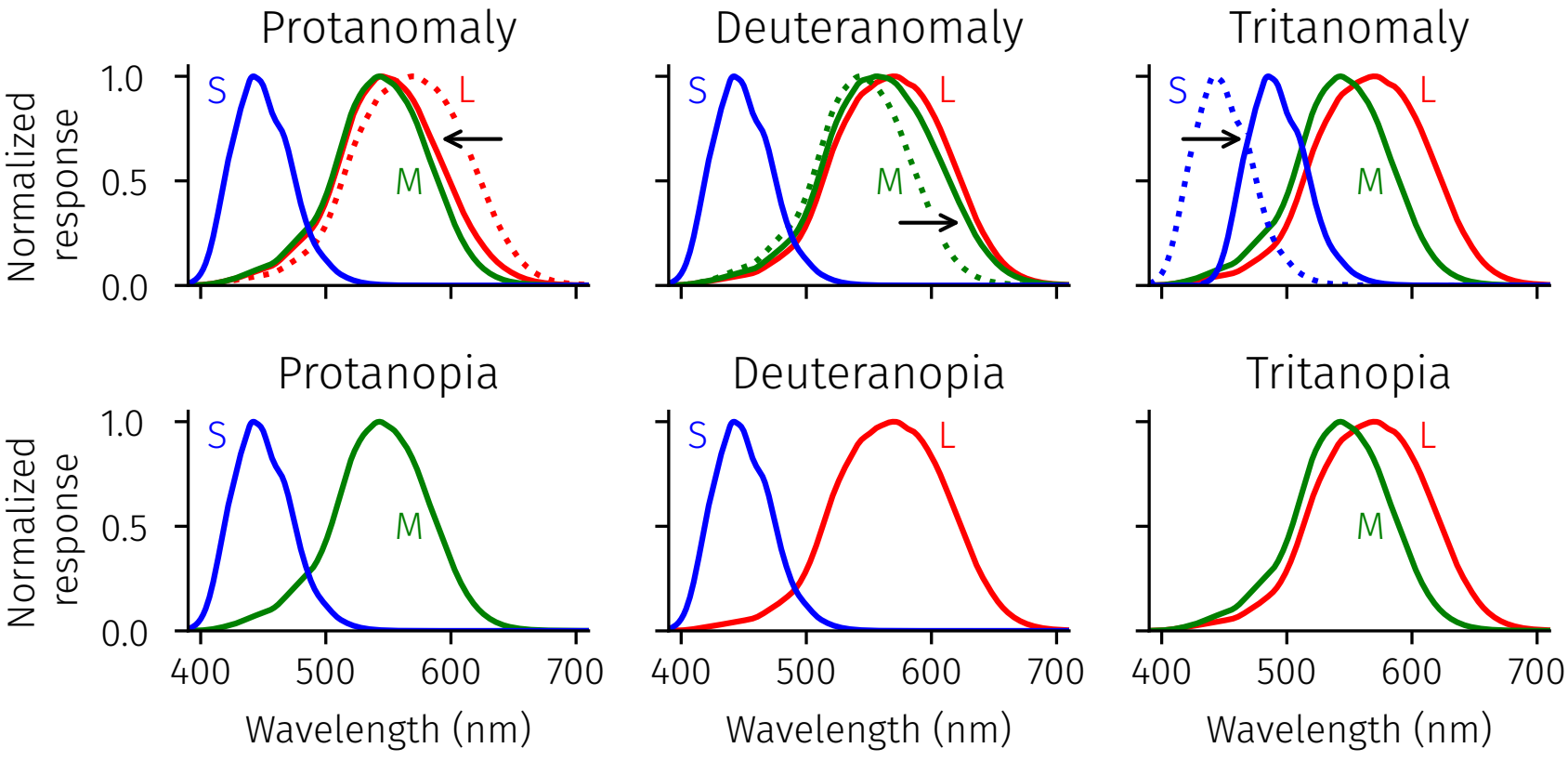
What are color-vision deficiencies?

- Also known as colorblindness
- Anomalous trichromacy: shift in spectral response of cone
- Dichromacy: missing one type of cone



What are color-vision deficiencies?

