SIGGRAPH THINK BEYOND

2020 S2020.SIGGRAPH.ORG

SEQUENTIAL GALLERY FOR INTERACTIVE VISUAL DESIGN OPTIMIZATION

Yuki Koyama, Issei Sato, and Masataka Goto

Sequential Gallery for Interactive Visual Design Optimization

Yuki Koyama[1], Issei Sato[2], and Masataka Goto[1]

1. National Institute of Advanced Industrial Science and Technology (AIST)
2. The University of Tokyo

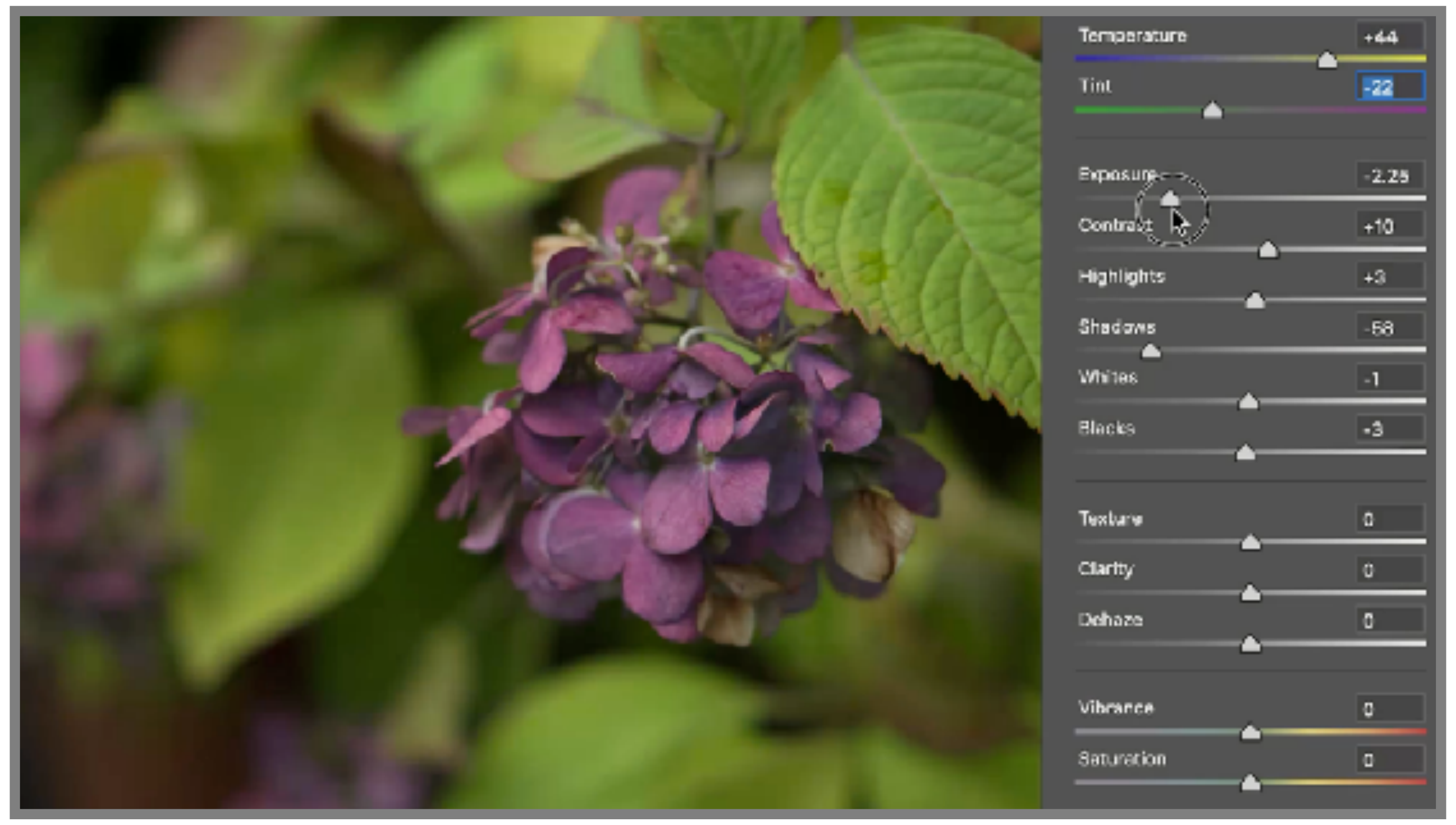
SIGGRAPH THINK BEYOND
2020 19-23 JULY WASHINGTON DC

Overview

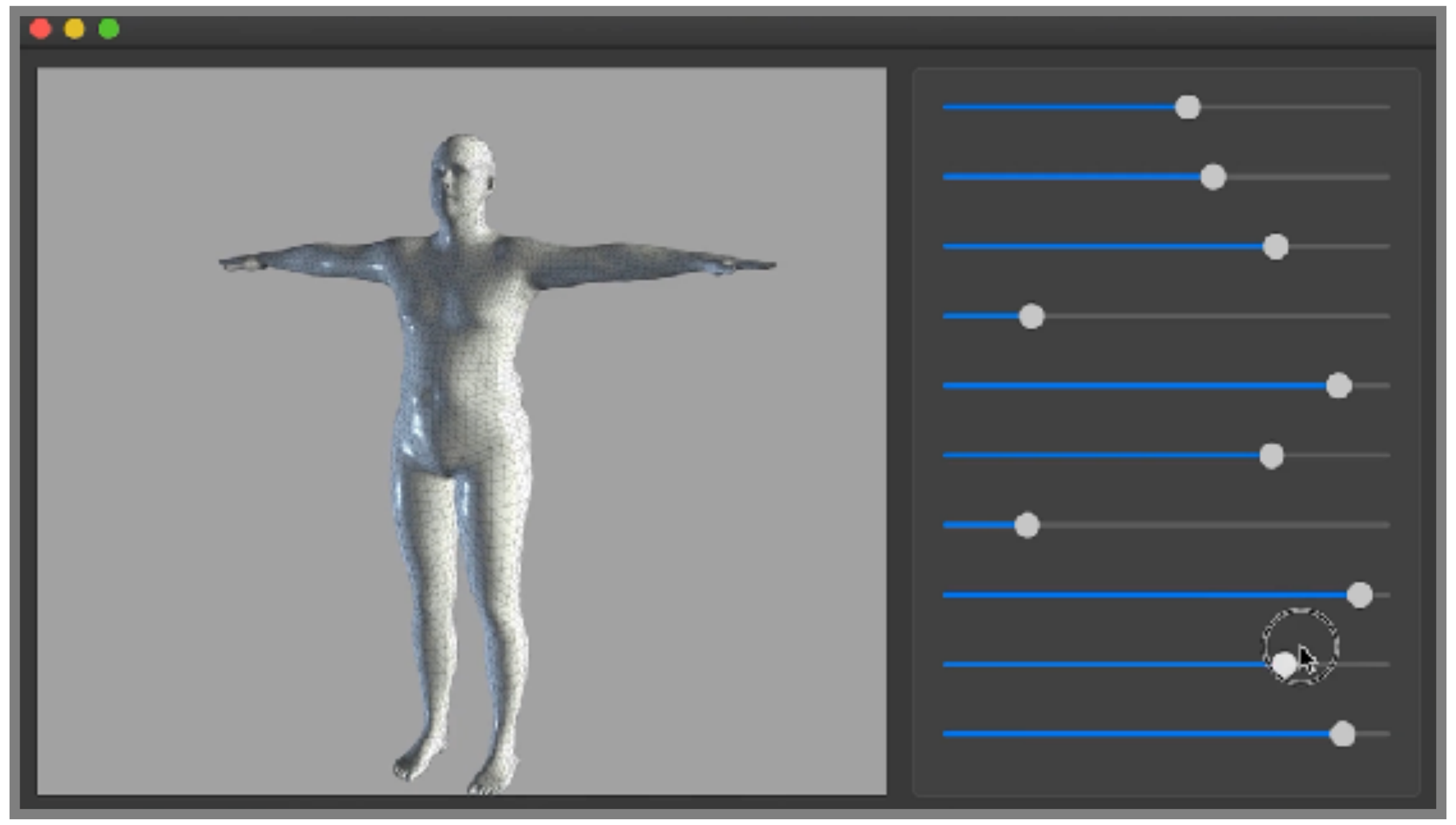
Background and Target Problem

Background: Parametric Visual Design is Everywhere

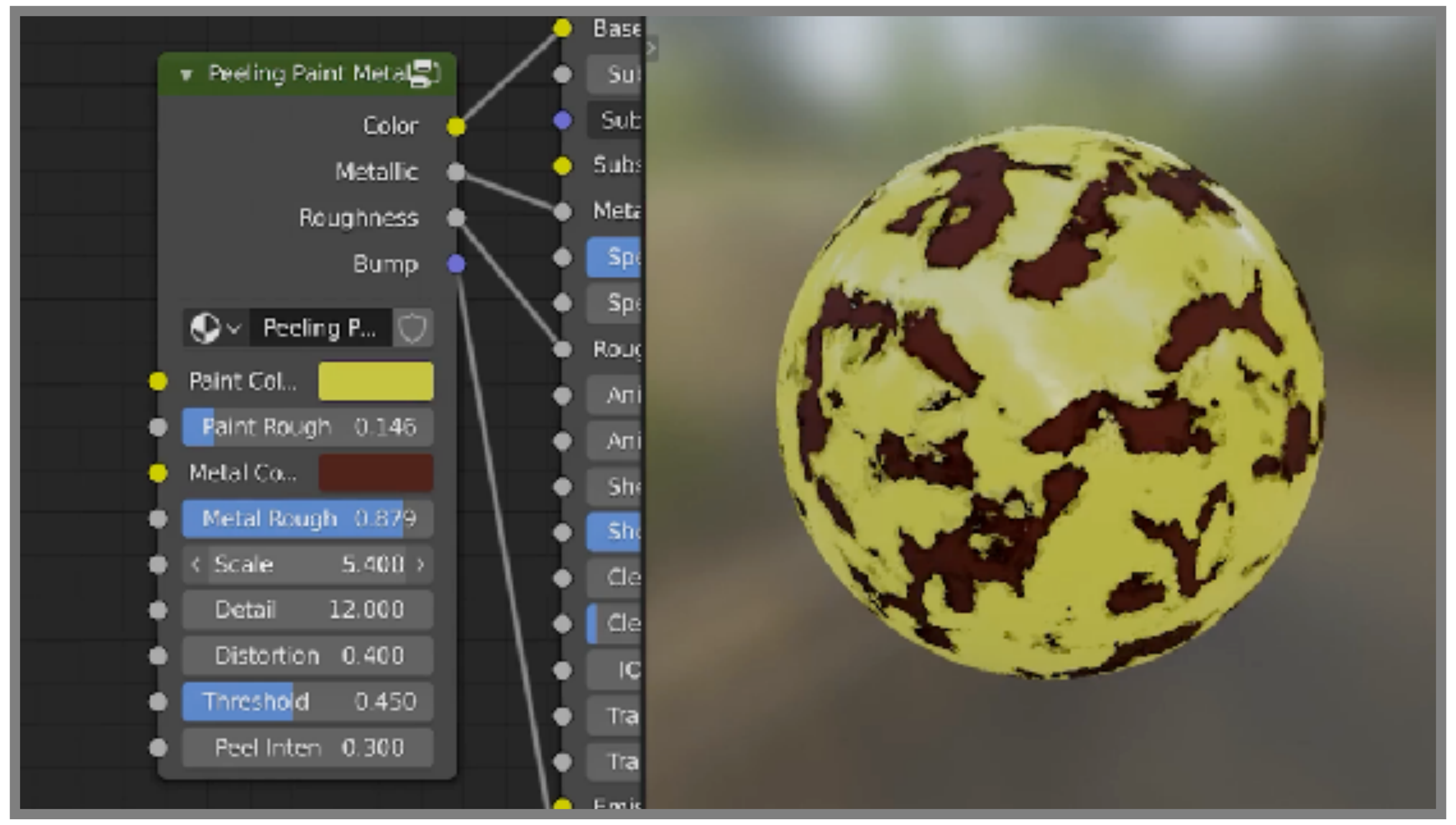
Photo color enhancement



Generative modeling

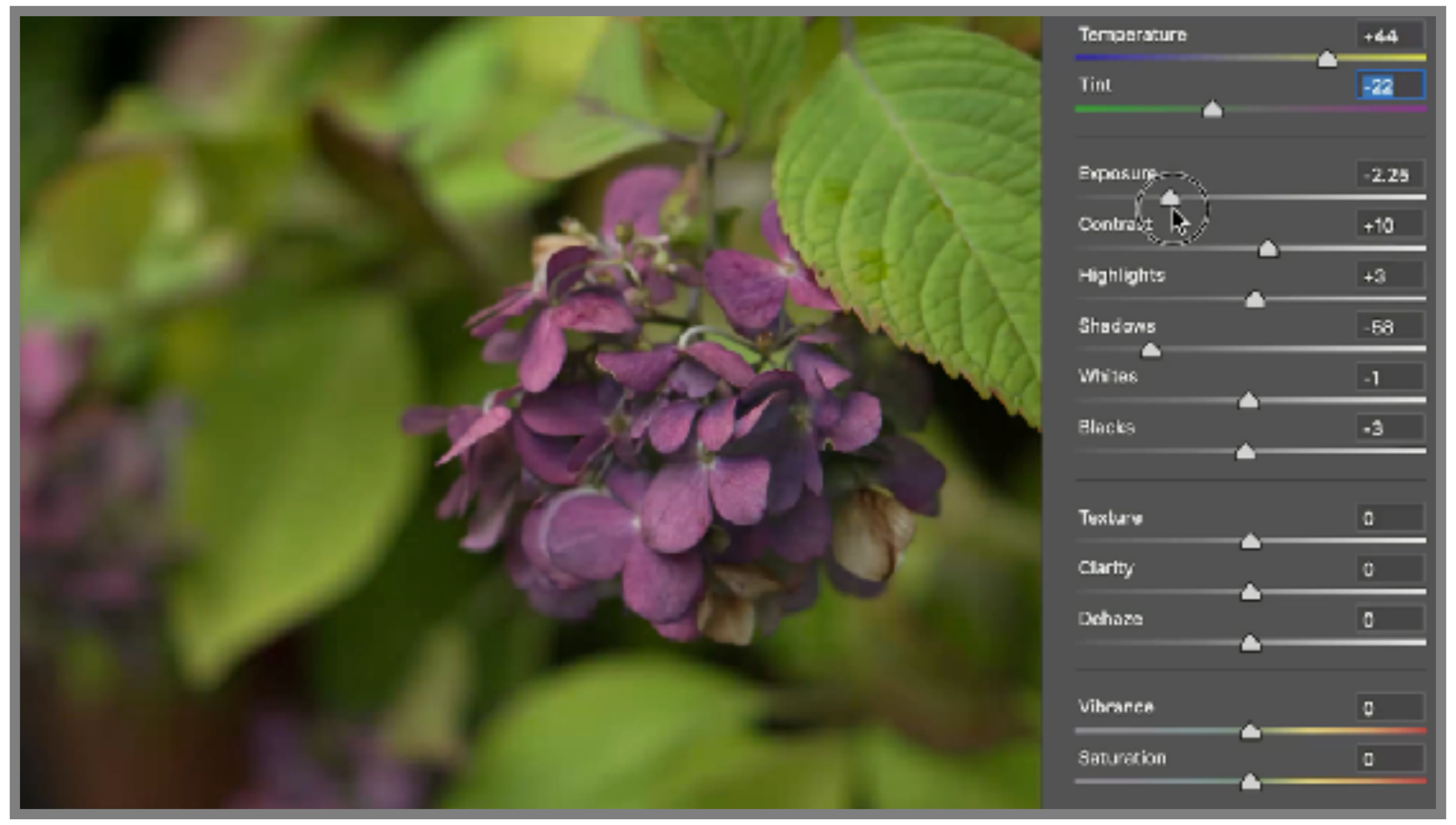


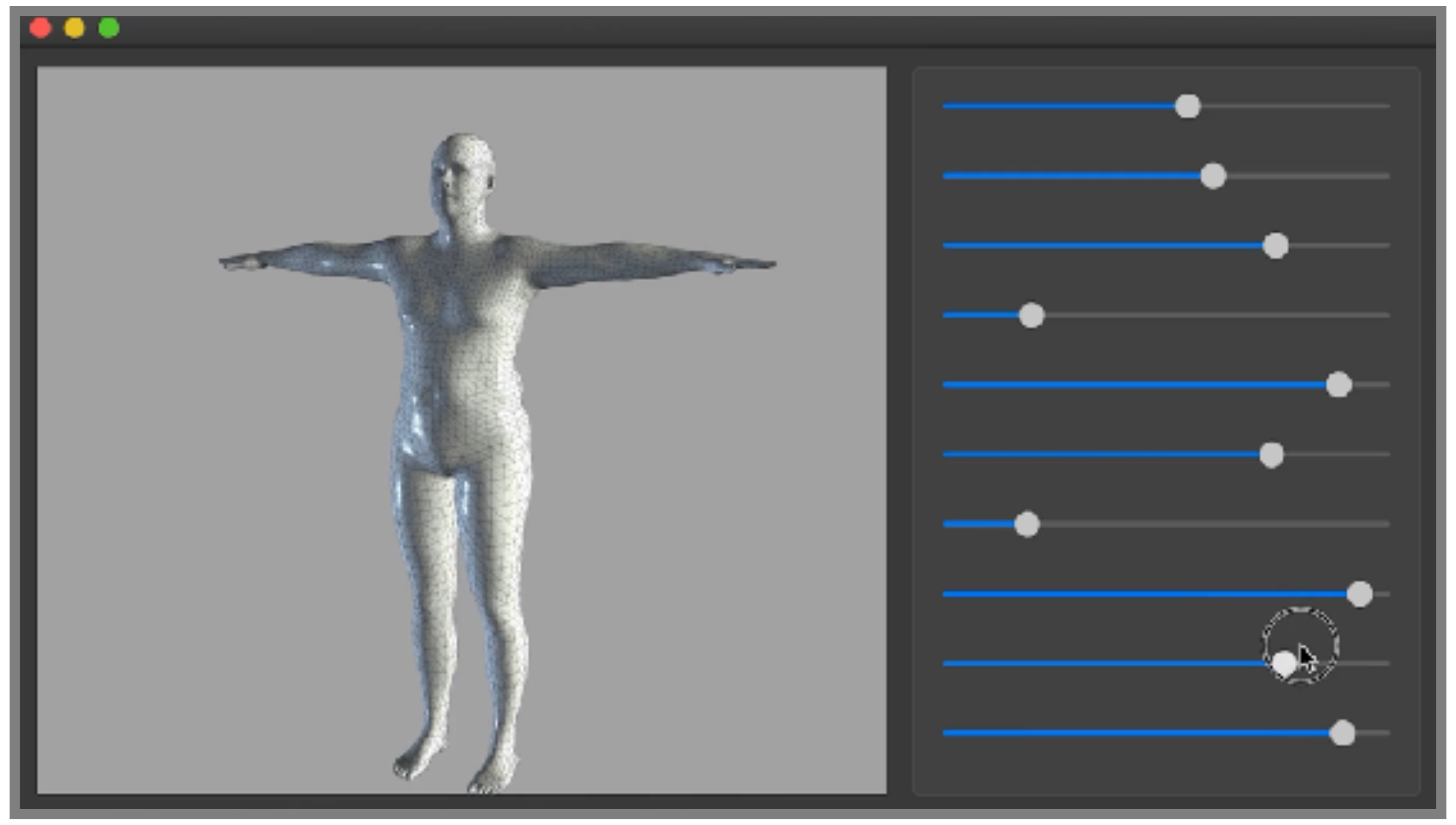
Procedural design

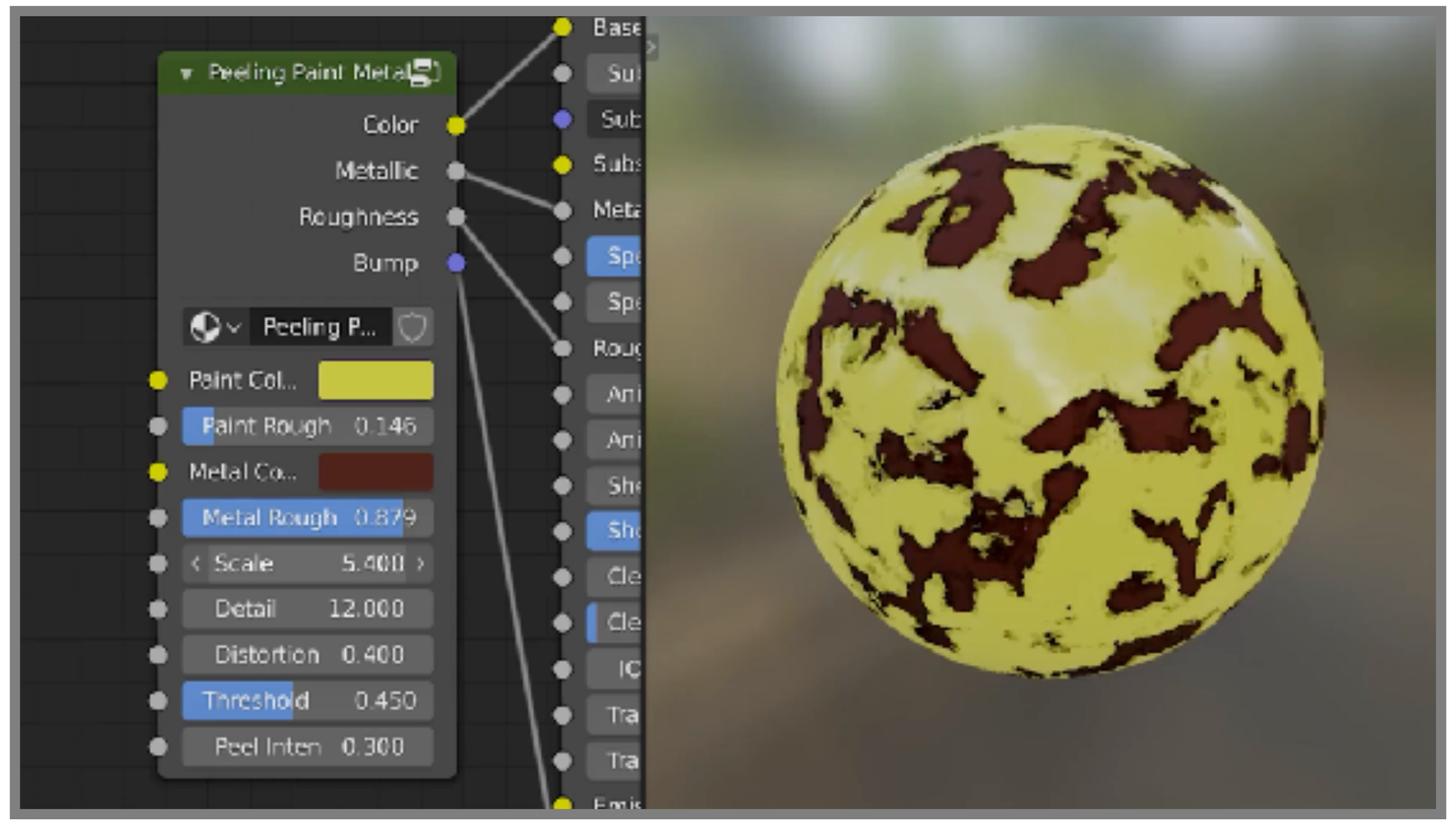


… etc.