- Also known as colorblindness
- Anomalous trichromacy: shift in spectral response of cone
- Dichromacy: missing one type of cone



Calculating perceptual distance

- CAM02-UCS is a perceptually-uniform color space, which was constructed with the intent that Euclidean distance in the color space is proportional to perceived color distance
- Machado et al. (2009) presented a method of simulating color-vision deficiencies (CVDs)
- Distance between simulation results in CAM02-UCS approximates perceptual distance for CVDs
- Calculate distance between all color pairs for all CVDs and take minimum: $\min \Delta E_{\text{cvd}}$

Perceptual distance example

Category 10 (Matplotlib default)

Color	1	min $\Delta E'$
1 	—	100.0

Perceptual distance example

Category 10 (Matplotlib default)

Color	1	2	min $\Delta E'$
1	—		100.0
2	65.7	—	65.7

Perceptual distance example

Category 10 (Matplotlib default)

Color	1	2	3	min $\Delta E'$
1	—			100.0
2	65.7	—		65.7
3	48.2	46.8	—	46.8

Perceptual distance example

Category 10 (Matplotlib default)

Color	1	2	3	4	5	6	7	8	9	10	min $\Delta E'$
1	—										100.0
2	**65.7**	—									65.7
3	48.2	**46.8**	—								46.8
4	63.0	**26.1**	59.2	—							26.1
5	26.9	53.0	57.7	44.6	—						26.1
6	41.9	33.4	42.5	**23.7**	31.3	—					23.7
7	46.3	38.1	60.3	32.9	**22.9**	32.1	—				22.9
8	26.7	39.4	32.6	40.2	25.3	**20.2**	31.7	—			20.2
9	60.4	29.0	25.1	50.6	59.8	43.6	52.7	37.7	—		20.2
10	27.0	59.7	37.0	67.9	41.1	49.8	49.6	29.6	45.3	—	20.2

Issues with existing color sequences

Category 10 (Matplotlib default)

Typical	min $\Delta E'$
	100.0
	65.7
	46.8
	26.1
	26.1
	23.7
	22.9
	20.2
	20.2
	20.2

Issues with existing color sequences

Category 10 (Matplotlib default)

Typical	min $\Delta E'$	Deut	min ΔE_{deut}	Prot	min ΔE_{prot}	Trit	min ΔE_{trit}	min ΔE_{cvd}
	100.0		100.0		100.0		100.0	100.0
	65.7		61.2		54.1		59.9	54.1
	46.8		16.7		3.4		13.4	3.4
	26.1		5.2		3.4		13.4	3.4
	26.1		5.2		2.0		13.4	2.0
	23.7		5.2		2.0		13.4	2.0
	22.9		5.2		2.0		12.7	2.0
	20.2		5.2		2.0		11.1	2.0
	20.2		4.5		2.0		11.1	2.0
	20.2		4.3		2.0		11.1	2.0

 clipped to displayable color gamut

Other considerations

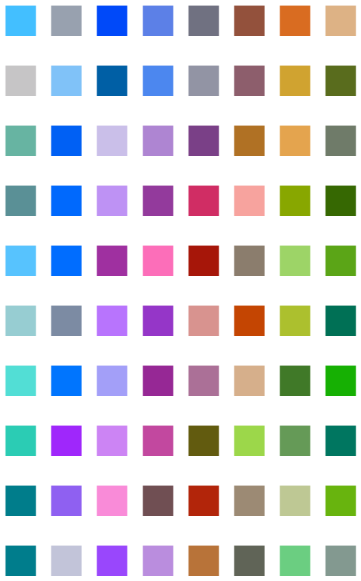
- Minimum contrast
  - Colors that are too light are difficult to see
  - Colors that are too dark have increased perceptual weight
- Grayscale compatibility
  - Handle black & white printing and e-paper displays
  - Improve accessibility for monochromacy
  - Trade off with color accessibility
- Color names
  - Favor easy-to-name colors, which have high color saliency
  - Avoid repeated basic color terms

Enforcing perceptual distance constraints

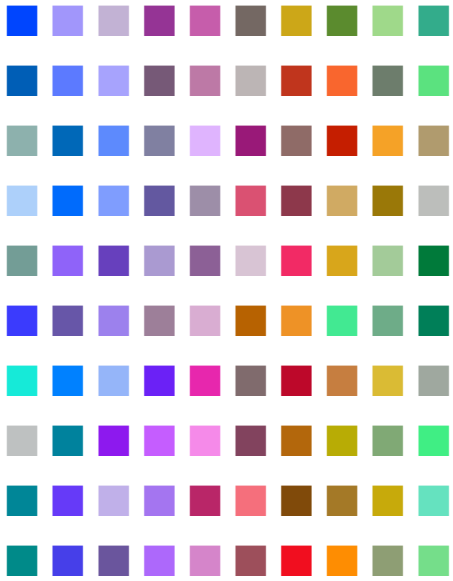
1. Start with random sRGB starting color
2. Randomly pick another color
3. If $\min \Delta E_{\text{cvd}}$ and $\min \Delta J'$ are greater than threshold, add new color to list
4. Repeat steps 2 and 3 until list is desired length or no sRGB color remain (restart from 1 in this case)

Six

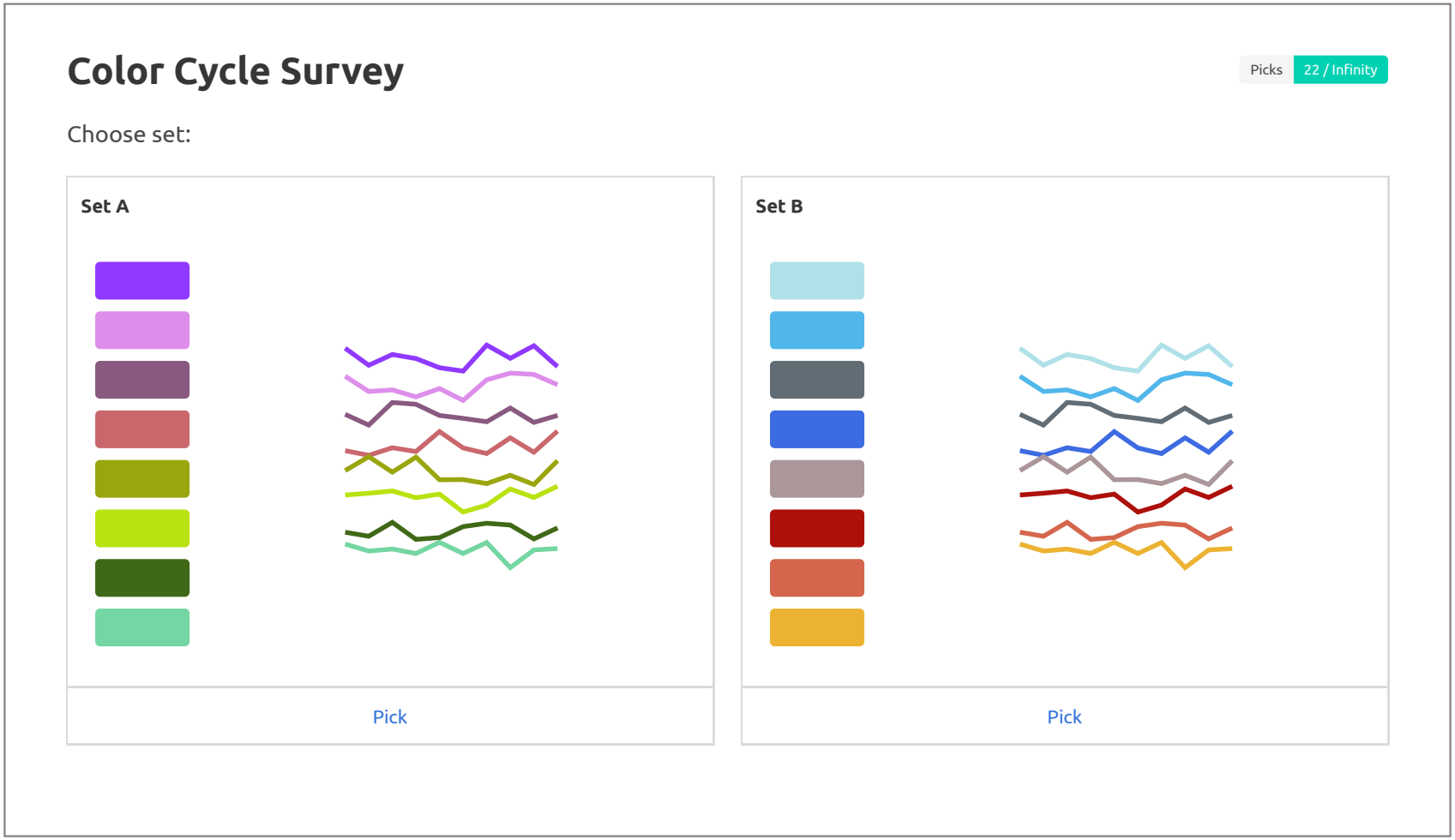
Eight



Ten



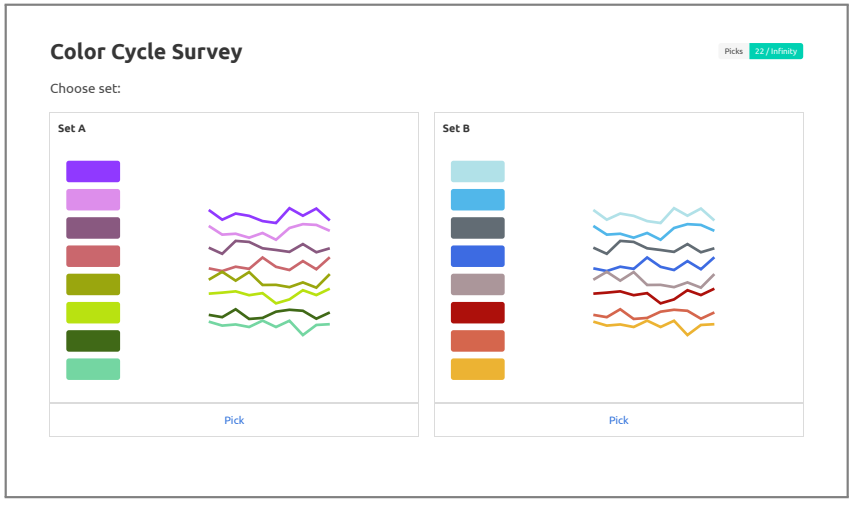
User survey for aesthetic preference



User survey for aesthetic preference

User survey for aesthetic preference

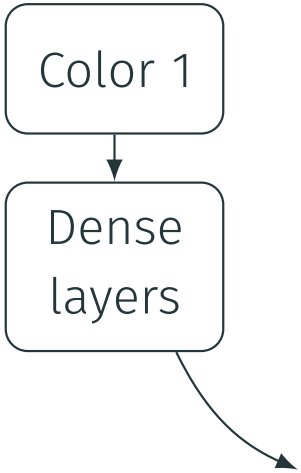
- Run December 2018 through December 2020
- ~22k responses
- ~2.2k user sessions
- Thanks to all those who participated!