Background: Parametric Visual Design is Everywhere





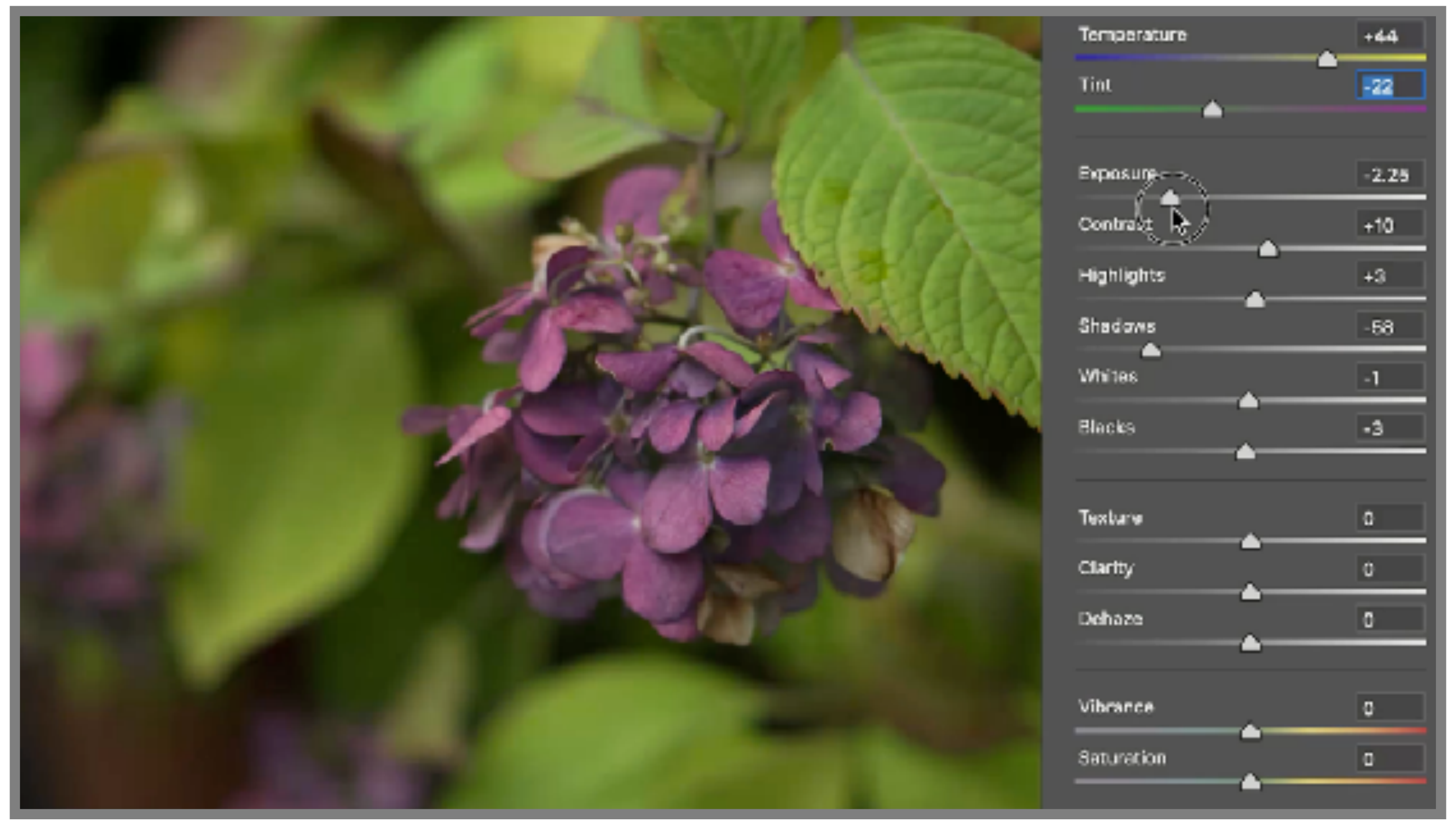


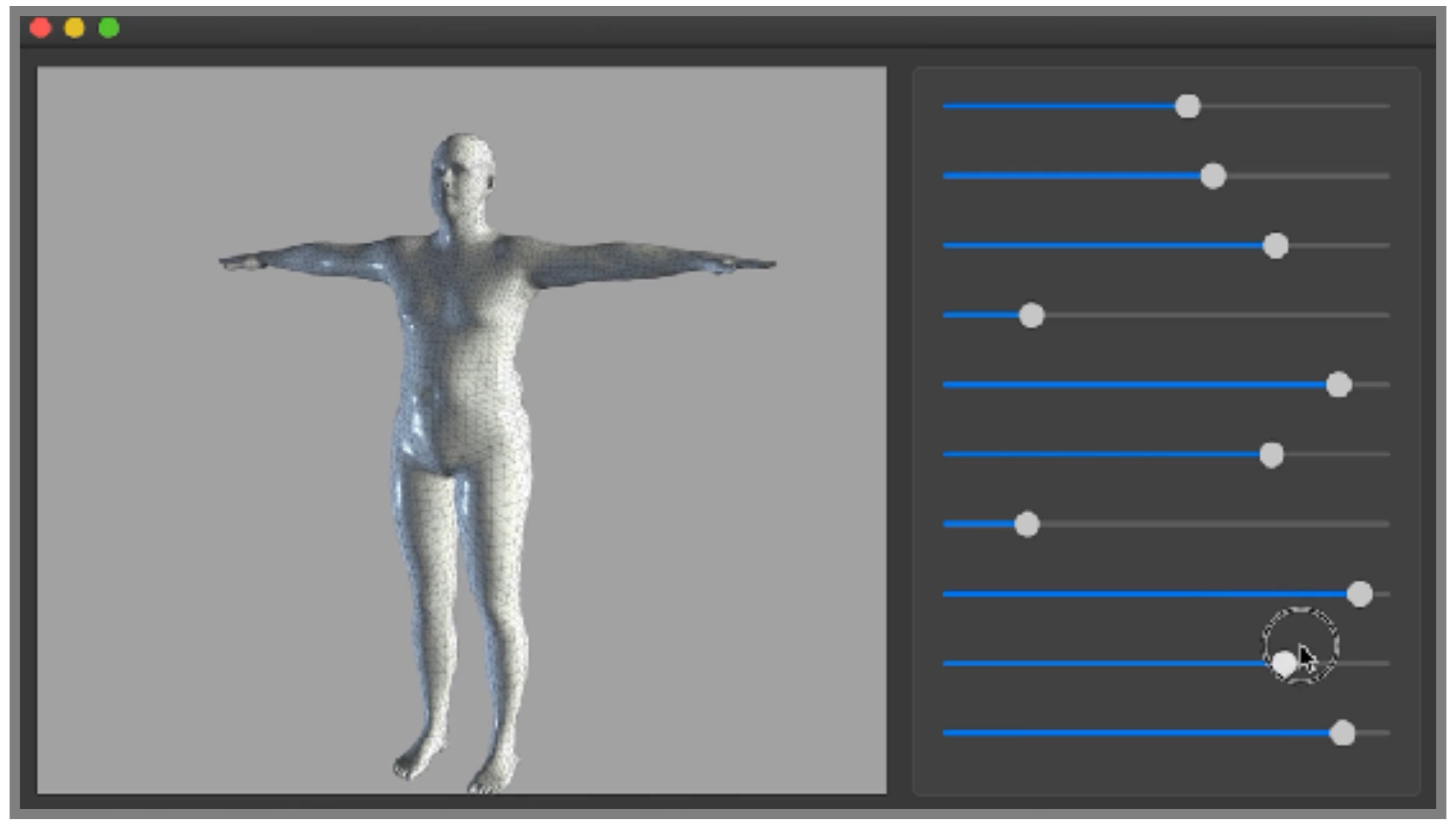
Problem: Need many **trials and errors**

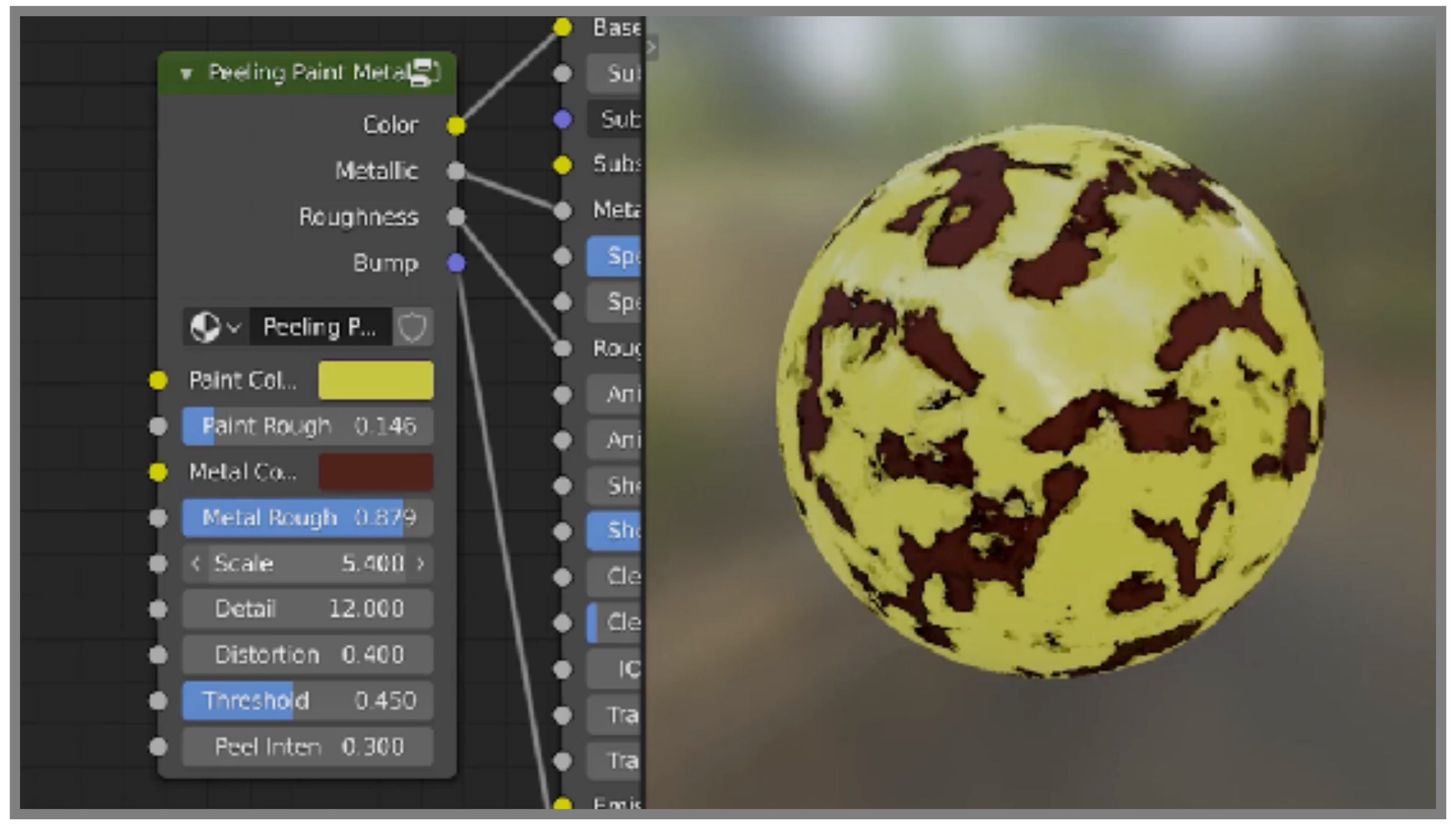
- E.g., try a different parameter combination, see the result, judge whether it is better or not, and then decide which combination to try next …
- This is a high-dimensional search task and can be **tedious and time-consuming**

Yuki Koyama, Issei Sato, and Masataka Goto. Sequential Gallery for Interactive Visual Design Optimization. ACM Trans. Graph. (SIGGRAPH 2020)

Background: Parametric Visual Design is Everywhere







Motivation: What computational support is possible?

Technical challenge:

- Parameters need to be determined based on the user's **preference**
- It is difficult to fully automate the parameter tweaking process

Yuki Koyama, Issei Sato, and Masataka Goto. Sequential Gallery for Interactive Visual Design Optimization. ACM Trans. Graph. (SIGGRAPH 2020)

Overview

Proposed System: Sequential Gallery

Target:
n design parameters
(e.g., photo enhance)

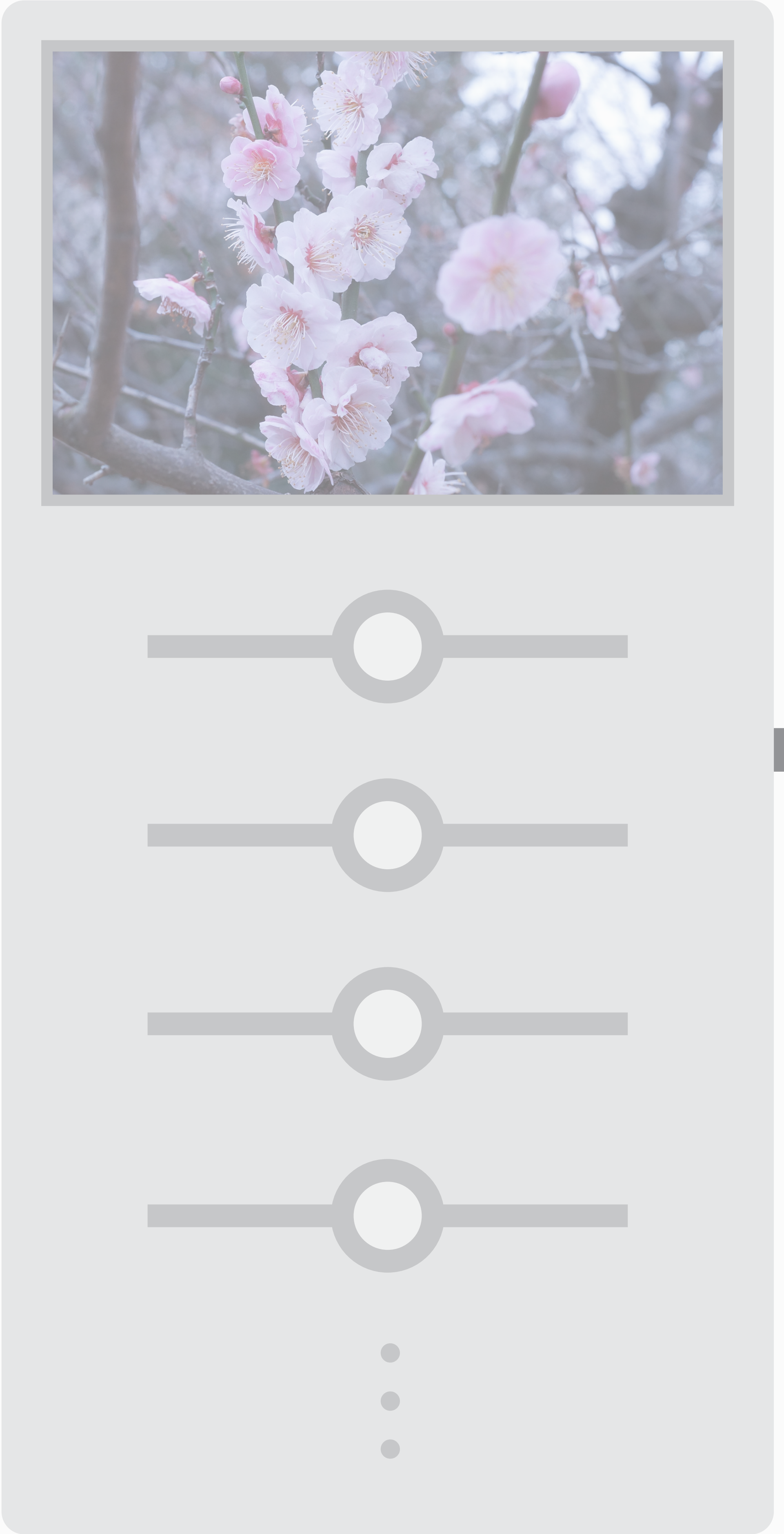
Output:
An optimal
parameter set

Target:
n design parameters
(e.g., photo enhance)

Sequential Gallery:
An interactive optimization framework
where the user sequentially performs
2D search subtasks via a grid interface

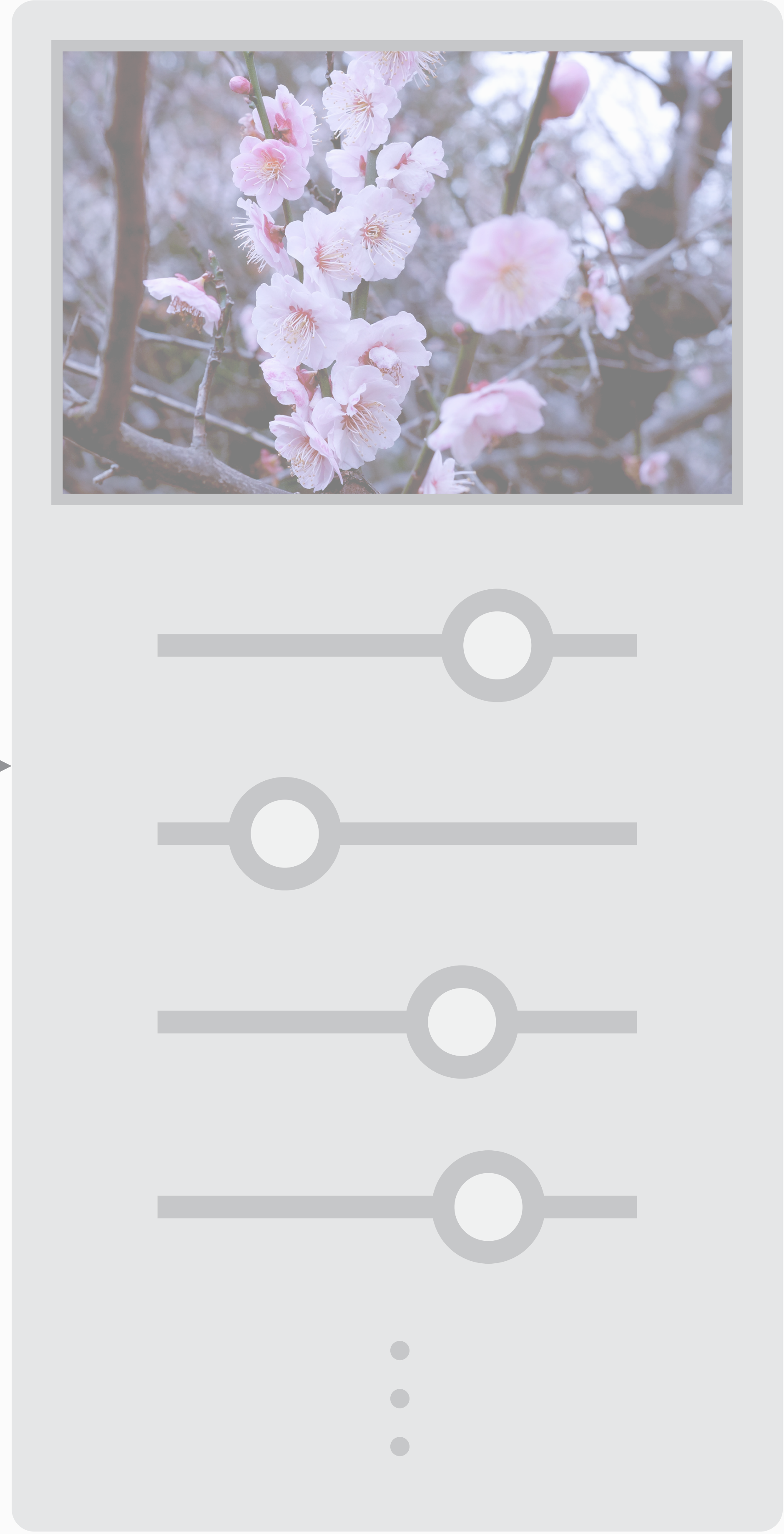
Output:
An optimal
parameter set

Target:
n design parameters
(e.g., photo enhance)