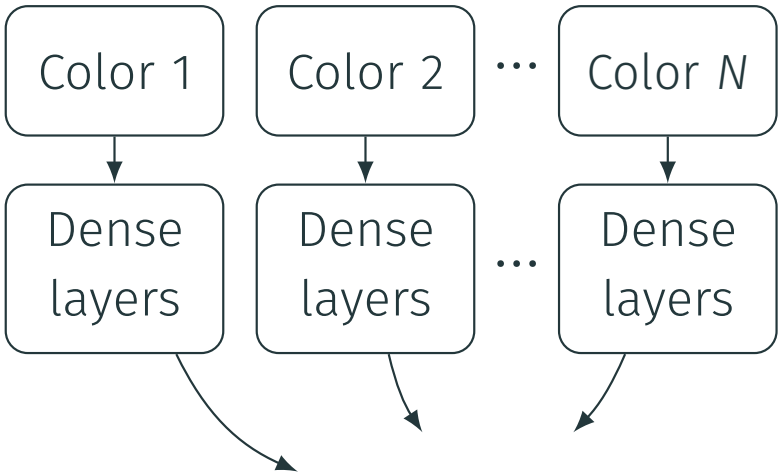
Modeling aesthetic preference

- Machine learning using artificial neural networks
- Sets and sequences handled separately
- Issues to address
  - How to handle color sets / sequences with variable length?
  - How to train a model that produces a numeric score with pairwise binary survey responses?
  - How to fully utilize limited, noisy data?

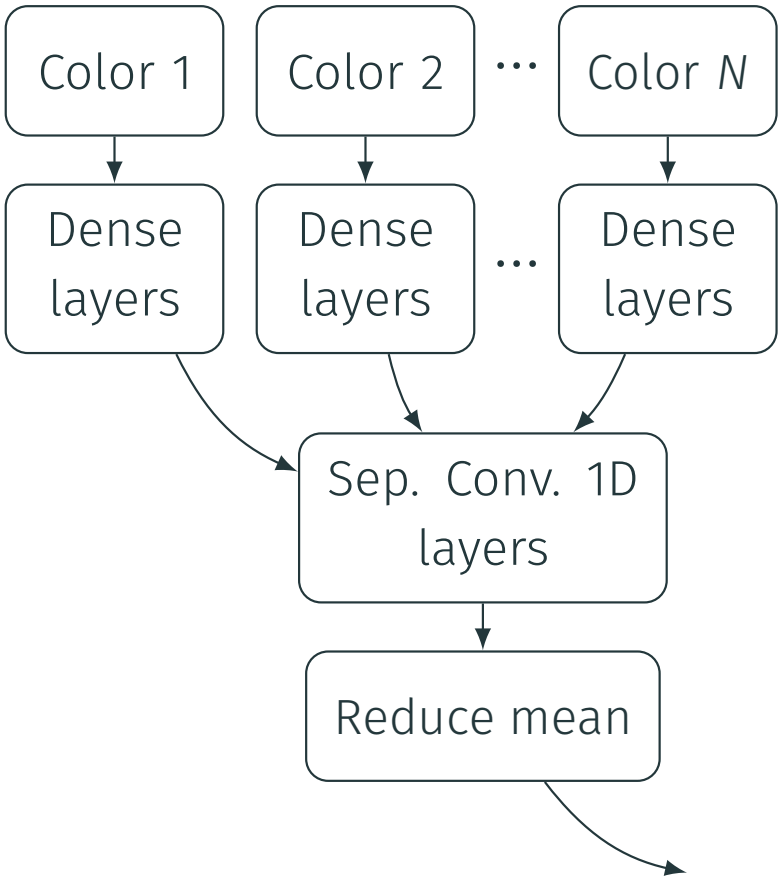
Modeling aesthetic preference



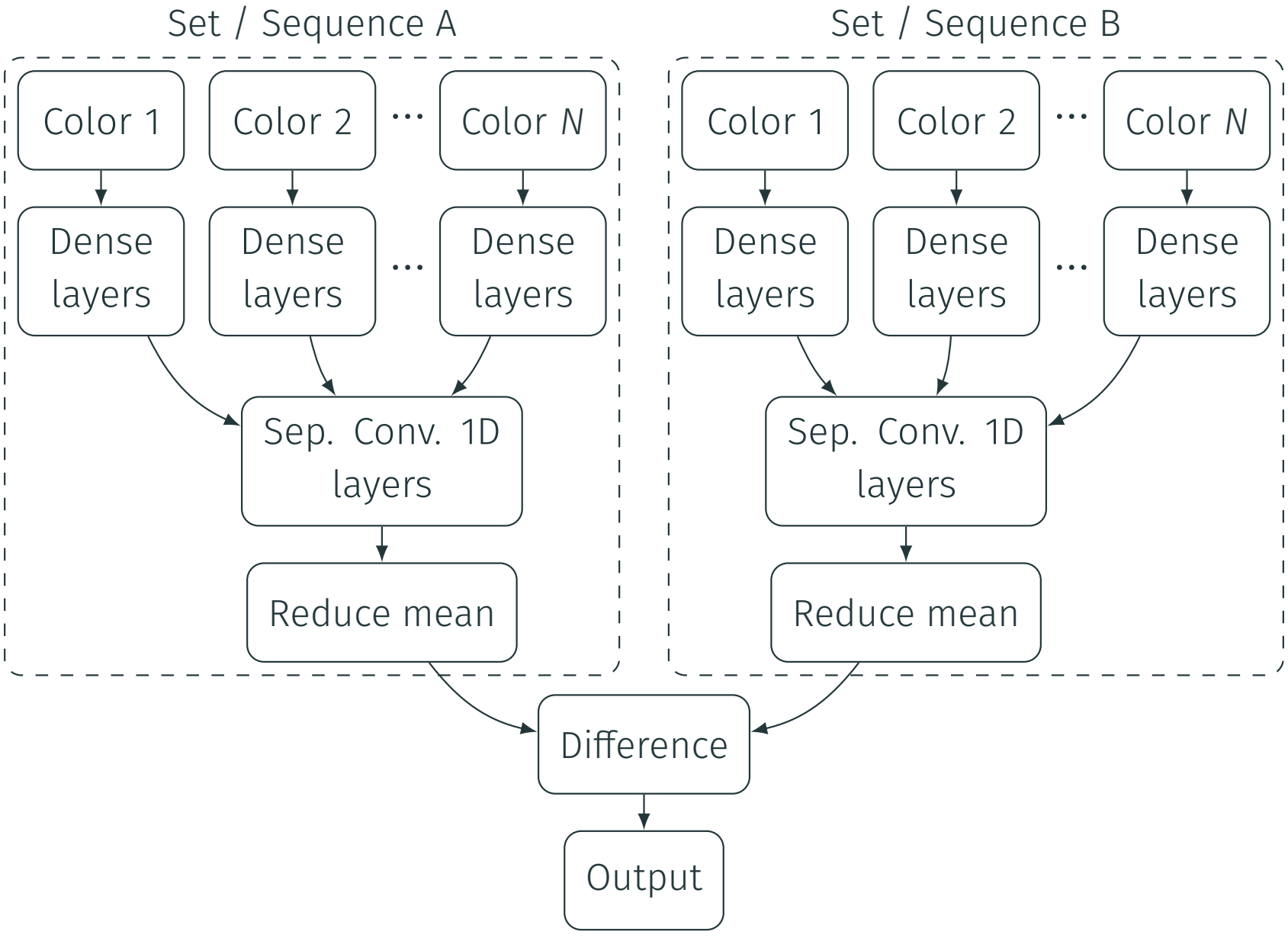
Modeling aesthetic preference



Modeling aesthetic preference



Modeling aesthetic preference



Modeling aesthetic preference

- Bootstrap aggregation used to maximize use of training data while still checking for overfitting
  - Data randomly split into train and test portions multiple times
  - Each train split used to train a separate model copy; test split used to check for overfitting
  - Model outputs averaged for final output
- Simultaneous training on six- and eight-color sets / sequences
- Ten-color sets / sequences used for model validation

Final metric

- Color sets
  - Multiply aesthetic model score with saliency score
- Color sequences
  - Accessibility score: favors orderings with larger perceptual and lightness differences and darker colors toward the beginning
  - Only use orderings where the first color matches the first color of the sequence most favored by the aesthetic model
  - Orderings where basic color terms are repeated earlier than necessary are eliminated
  - Multiply aesthetic model score with accessibility score