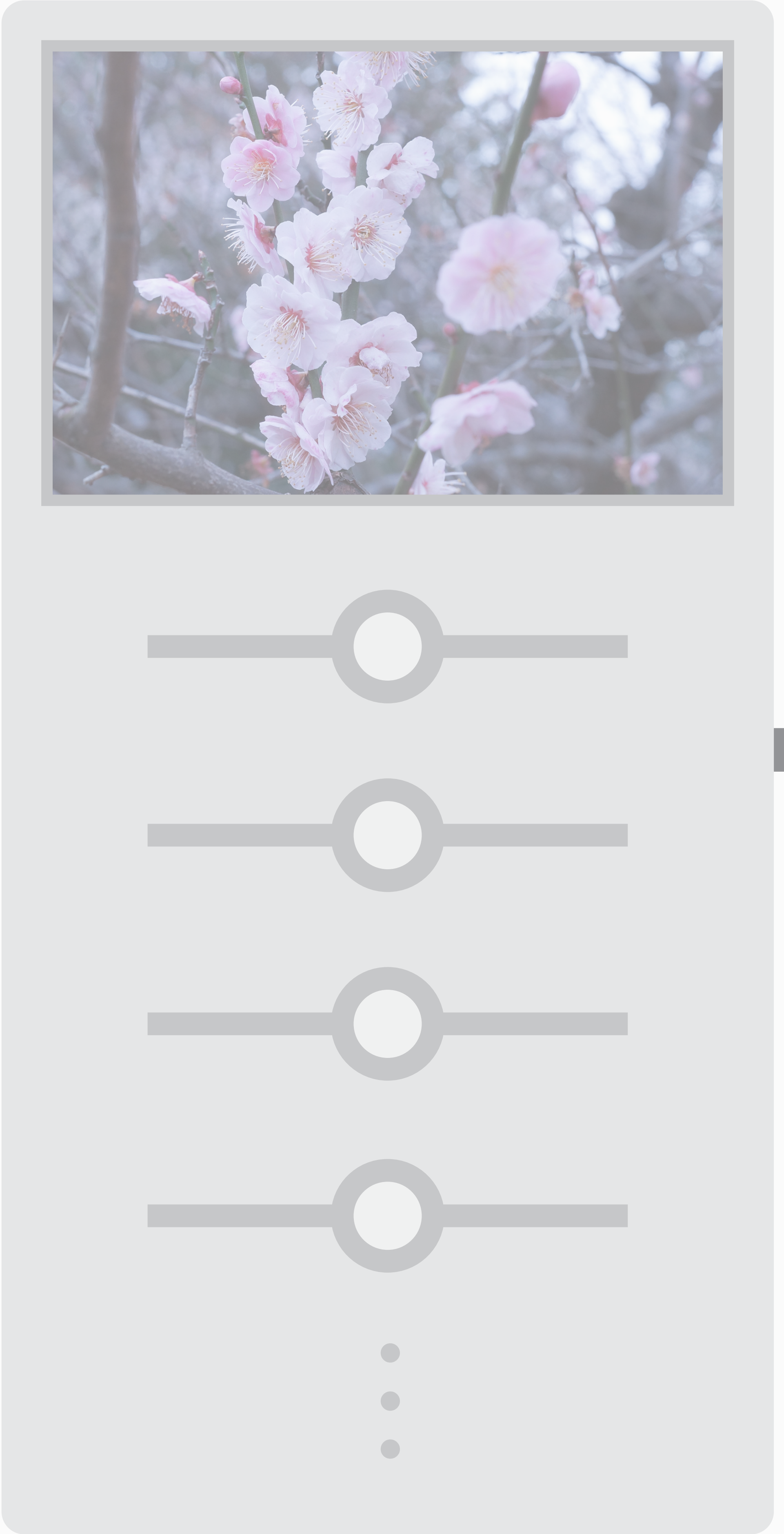


Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface

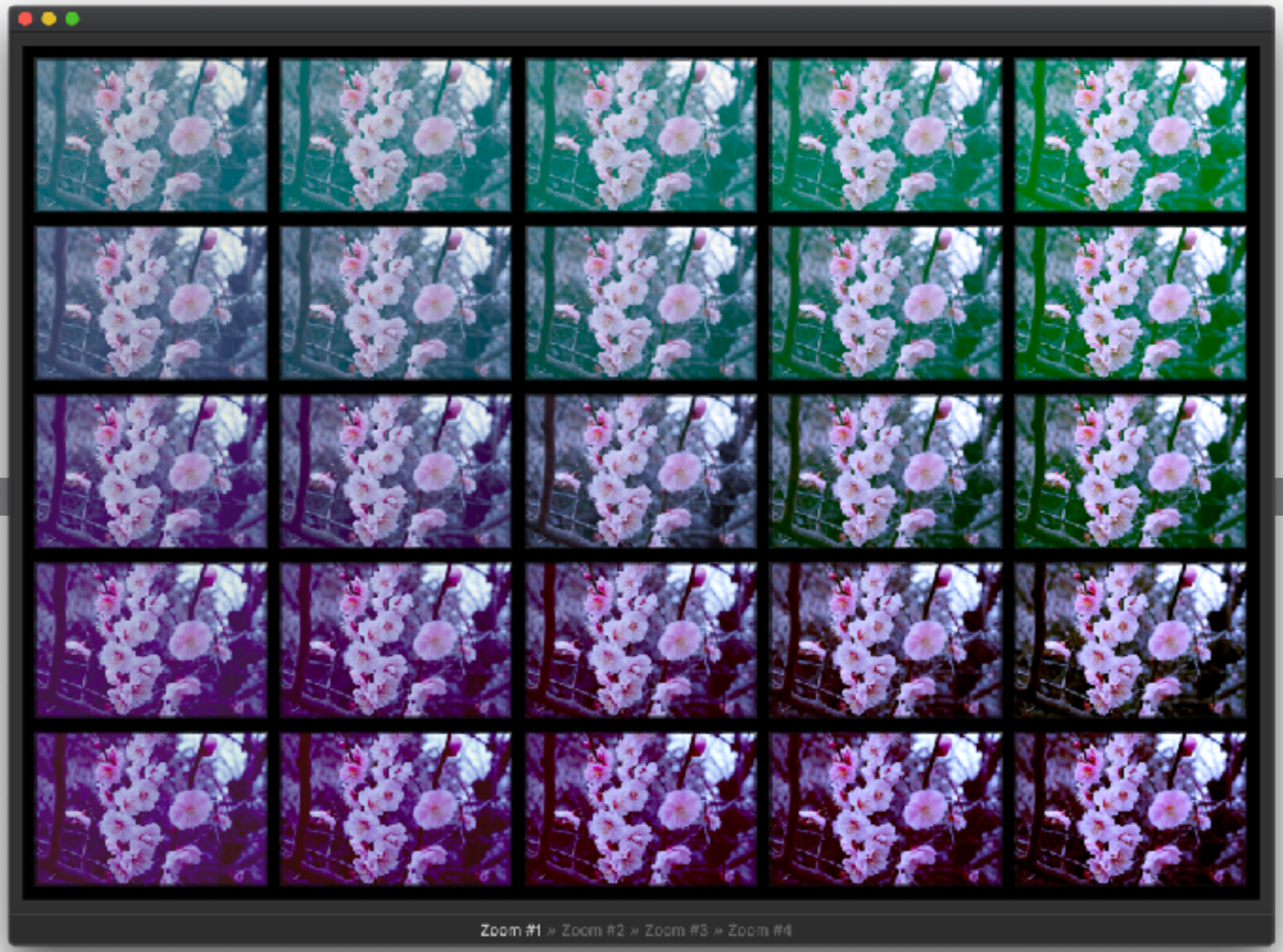
Output:
An optimal parameter set



Target:
n design parameters
(e.g., photo enhance)

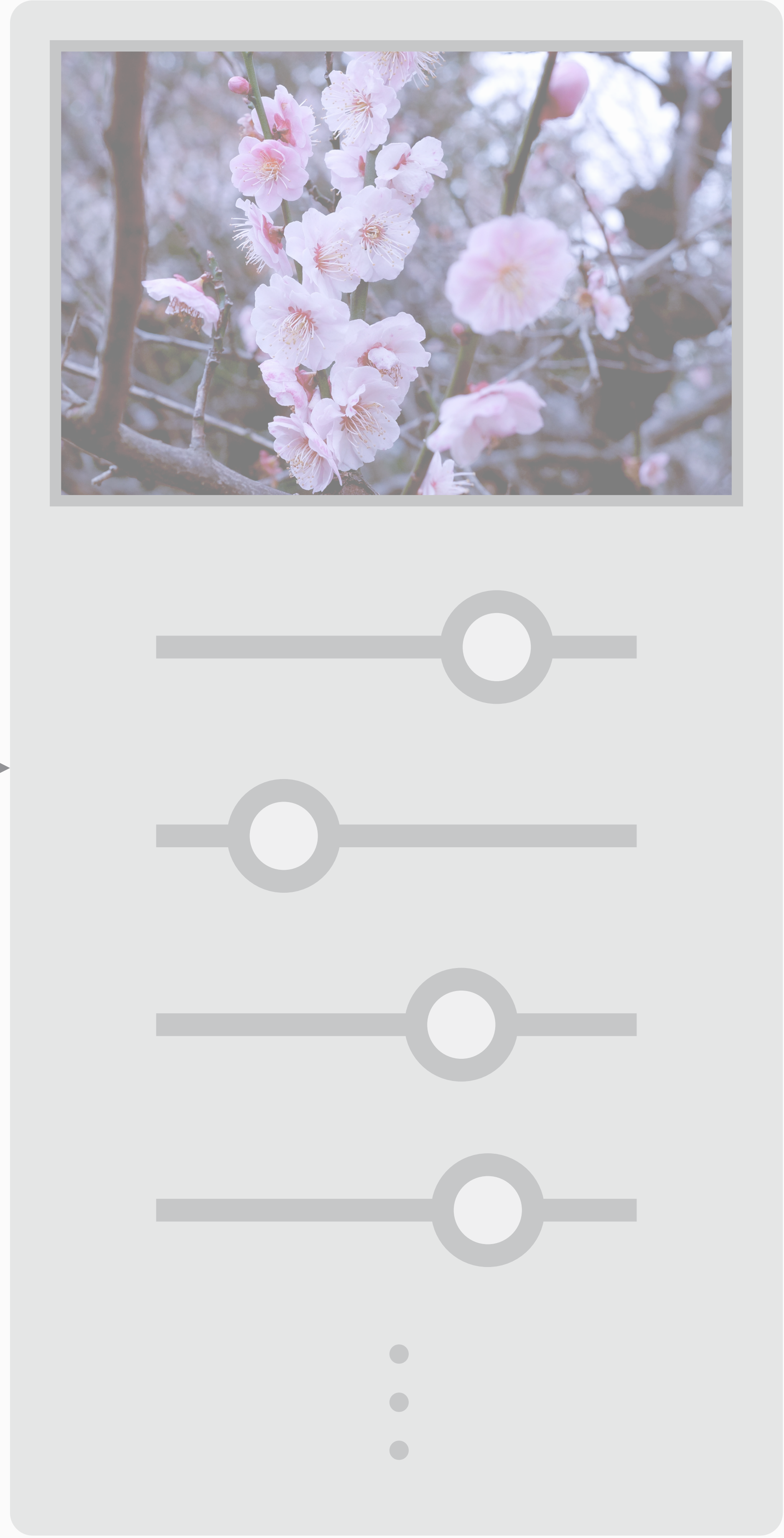


Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface



2D search subtask #1

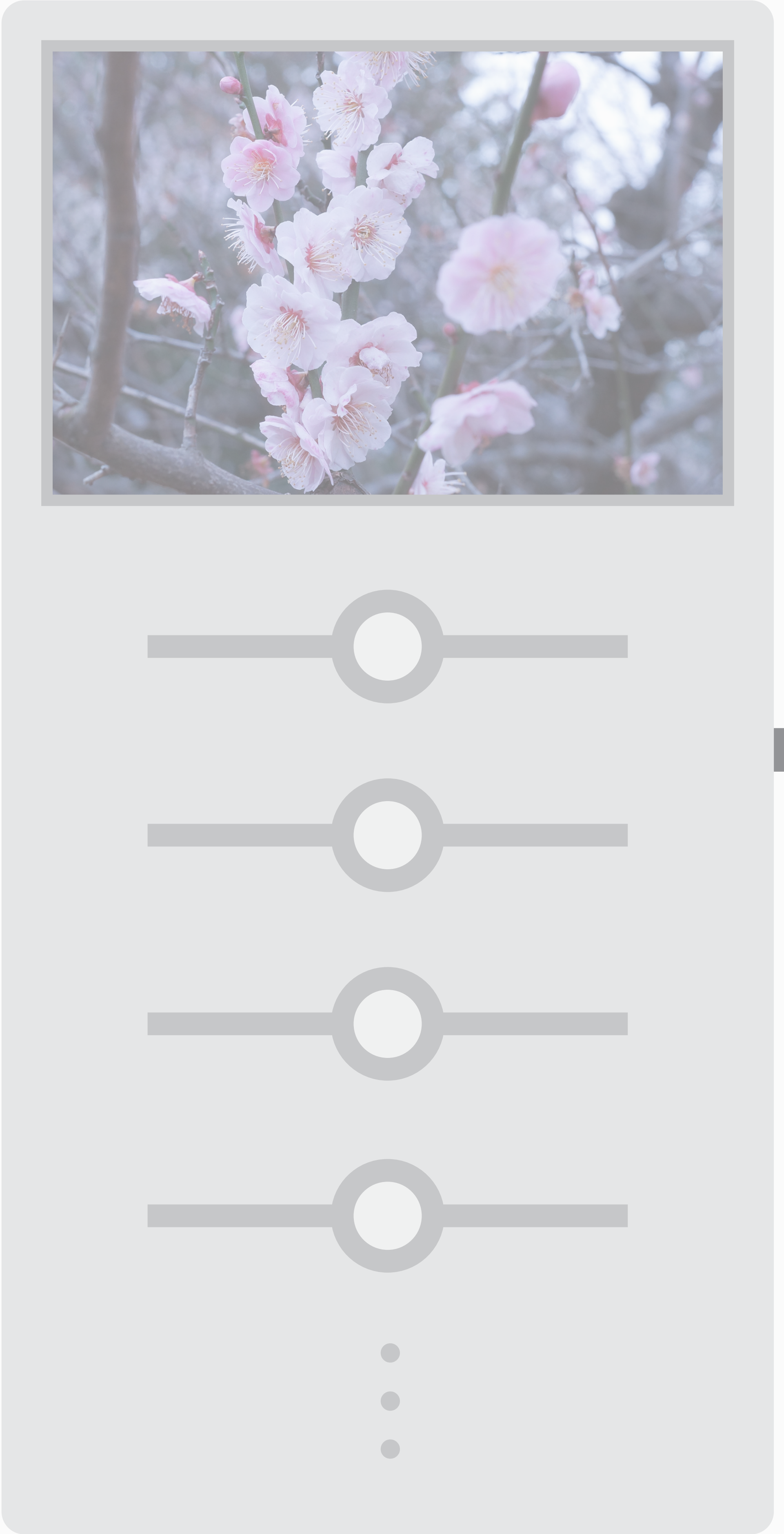
Output:
An optimal parameter set

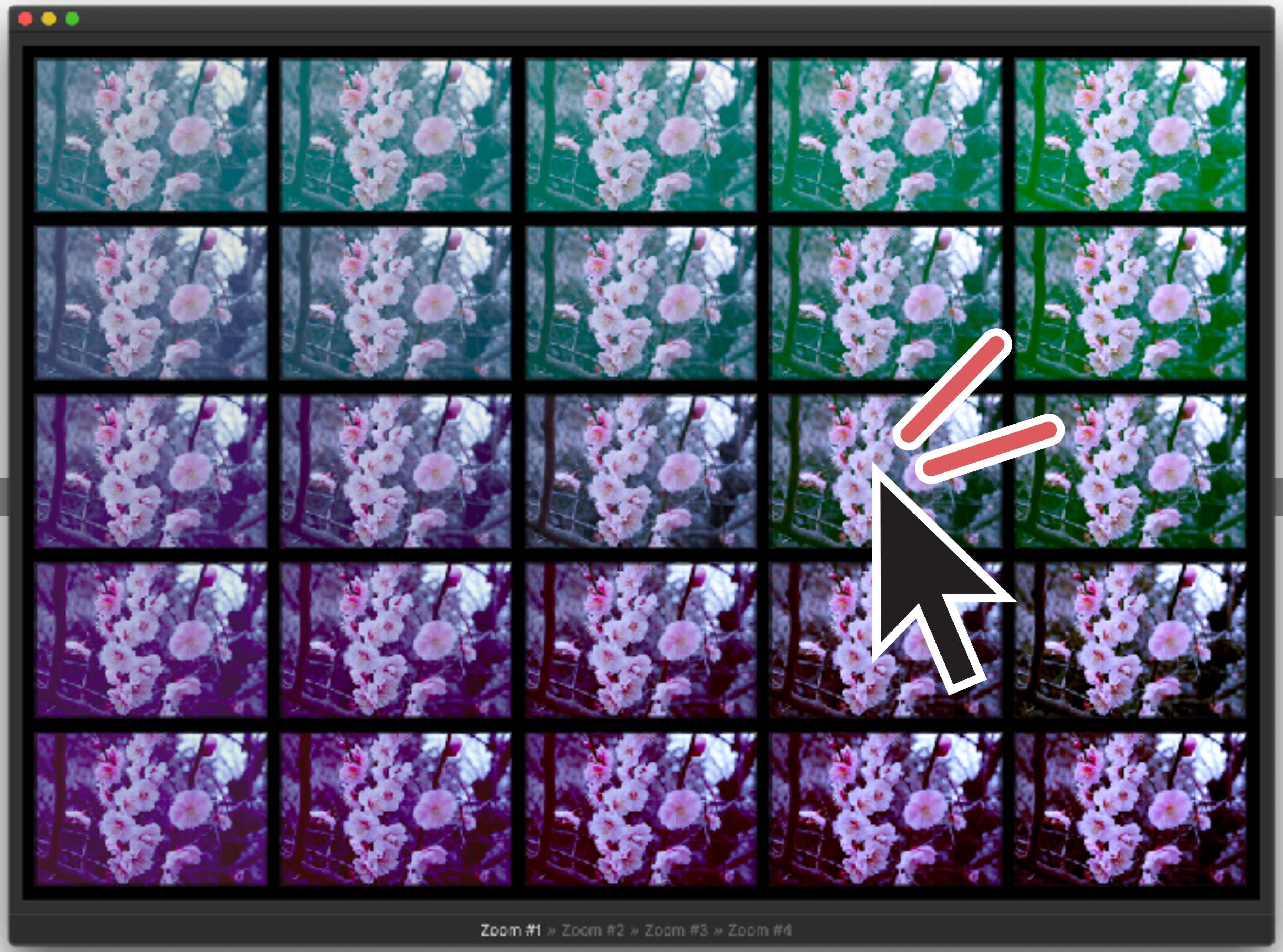


Target:
n design parameters
(e.g., photo enhance)