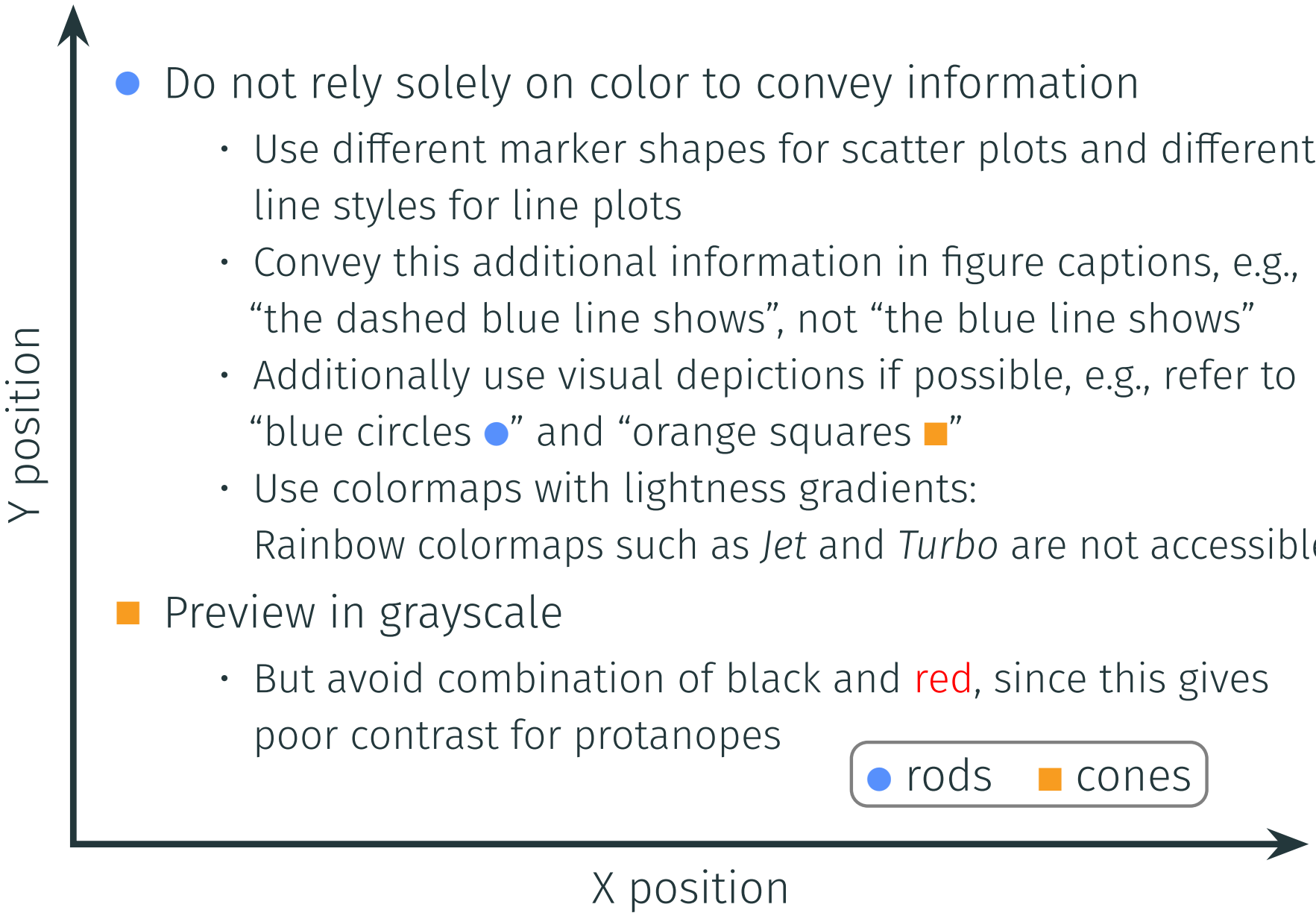
Results

Six	min ΔE_{cvd}	Eight	min ΔE_{cvd}	Ten	min ΔE_{cvd}
	100.0		100.0		100.0
	57.1		66.9		56.8
	21.3		18.2		33.4
	21.3		18.1		22.3
	21.3		18.1		18.3
	20.5		18.1		16.4
			18.1		16.3
			18.1		16.1
					16.1
					16.1

Best practices

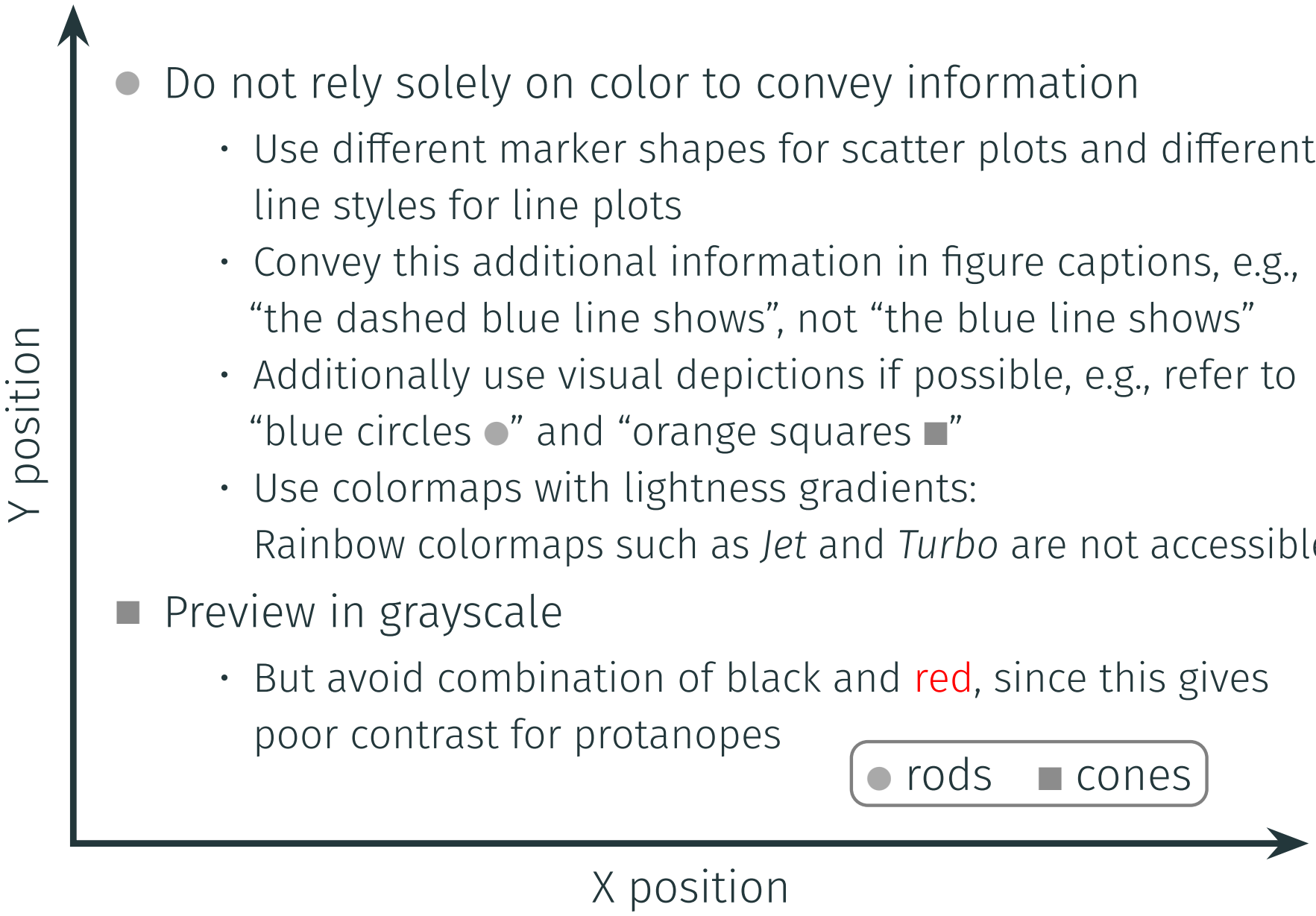
- Do not rely solely on color to convey information
  - Use different marker shapes for scatter plots and different line styles for line plots
  - Convey this additional information in figure captions, e.g., “the dashed blue line shows”, not “the blue line shows”
  - Additionally use visual depictions if possible, e.g., refer to “blue circles ●” and “orange squares ■”
  - Use colormaps with lightness gradients:
    Rainbow colormaps such as *Jet* and *Turbo* are not accessible
- Preview in grayscale
  - But avoid combination of black and red, since this gives poor contrast for protanopes