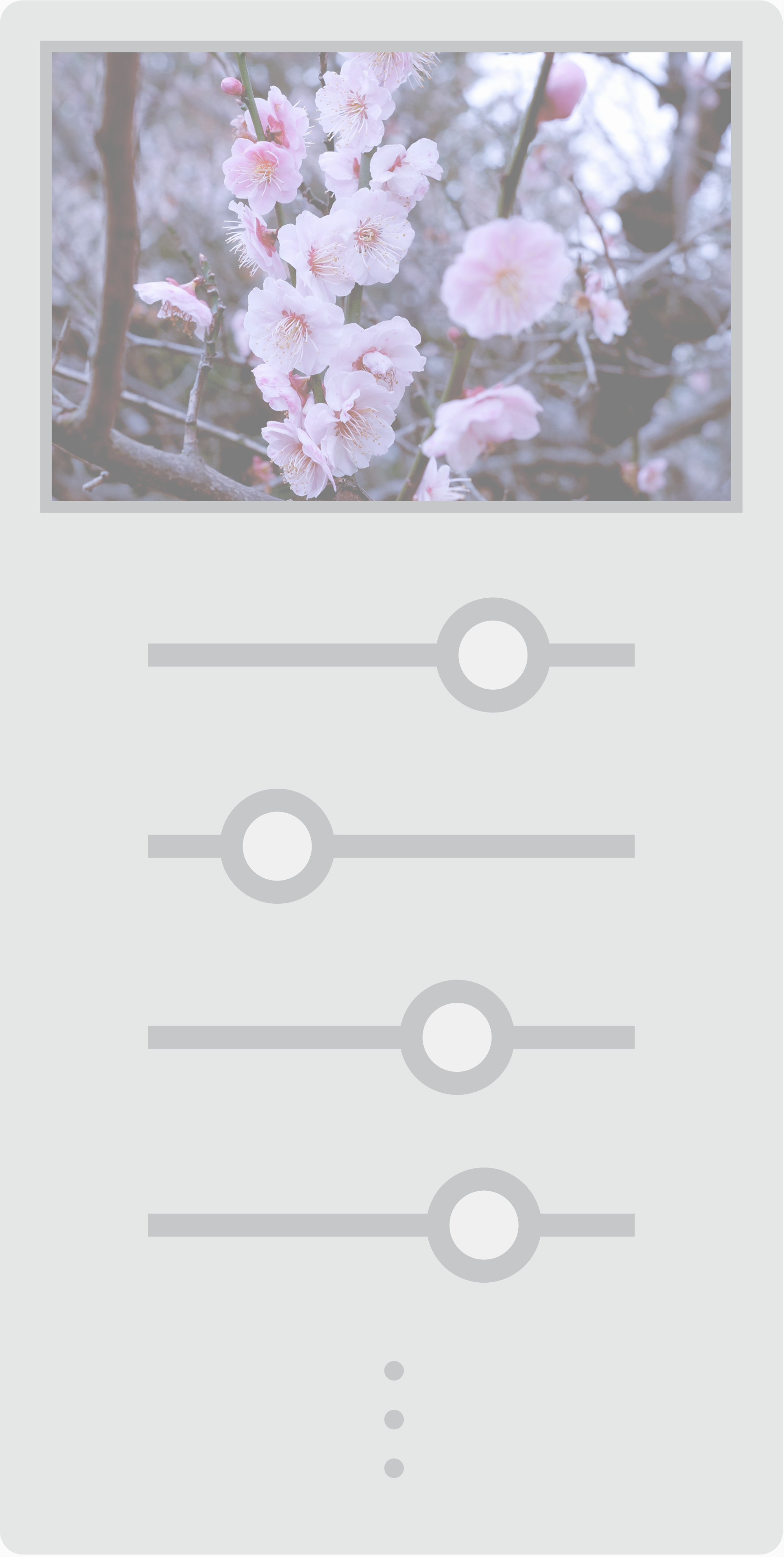
Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface

Output:
An optimal parameter set





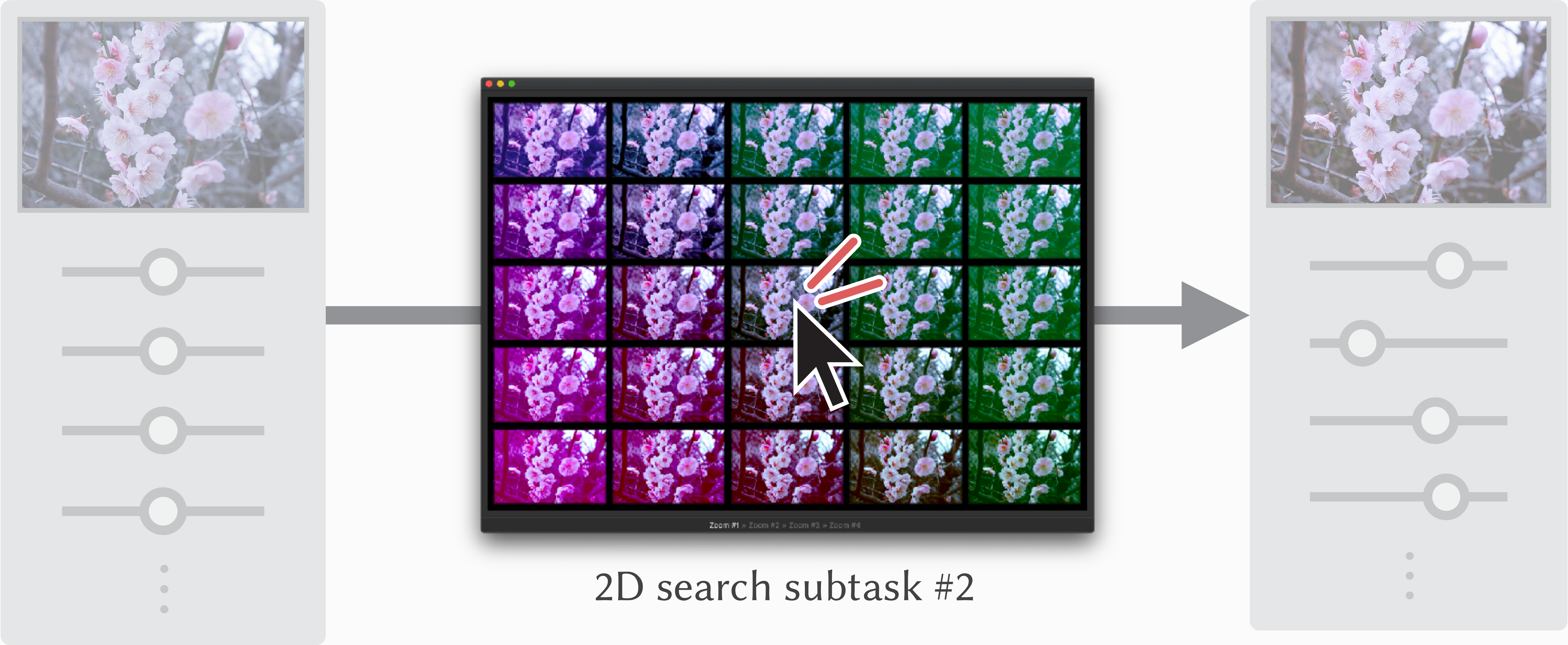
2D search subtask #1



Target:
n design parameters
(e.g., photo enhance)

Sequential Gallery:
An interactive optimization framework
where the user sequentially performs
2D search subtasks via a grid interface

Output:
An optimal
parameter set

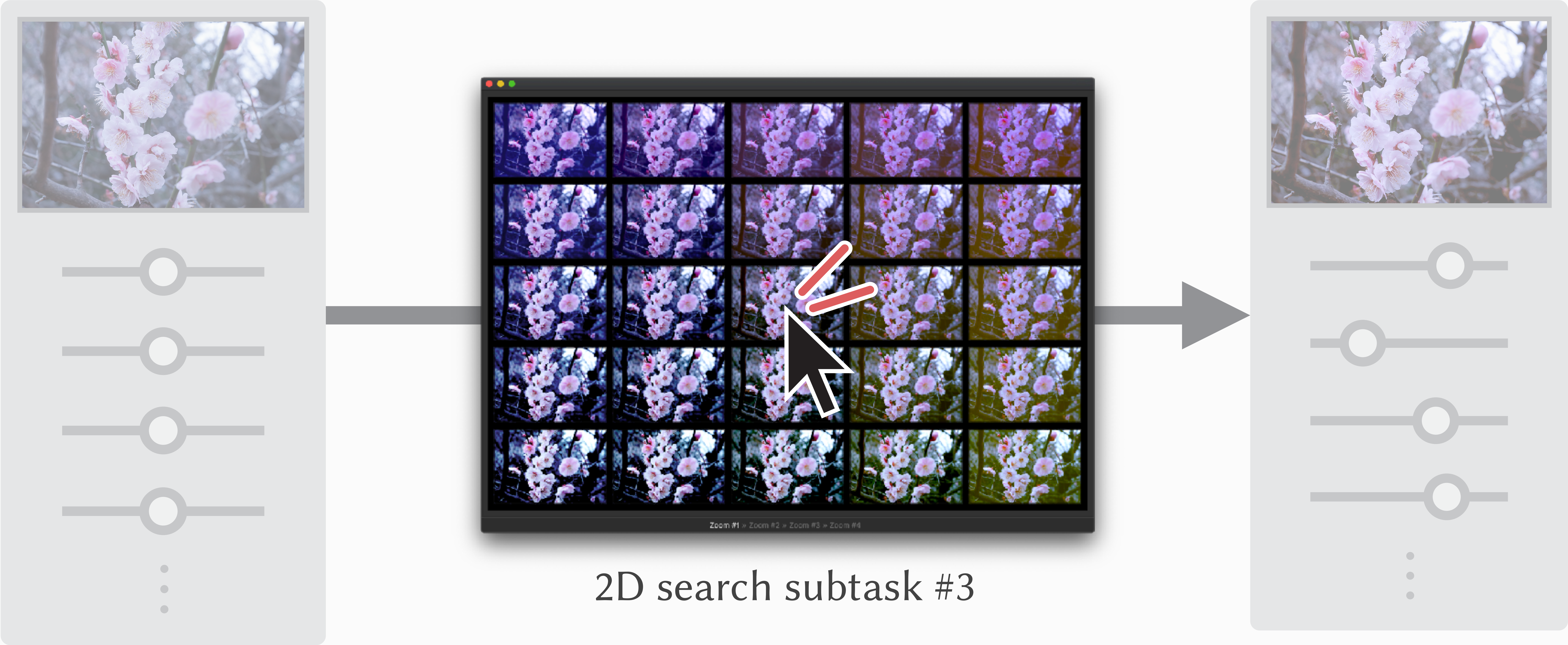


2D search subtask #2

Target:
n design parameters
(e.g., photo enhance)

Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface

Output:
An optimal parameter set

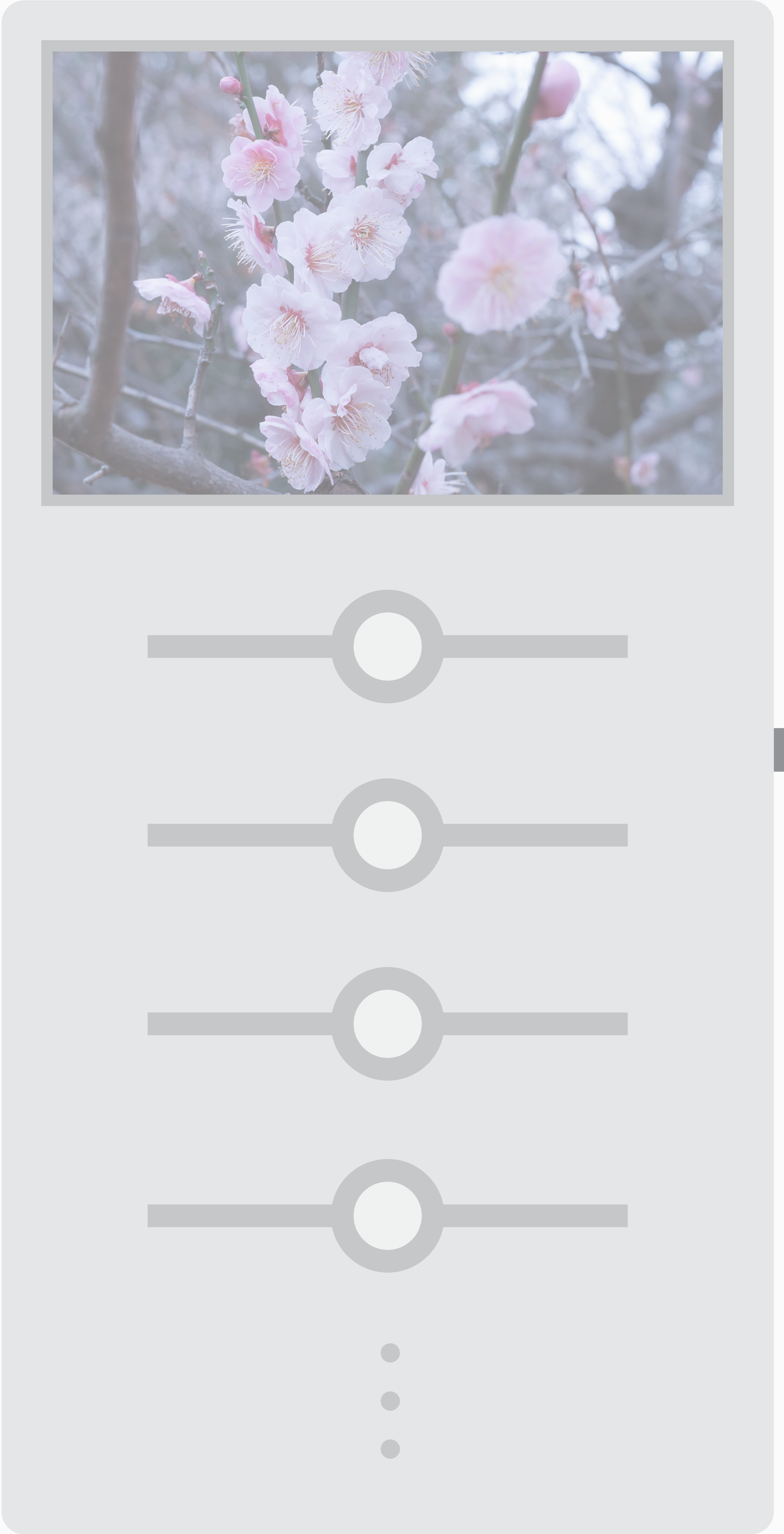


2D search subtask #3

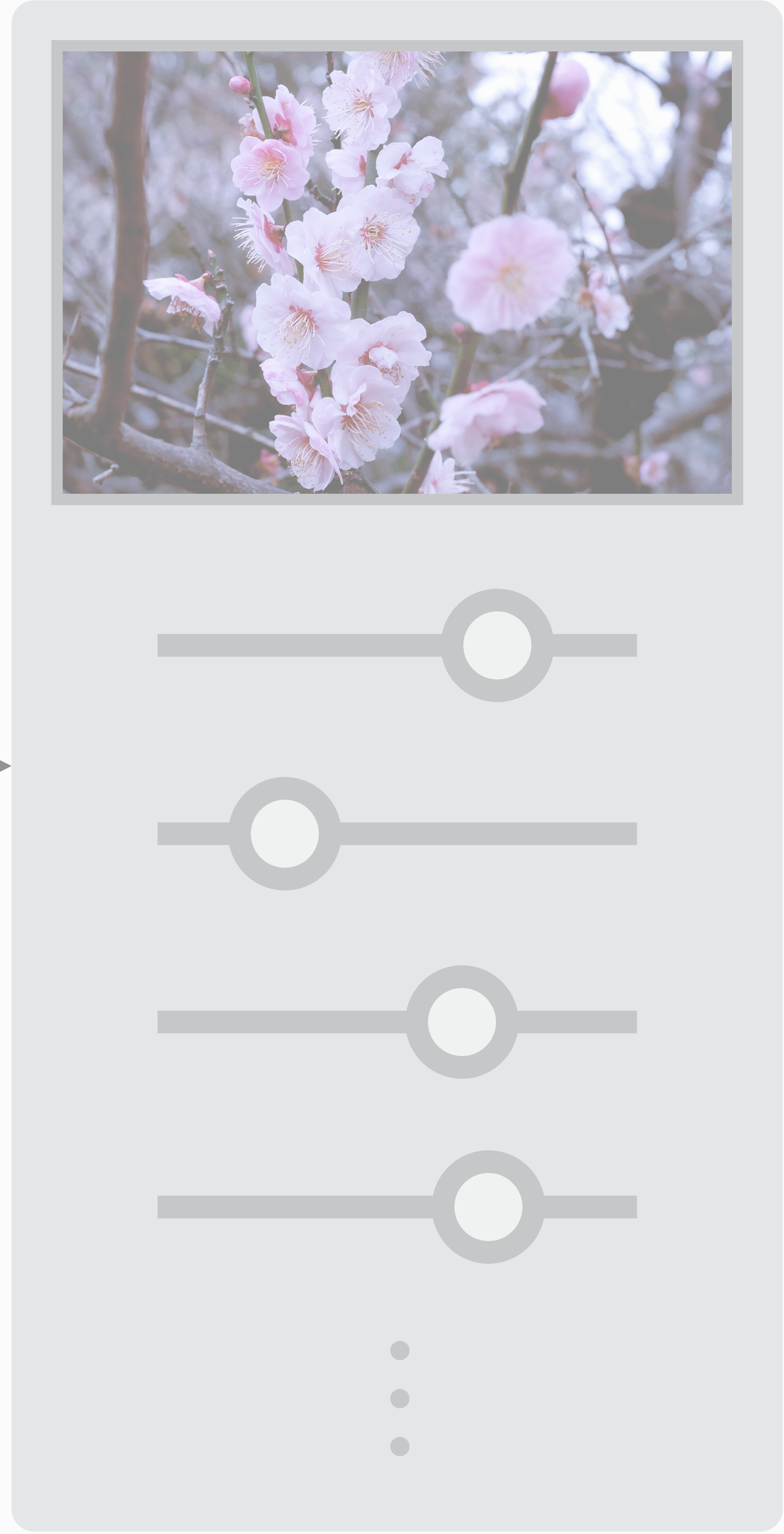
Target:
n design parameters
(e.g., photo enhance)

Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface

Output:
An optimal parameter set



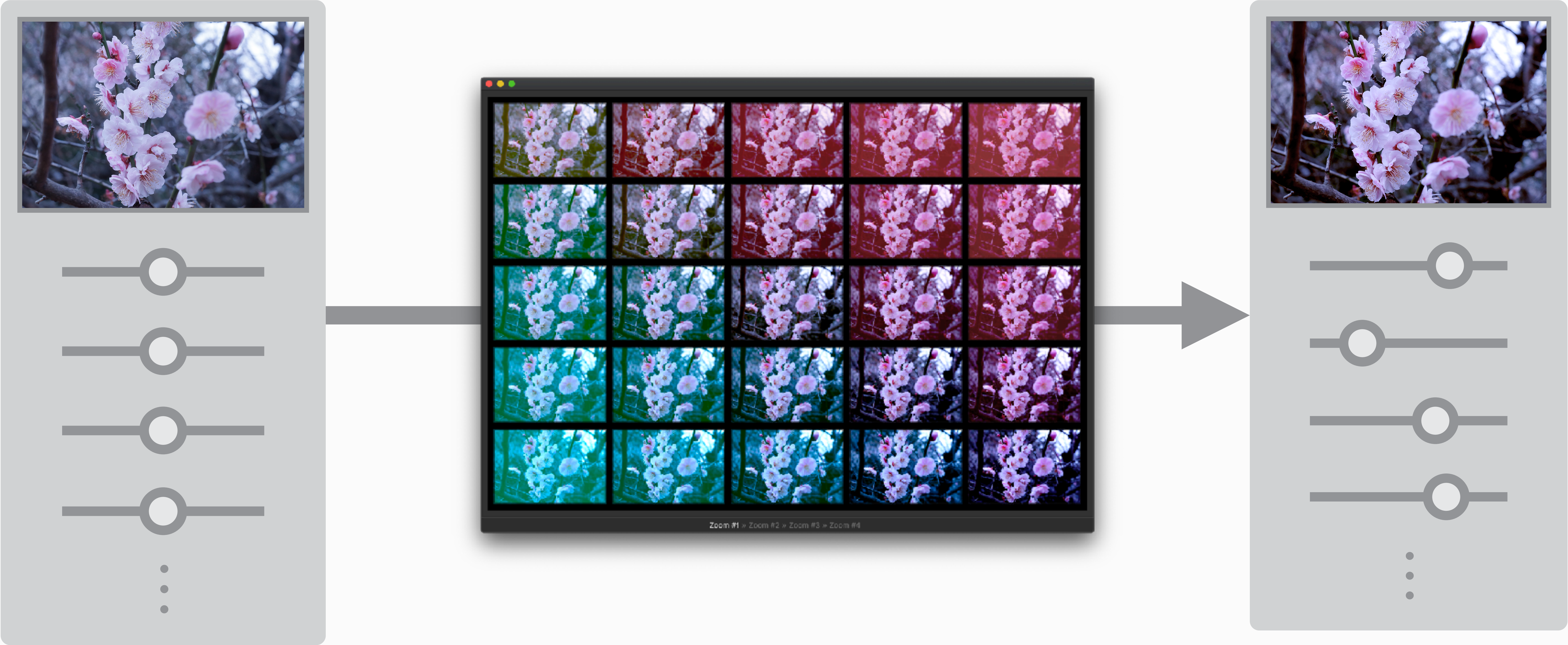
2D search subtask #4



Target:
n design parameters
(e.g., photo enhance)

Sequential Gallery:
An interactive optimization framework where the user sequentially performs 2D search subtasks via a grid interface

Output:
An optimal parameter set

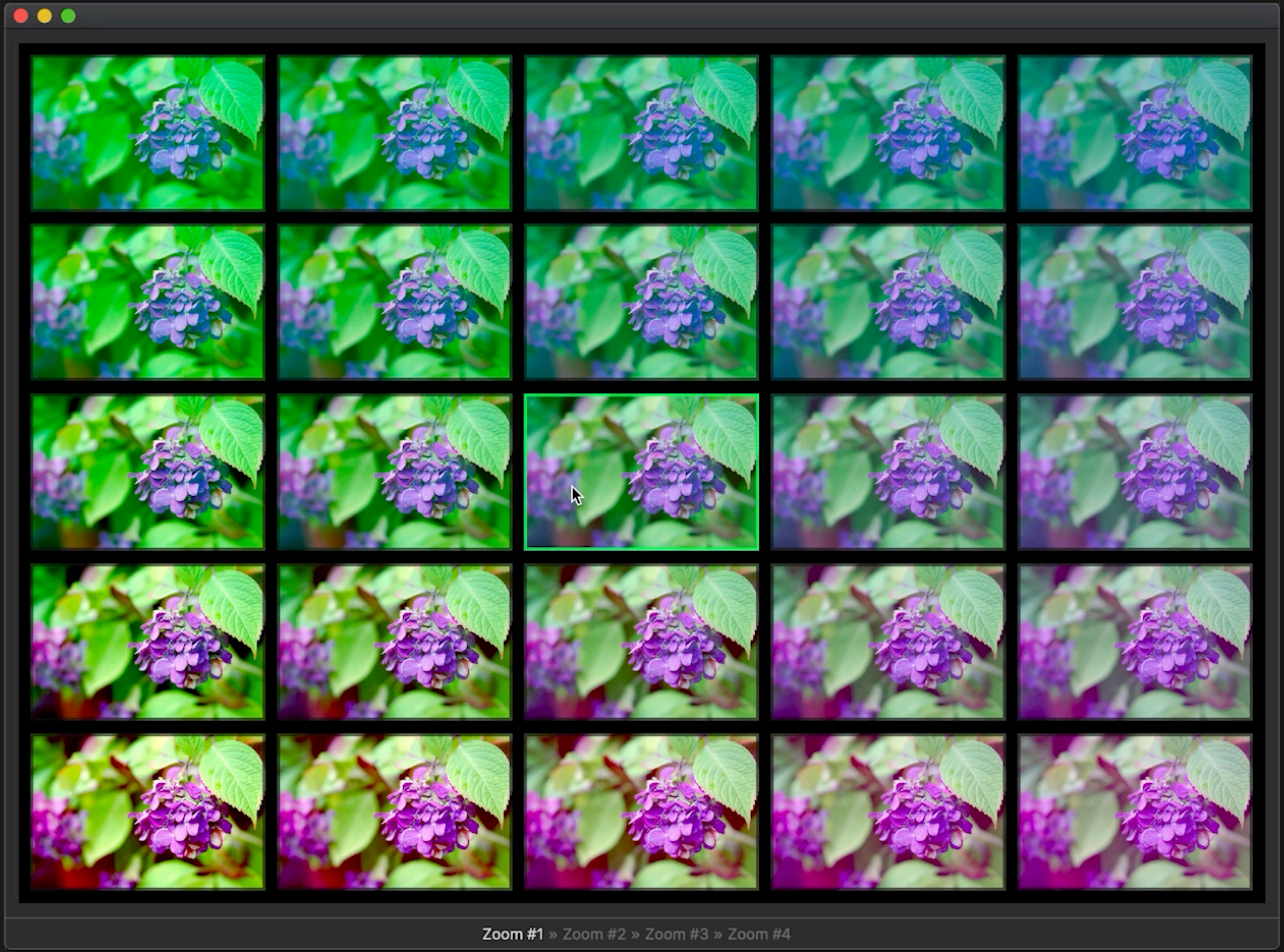


Overview

Example Usage

Photo Color Enhancement (12D)

Brightness, contrast, saturation, shadows (RGB), midtones (RGB), and highlights (RGB)



x1.5 speed

Original photograph

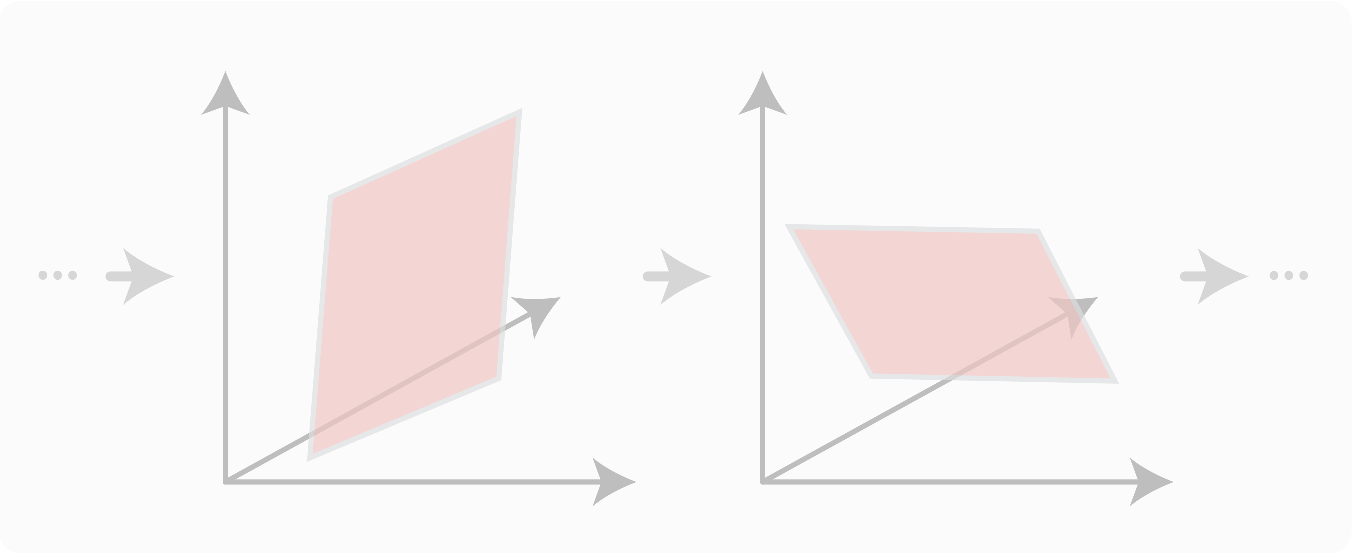
Enhanced photograph
(after 4 iterations)

Overview

Summary of Contributions

Summary of Contributions

- **Novel algorithm: Sequential plane search**
  - A variant of **preferential Bayesian optimization (PBO)**, enabling user-in-the-loop optimization
  - Can find optimal solutions with fewer iterations than the previous algorithm [Koyama+17]
- **Interactive system: Sequential Gallery**
  - Use the sequential-plane-search algorithm in combination with a zoomable grid interface
  - Enable the user to effectively explore the design space and perform the optimization

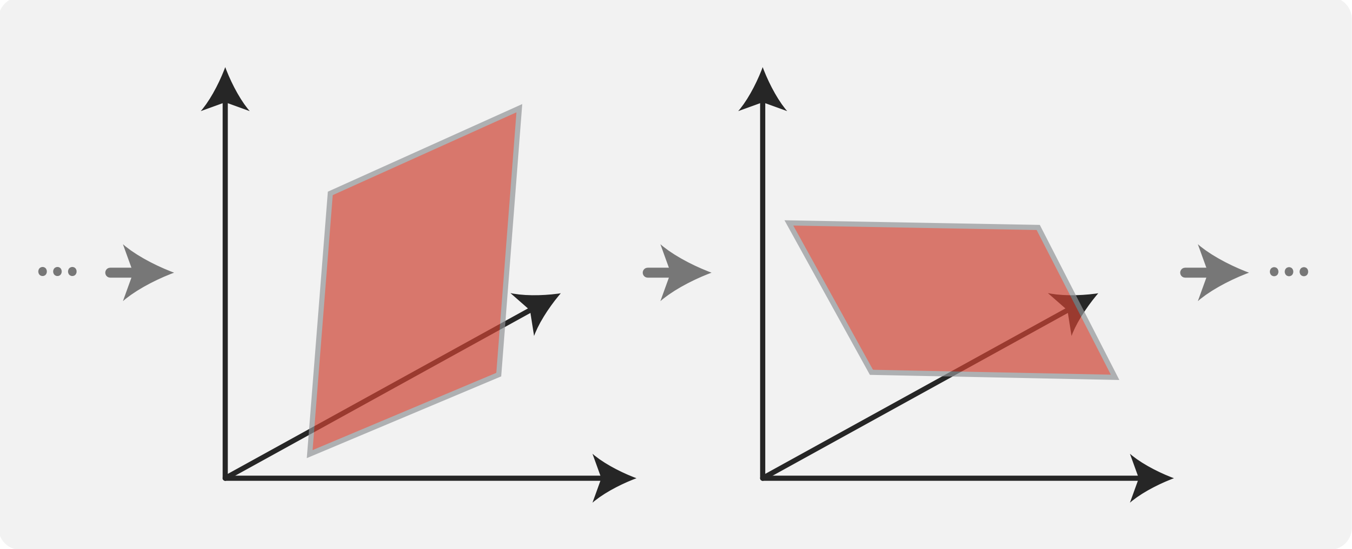


Sequential plane search

Sequential Gallery

Summary of Contributions

- **Novel algorithm: Sequential plane search**
  - A variant of **preferential Bayesian optimization (PBO)**, enabling user-in-the-loop optimization
  - Can find optimal solutions with fewer iterations than the previous algorithm [Koyama+17]
- **Interactive system: Sequential Gallery**
  - Use the sequential-plane-search algorithm in combination with a zoomable grid interface
  - Enable the user to effectively explore the design space and perform the optimization

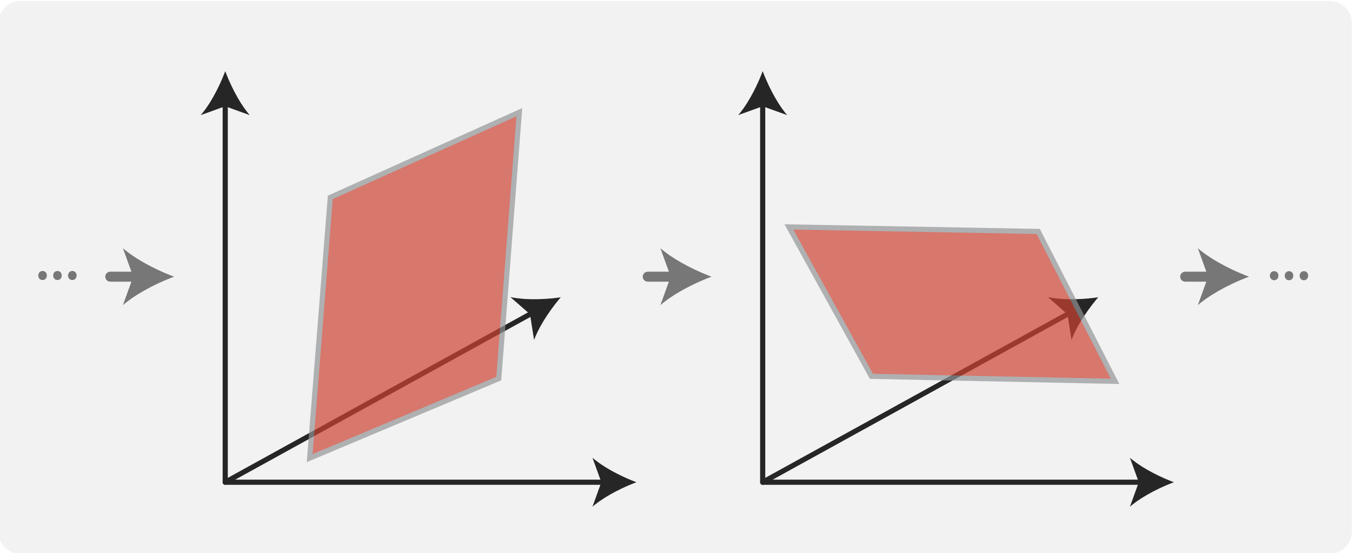


Sequential plane search

Sequential Gallery

Summary of Contributions

- **Novel algorithm: Sequential plane search**
  - A variant of **preferential Bayesian optimization (PBO)**, enabling user-in-the-loop optimization
  - Can find optimal solutions with fewer iterations than the previous algorithm [Koyama+17]
- **Interactive system: Sequential Gallery**
  - Use the sequential-plane-search algorithm in combination with a zoomable grid interface
  - Enable the user to effectively explore the design space and perform the optimization



Sequential plane search

Sequential Gallery

Problem Definition

From Mathematical Viewpoint

Problem Definition from Mathematical Viewpoint

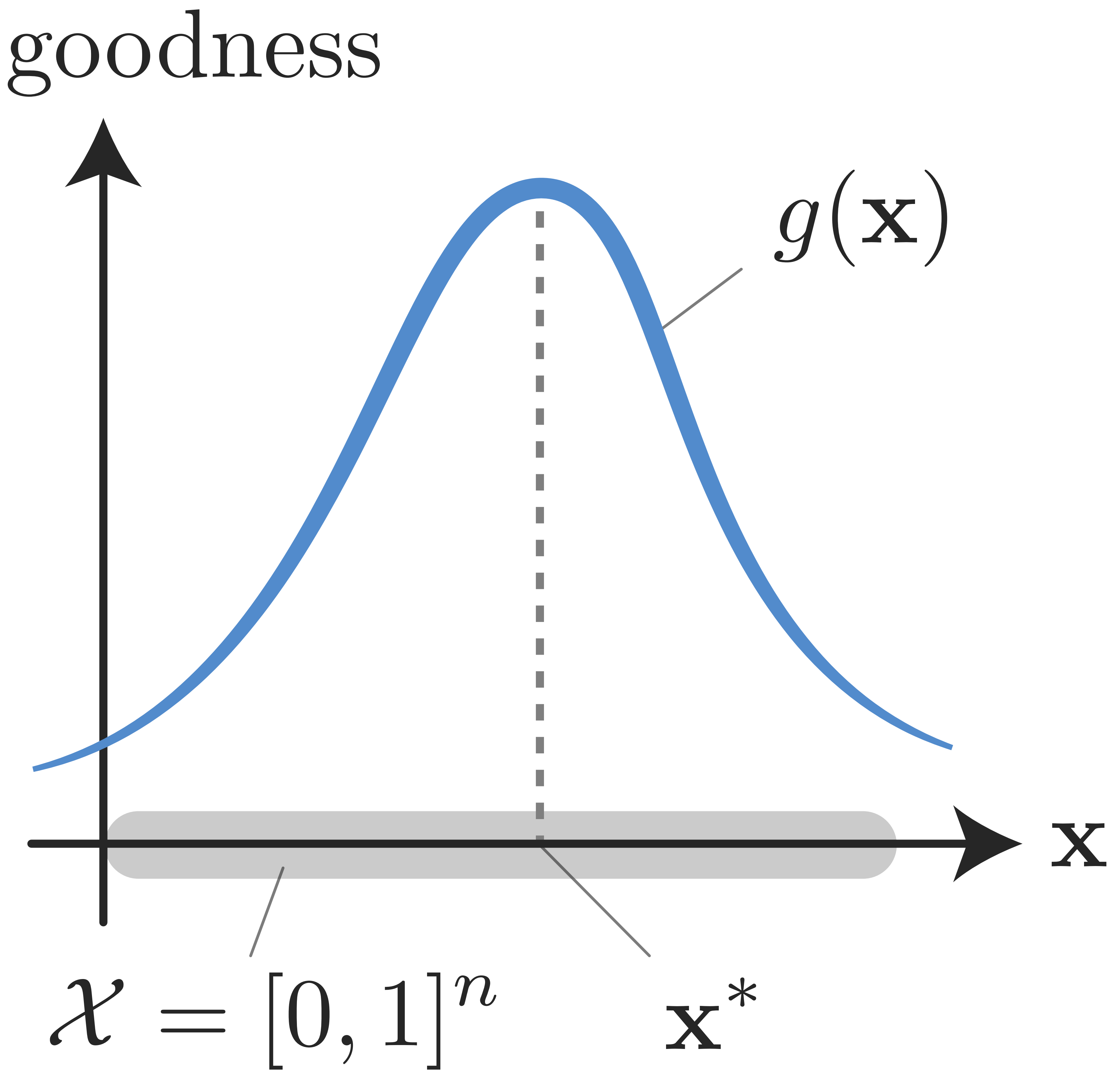
- Suppose that we have n sliders to adjust
- Let $\mathscr{X} = [0,1]^n$ be the search space and $\mathbf{x} \in \mathscr{X}$ be a set of n parameter values

[Koyama+, Computational Interaction (2018)] Yuki Koyama and Takeo Igarashi. 2018. Computational Design with Crowds. In Computational Interaction (Eds. Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes), Oxford University Press, pp.153—184.

Problem Definition from Mathematical Viewpoint

- Suppose that we have n sliders to adjust
- Let $\mathscr{X} = [0,1]^n$ be the search space and $\mathbf{x} \in \mathscr{X}$ be a set of n parameter values
- Let $g : \mathscr{X} \rightarrow \mathbb{R}$ be a perceptual preference function (= **goodness function**) which returns a goodness value
- We want to solve an optimization problem:

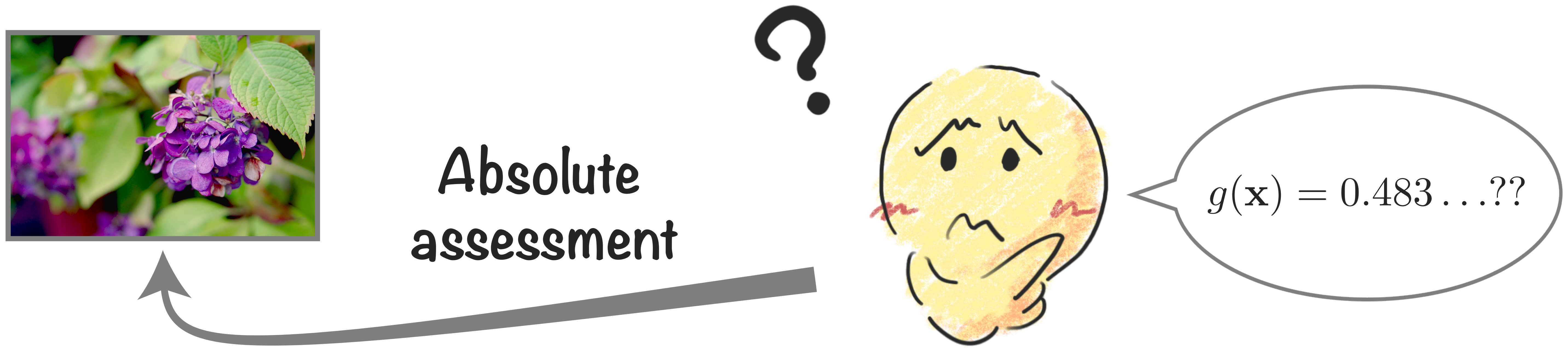
$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathscr{X}}{\operatorname{argmax}}\, g(\mathbf{x})$$



[Koyama+, Computational Interaction (2018)] Yuki Koyama and Takeo Igarashi. 2018. Computational Design with Crowds. In Computational Interaction (Eds. Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes), Oxford University Press, pp.153—184.