Best practices



- Do not rely solely on color to convey information
  - Use different marker shapes for scatter plots and different line styles for line plots
  - Convey this additional information in figure captions, e.g., “the dashed blue line shows”, not “the blue line shows”
  - Additionally use visual depictions if possible, e.g., refer to “blue circles ●” and “orange squares ■”
  - Use colormaps with lightness gradients:
    Rainbow colormaps such as *Jet* and *Turbo* are not accessible
- Preview in grayscale
  - But avoid combination of black and red, since this gives poor contrast for protanopes

Best practices



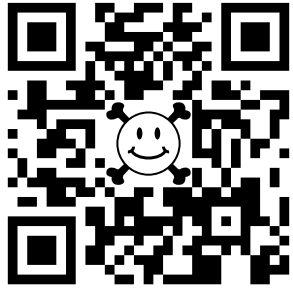
- Do not rely solely on color to convey information
  - Use different marker shapes for scatter plots and different line styles for line plots
  - Convey this additional information in figure captions, e.g., “the dashed blue line shows”, not “the blue line shows”
  - Additionally use visual depictions if possible, e.g., refer to “blue circles ● ” and “orange squares ■ ”
  - Use colormaps with lightness gradients:
    Rainbow colormaps such as *Jet* and *Turbo* are not accessible
- Preview in grayscale
  - But avoid combination of black and red, since this gives poor contrast for protanopes

Conclusions

- New color-vision-deficiency-friendly color sequences developed using algorithmically-enforced accessibility constraints and machine-learning-based aesthetic-preference model
- Pre-print:
  Accessible Color Sequences for Data Visualization
  `https://arxiv.org/abs/2107.02270`
- Code & data (including hex triplets):
  `https://github.com/mpetroff/accessible-color-cycles`
- Aspirational goal: accessible defaults

Conclusions

- New color-vision-deficiency-friendly color sequences developed using algorithmically-enforced accessibility constraints and machine-learning-based aesthetic-preference model
- Pre-print:
  Accessible Color Sequences for Data Visualization
  `https://arxiv.org/abs/2107.02270`
- Code & data (including hex triplets):
  `https://github.com/mpetroff/accessible-color-cycles`
- Aspirational goal: accessible defaults



Questions?