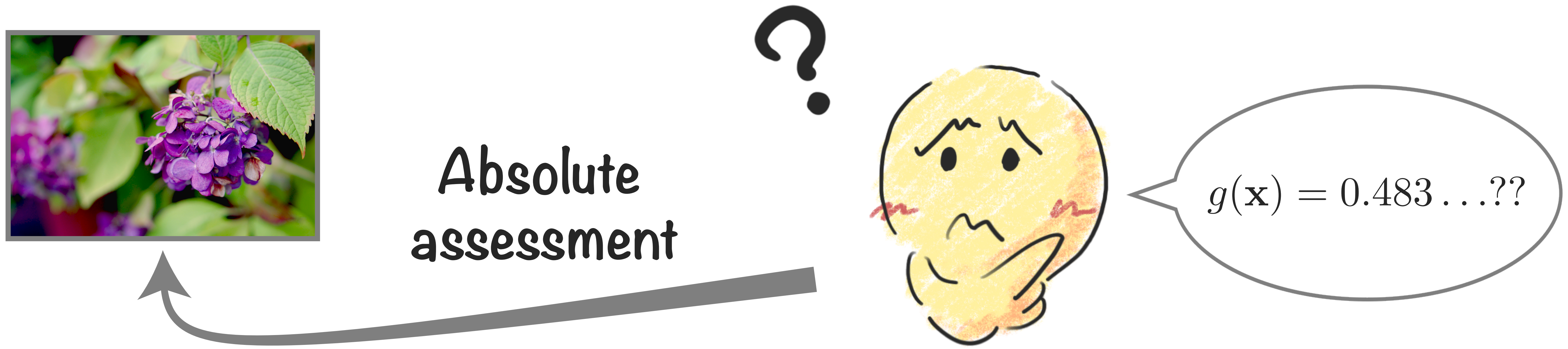
Interacting with Goodness Function

Absolute assessment should not be used:
The user cannot directly answer the function value reliably [Brochu+10; Koyama+18]

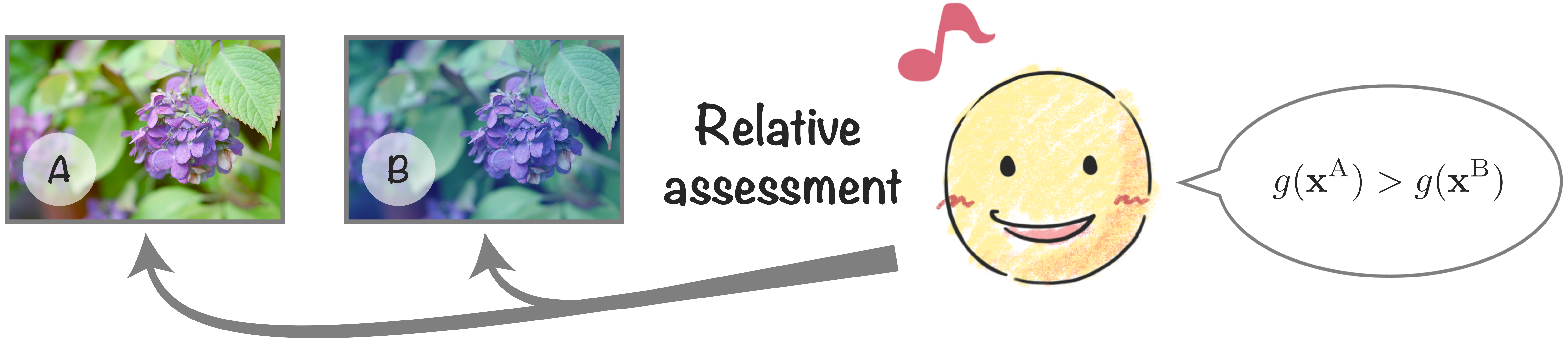


Interacting with Goodness Function

Absolute assessment should not be used:
The user cannot directly answer the function value reliably [Brochu+10; Koyama+18]



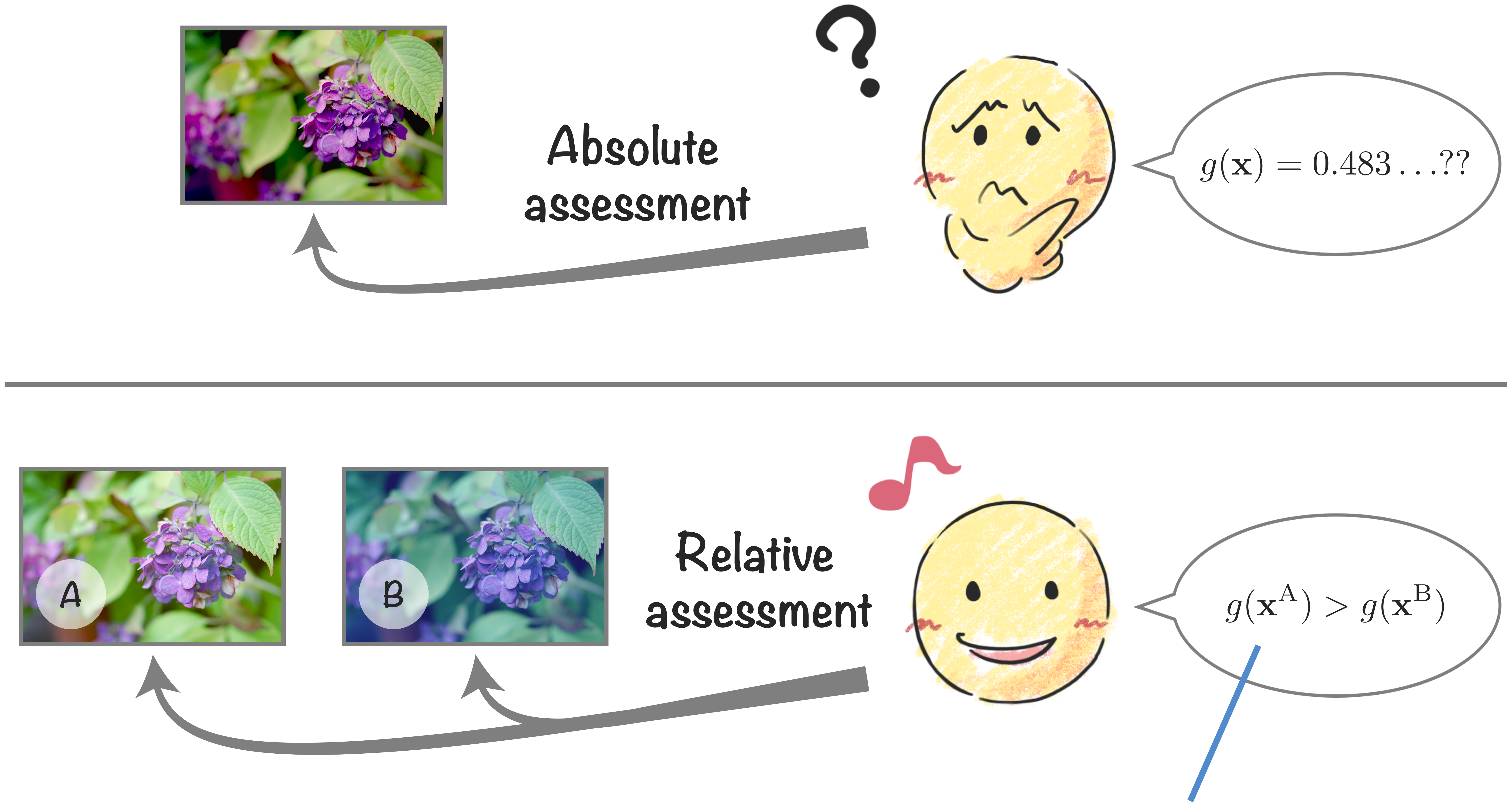
Relative assessment should be used:
The user can answer which option is better among two (or more) options



Interacting with Goodness Function

Absolute assessment should not be used:
The user cannot directly answer the function value reliably [Brochu+10; Koyama+18]

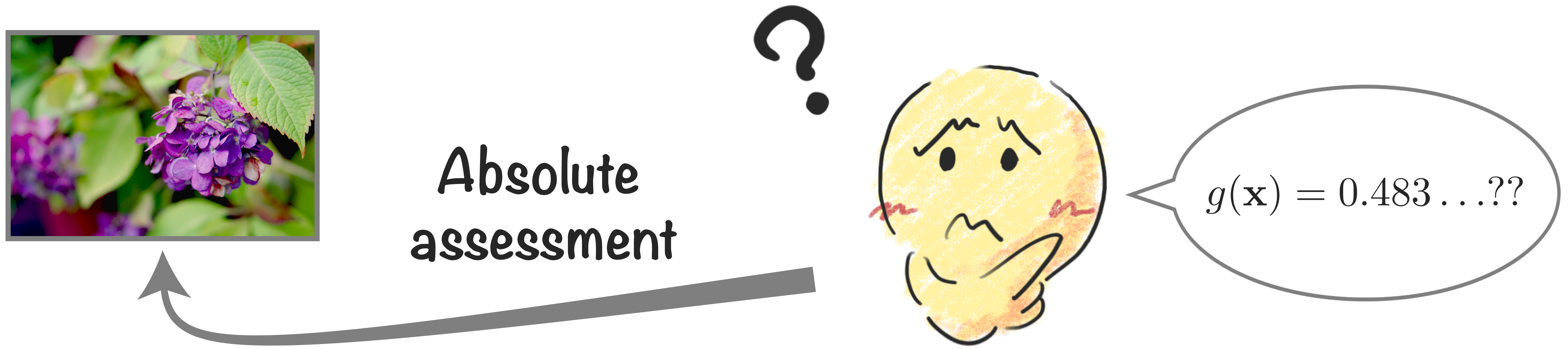
Relative assessment should be used:
The user can answer which option is better among two (or more) options



Preferential feedback

Interacting with Goodness Function

Absolute assessment should not be used:
The user cannot directly answer the function value reliably [Brochu+10; Koyama+18]



Relative assessment should be used:
The user can answer which option is better among two (or more) options

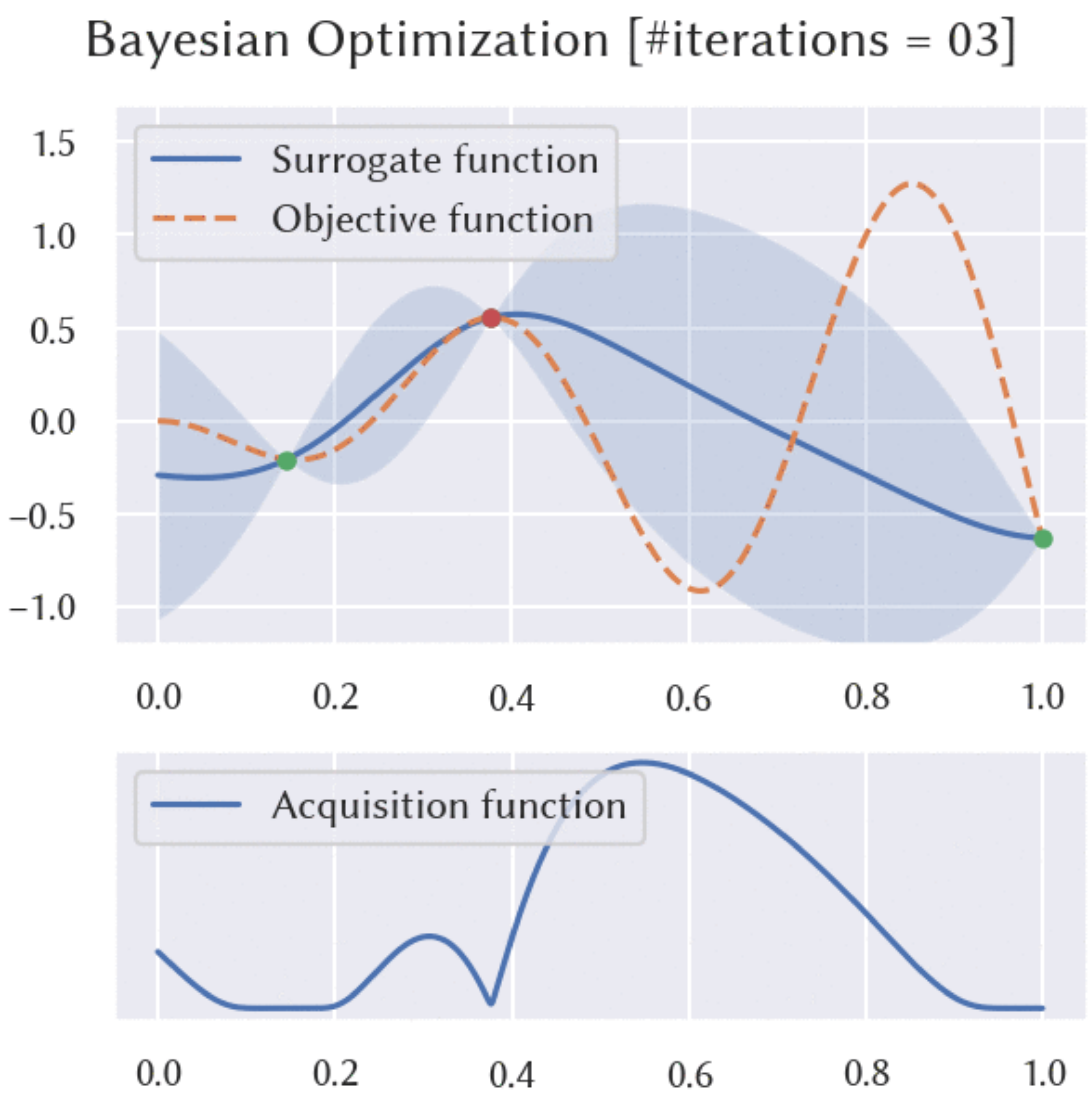
Preferential Bayesian optimization (PBO) can run using relative assessment (i.e., preferential feedback)

Preferential Bayesian Optimization (PBO)

Previous Techniques and Our New Technique

Basics: “Standard” Bayesian Optimization (BO)

- Is a global “black-box” optimization algorithm
- Can find optimal solutions with **only a small number of function evaluations**
- Thus, useful to handle expensive-to-evaluate objective functions
- Example applications: hyperparameter tuning for machine learning models [Akiba+, KDD 2019]



See [Shahriari+, Proc. IEEE 2016] for details

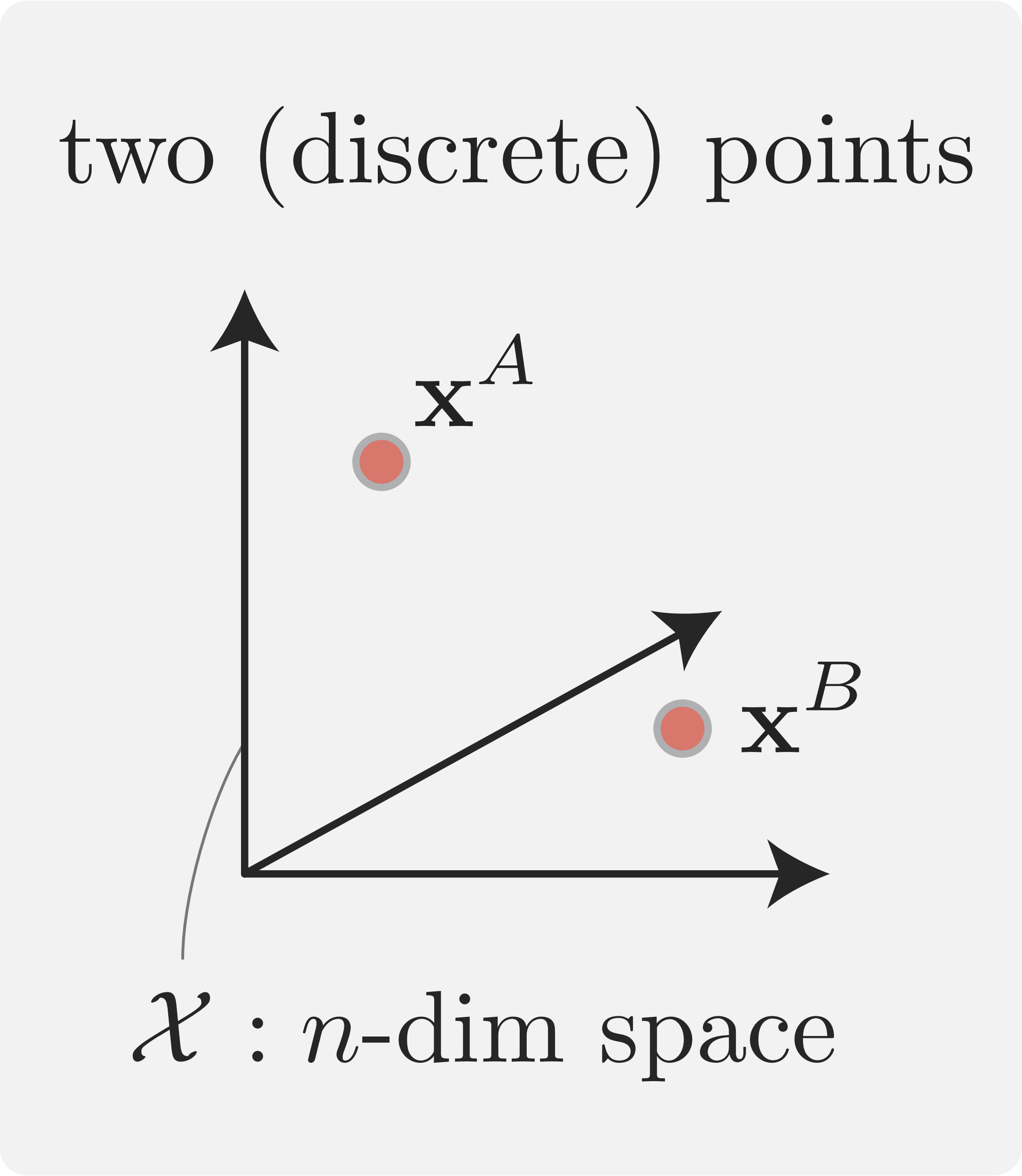
Preferential BO (PBO)

- PBO is an extension of BO, which **runs with relative assessment (or preferential feedback)**, rather than absolute assessment of function values
- PBO can find optimal solutions with **only a small number of preferential feedbacks**
    - Note: human is expensive-to-query

Yuki Koyama, Issei Sato, and Masataka Goto. Sequential Gallery for Interactive Visual Design Optimization. ACM Trans. Graph. (SIGGRAPH 2020)

[Brochu+, NIPS 2007]
"Pairwise Comparison" Query

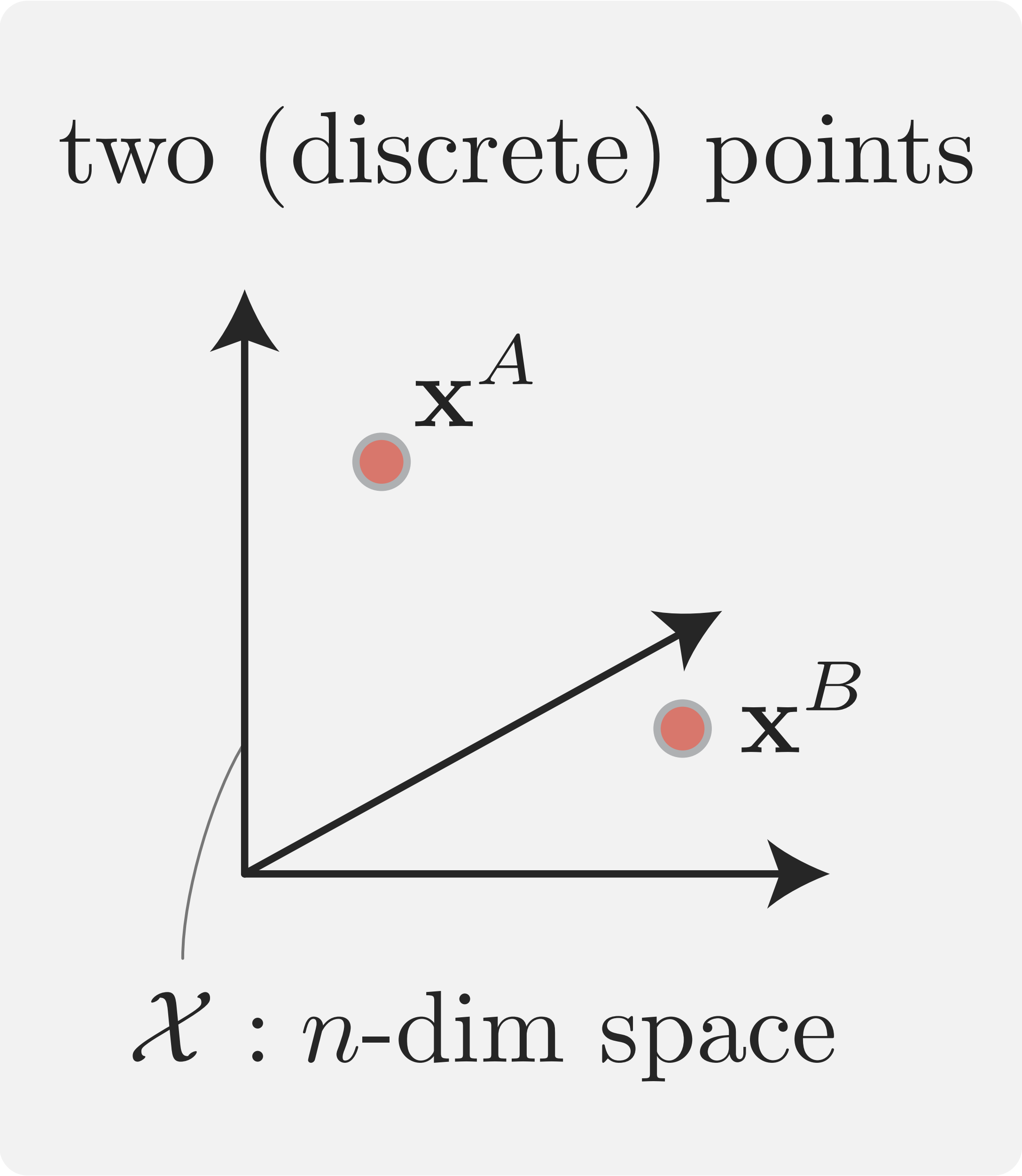
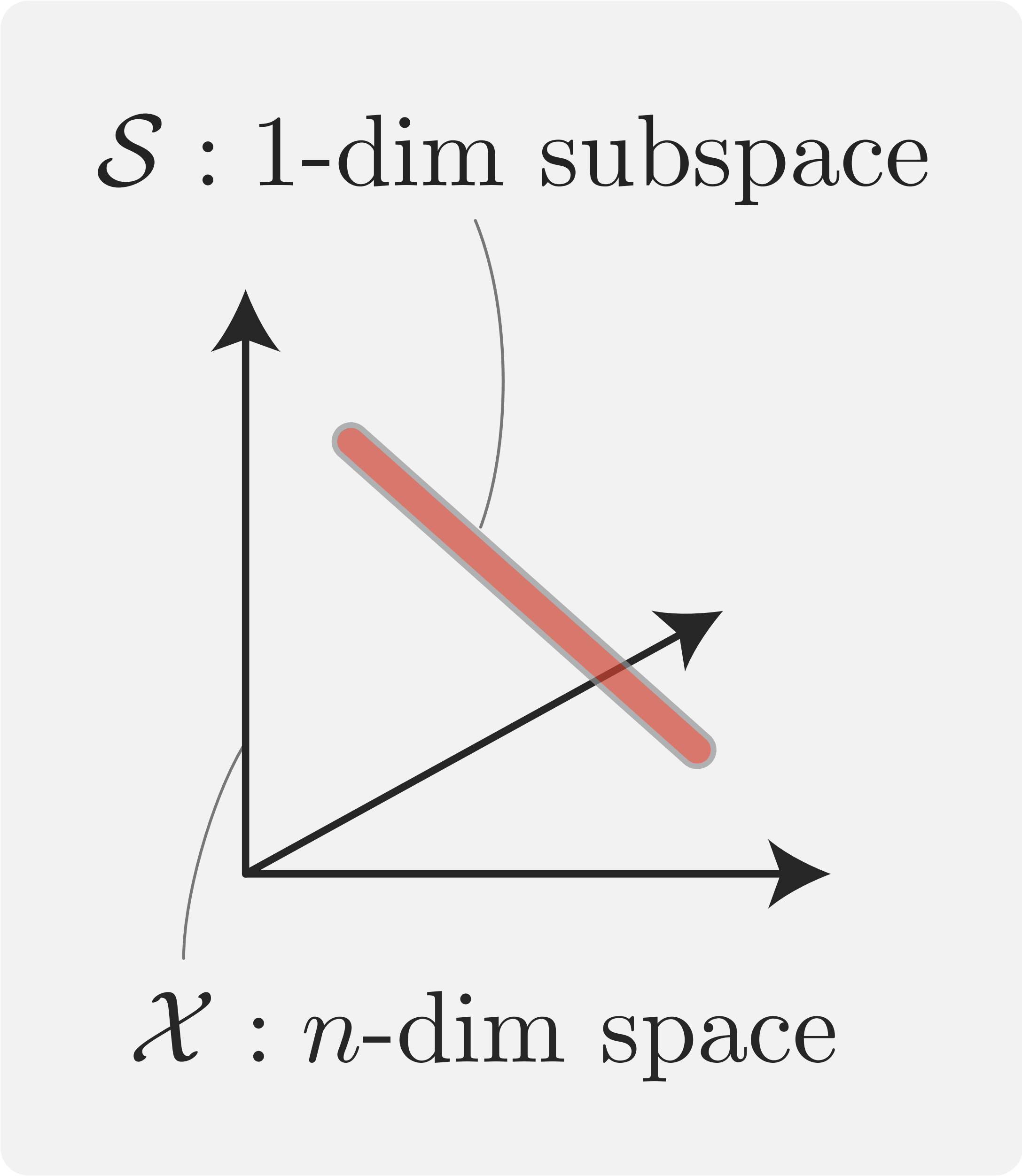
Query type:

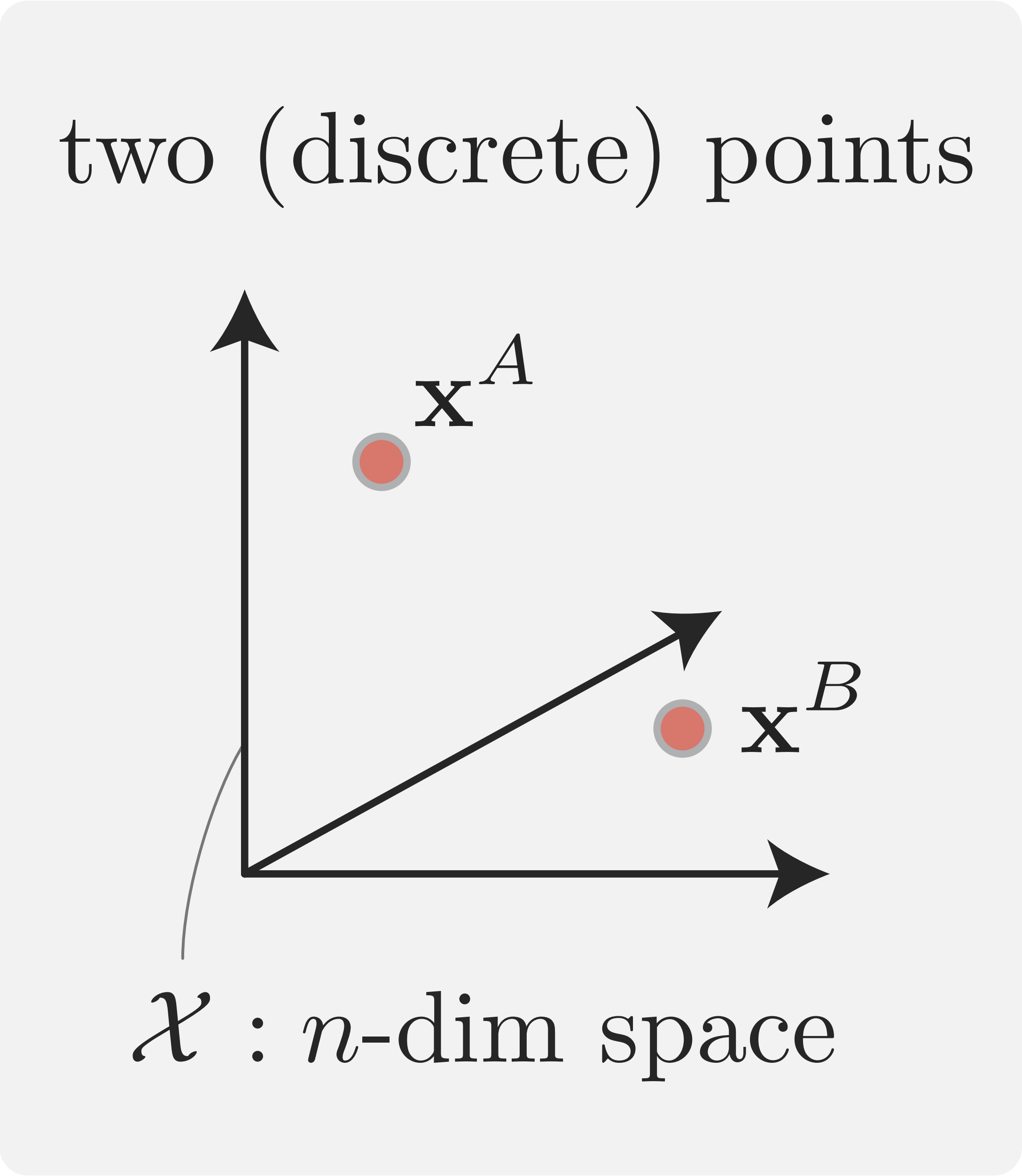
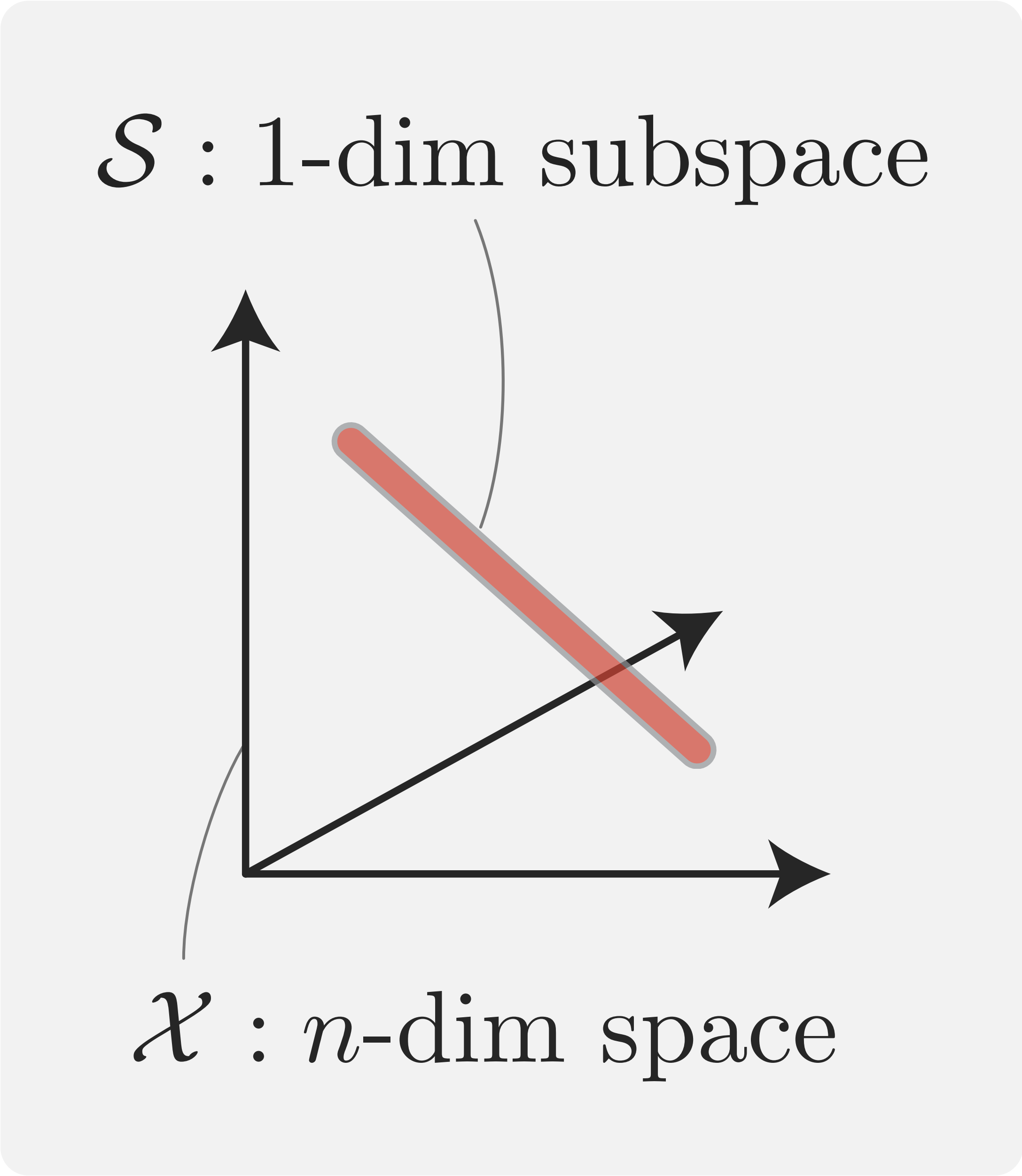
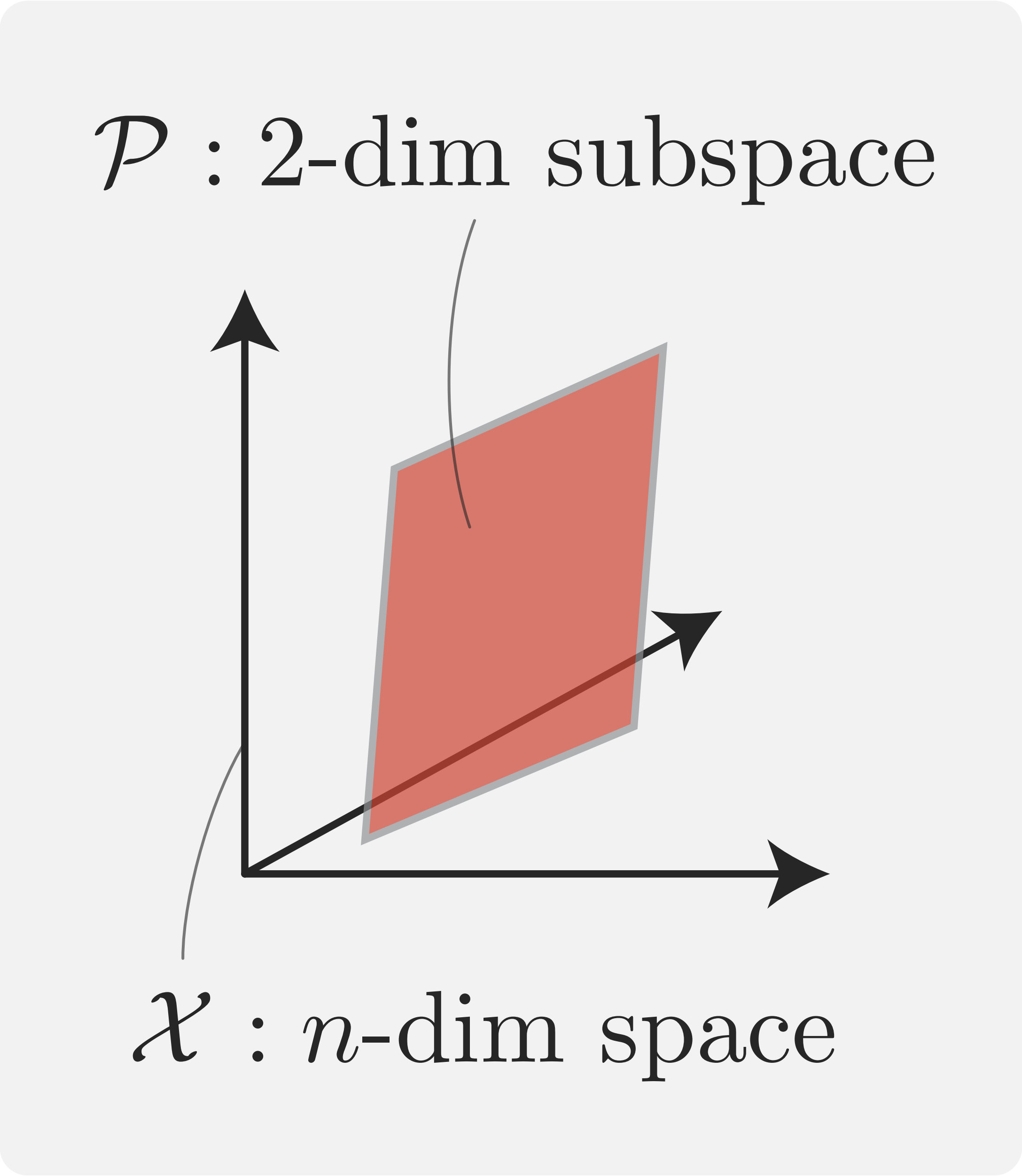


User task:

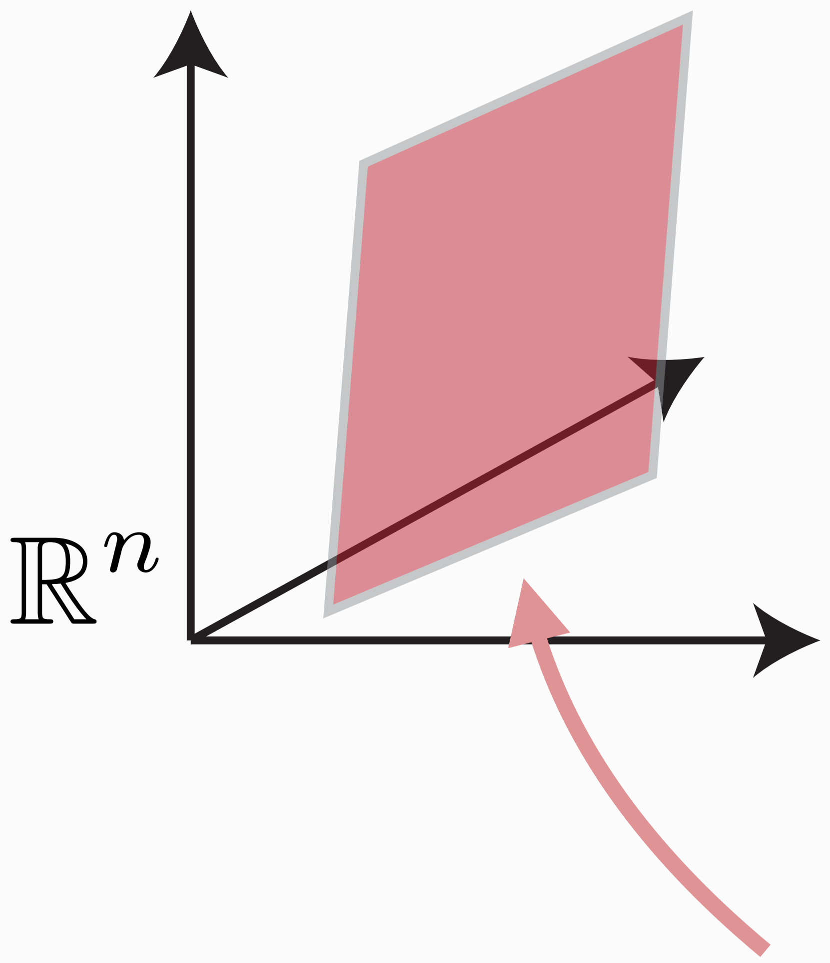
$$\mathbf{x}^{\text{chosen}} = \underset{\mathbf{x} \in \{\mathbf{x}^A, \mathbf{x}^B\}}{\operatorname{argmax}} \; g(\mathbf{x})$$

Notes: Is the first PBO algorithm

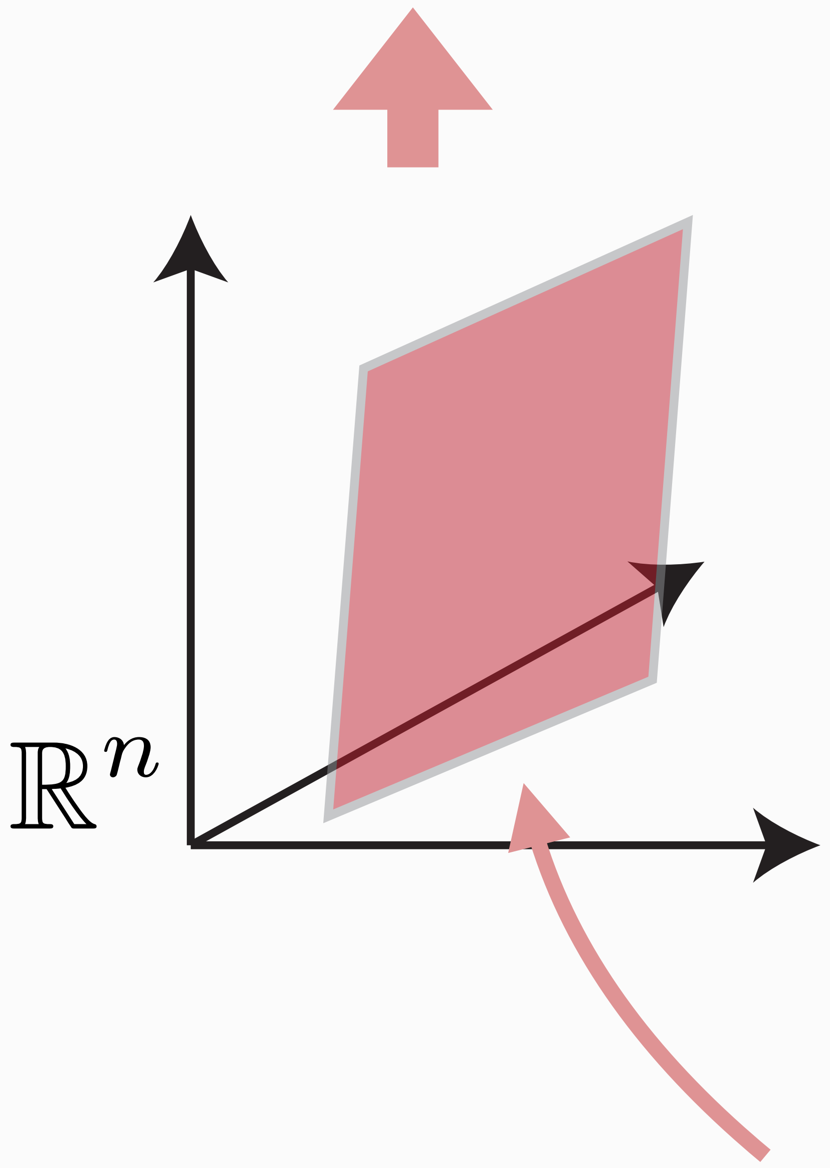
<table>
<tr><th></th><th>[Brochu+, NIPS 2007]
“Pairwise Comparison” Query</th><th>[Koyama+, SIGGRAPH 2017]
“Line Search” Query
(Sequential Line Search)</th></tr>
<tr><td>Query type:</td><td>
</td><td>
</td></tr>
<tr><td>User task:</td><td>$\mathbf{x}^{\text{chosen}} = \underset{\mathbf{x} \in \{\mathbf{x}^A, \mathbf{x}^B\}}{\operatorname{argmax}} \; g(\mathbf{x})$</td><td>$\mathbf{x}^{\text{chosen}} = \underset{\mathbf{x} \in \mathcal{S}}{\operatorname{argmax}} \, g(\mathbf{x})$</td></tr>
<tr><td>Notes:</td><td>Is the first PBO algorithm</td><td>Needs fewer iterations than [Brochu+ NIPS 2007]</td></tr>
</table>

	[Brochu+, NIPS 2007] "Pairwise Comparison" Query	[Koyama+, SIGGRAPH 2017] "Line Search" Query (Sequential Line Search)	[Ours] "Plane Search" Query (Sequential Plane Search)
Query type:			
User task:	$\mathbf{x}^{\text{chosen}} = \operatorname*{argmax}_{\mathbf{x}\in\{\mathbf{x}^A,\mathbf{x}^B\}} g(\mathbf{x})$	$\mathbf{x}^{\text{chosen}} = \operatorname*{argmax}_{\mathbf{x}\in\mathcal{S}} g(\mathbf{x})$	$\mathbf{x}^{\text{chosen}} = \operatorname*{argmax}_{\mathbf{x}\in\mathscr{P}} g(\mathbf{x})$
Notes:	Is the first PBO algorithm	Needs **fewer iterations** than [Brochu+ NIPS 2007]	Needs **even fewer iterations**, and has a good compatibility with grid interfaces

Sequential Gallery Workflow

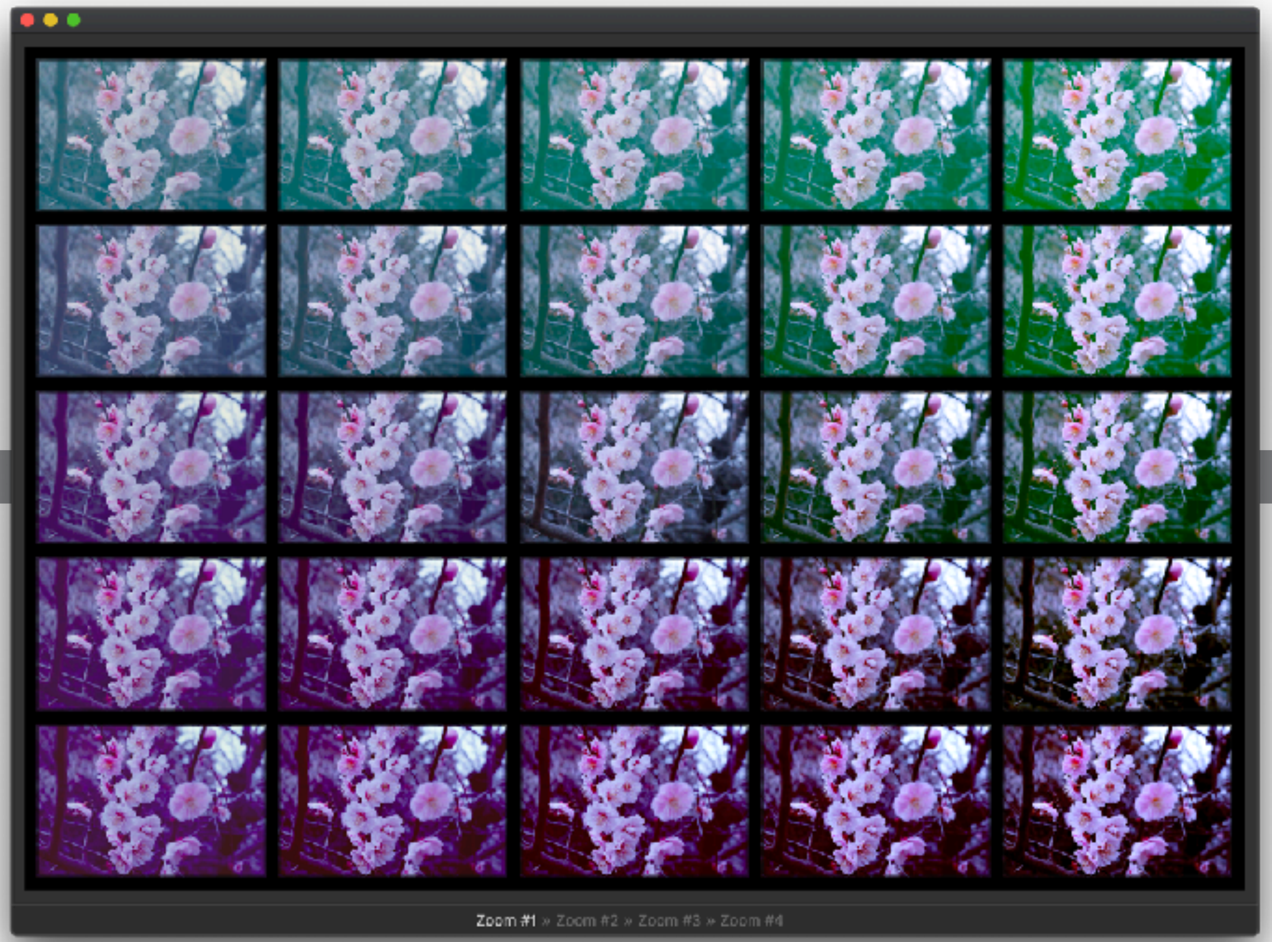


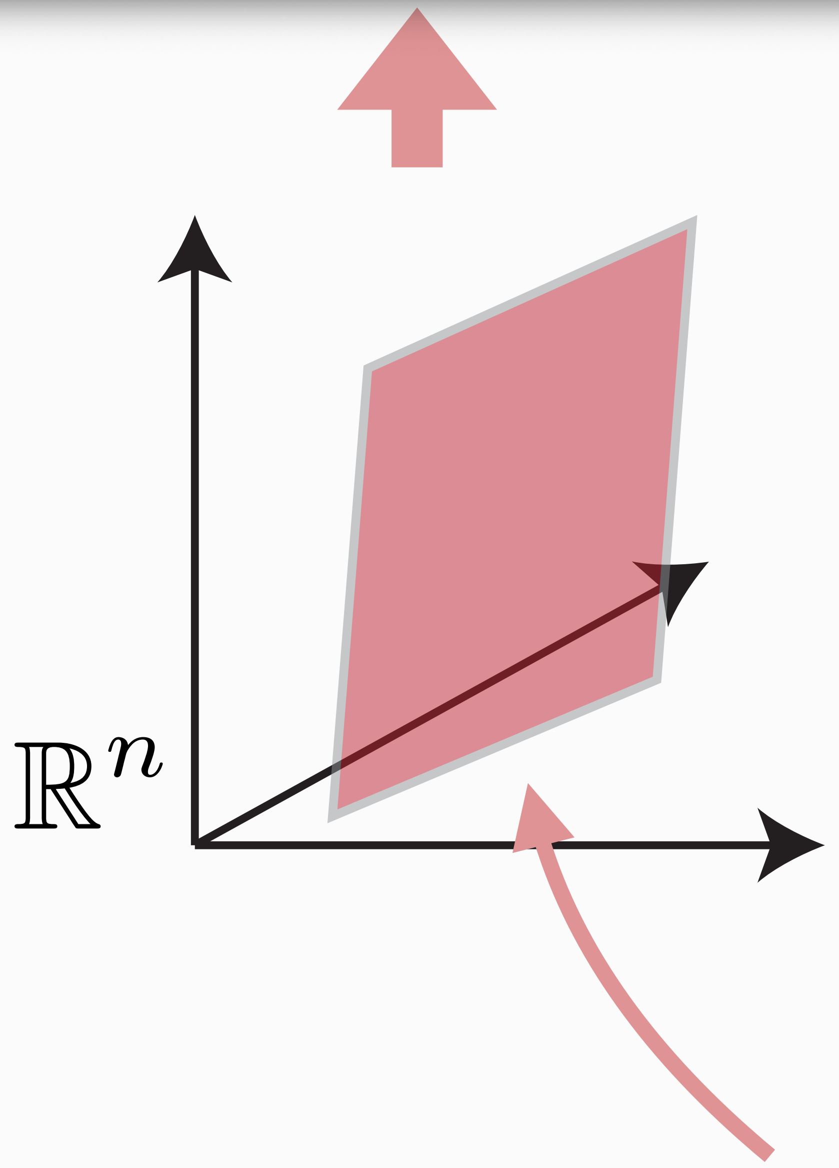
2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm



2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

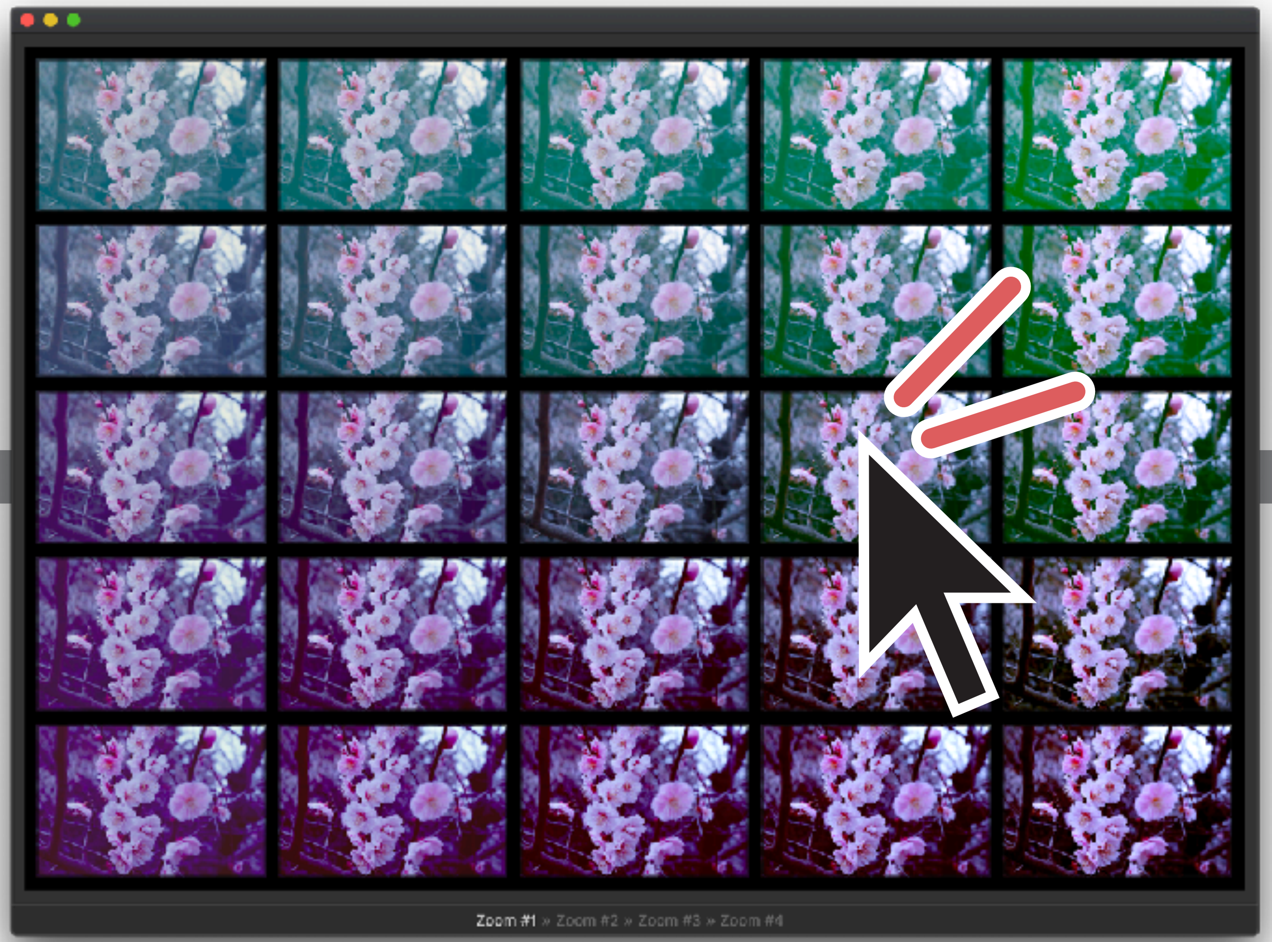
… 2D search subtask

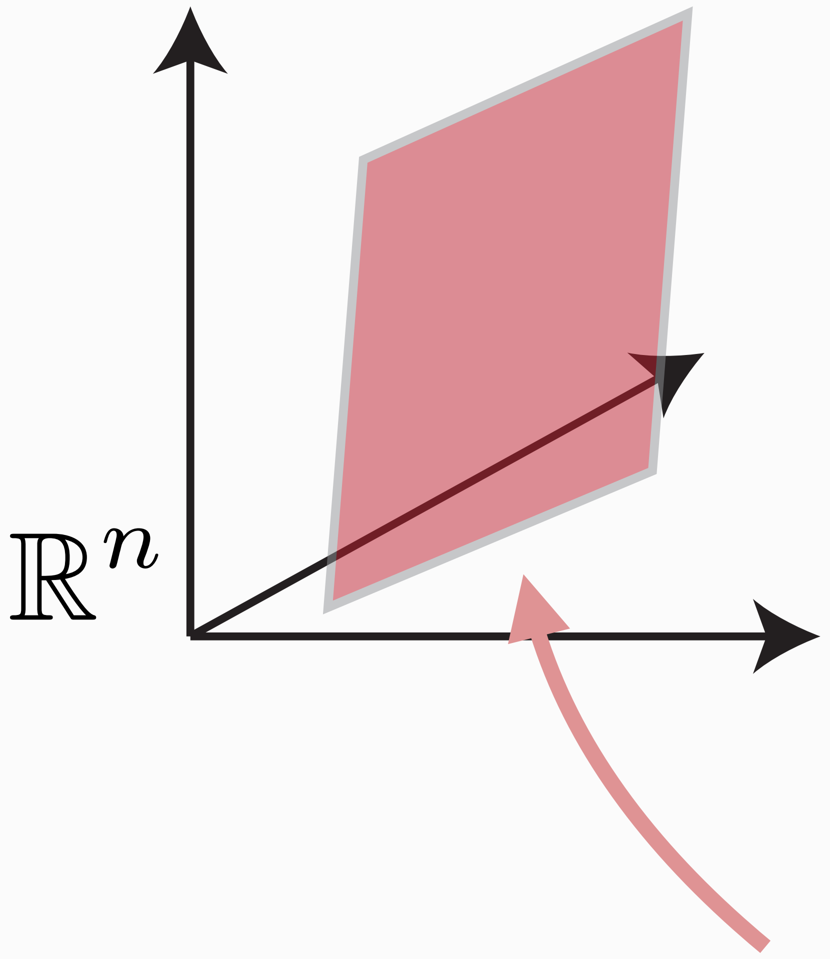




2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

... 2D search subtask





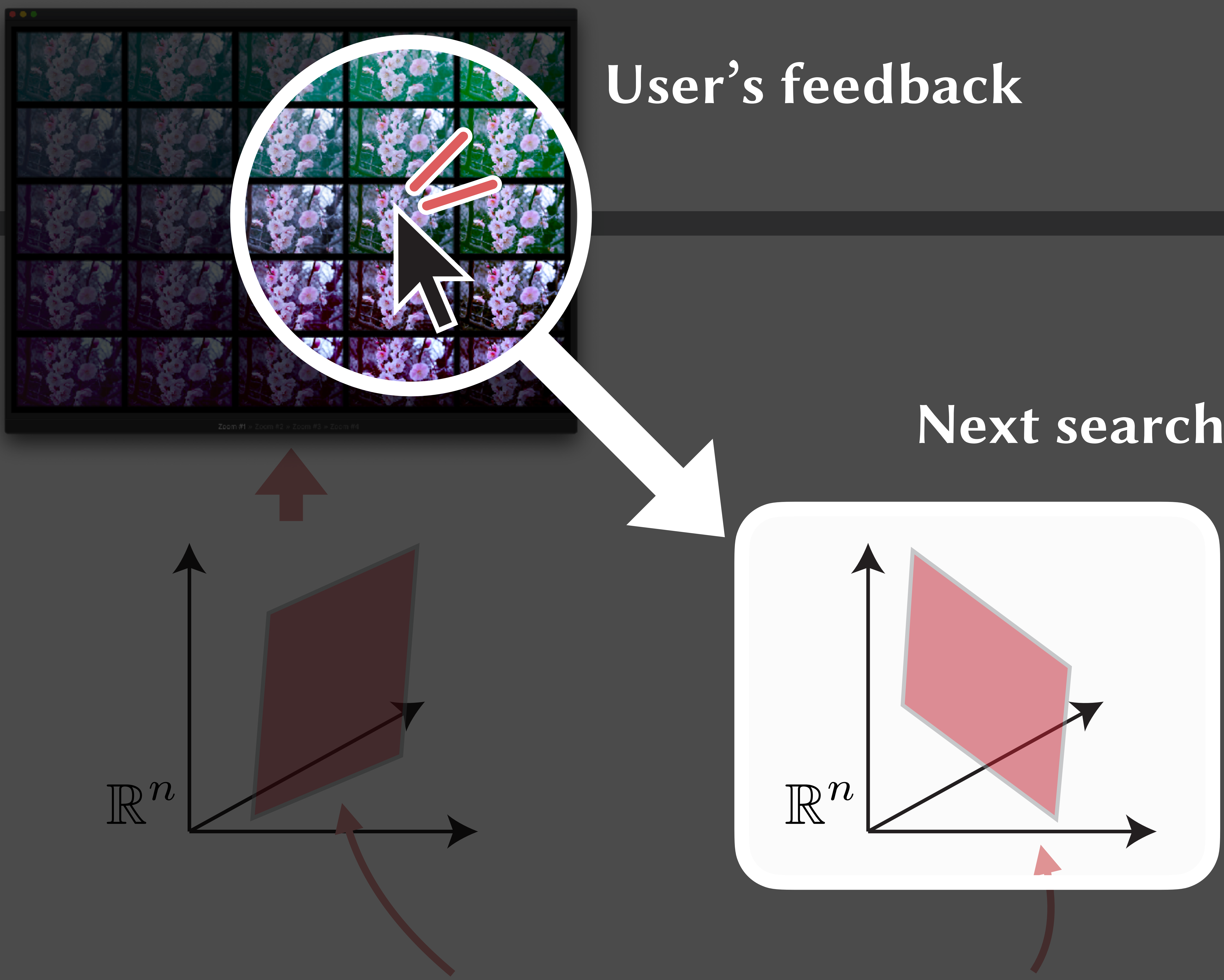
2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

Yuki Koyama, Issei Sato, and Masataka Goto, Sequential Gallery for Interactive Visual Design Optimization. ACM Trans. Graph. (SIGGRAPH 2020)

... 2D search subtask

User's feedback

Next search plane



2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

... 2D search subtask

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

… 2D search subtask

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm


...
2D search subtask
2D search subtask
Next search plane
User's feedback
$\mathbb{R}^n$
$\mathbb{R}^n$
$\mathbb{R}^n$
2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

... 2D search subtask 2D search subtask

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

Yuki Koyama, Issei Sato, and Masataka Goto, Sequential Gallery for Interactive Visual Design Optimization. ACM Trans. Graph. (SIGGRAPH 2020)

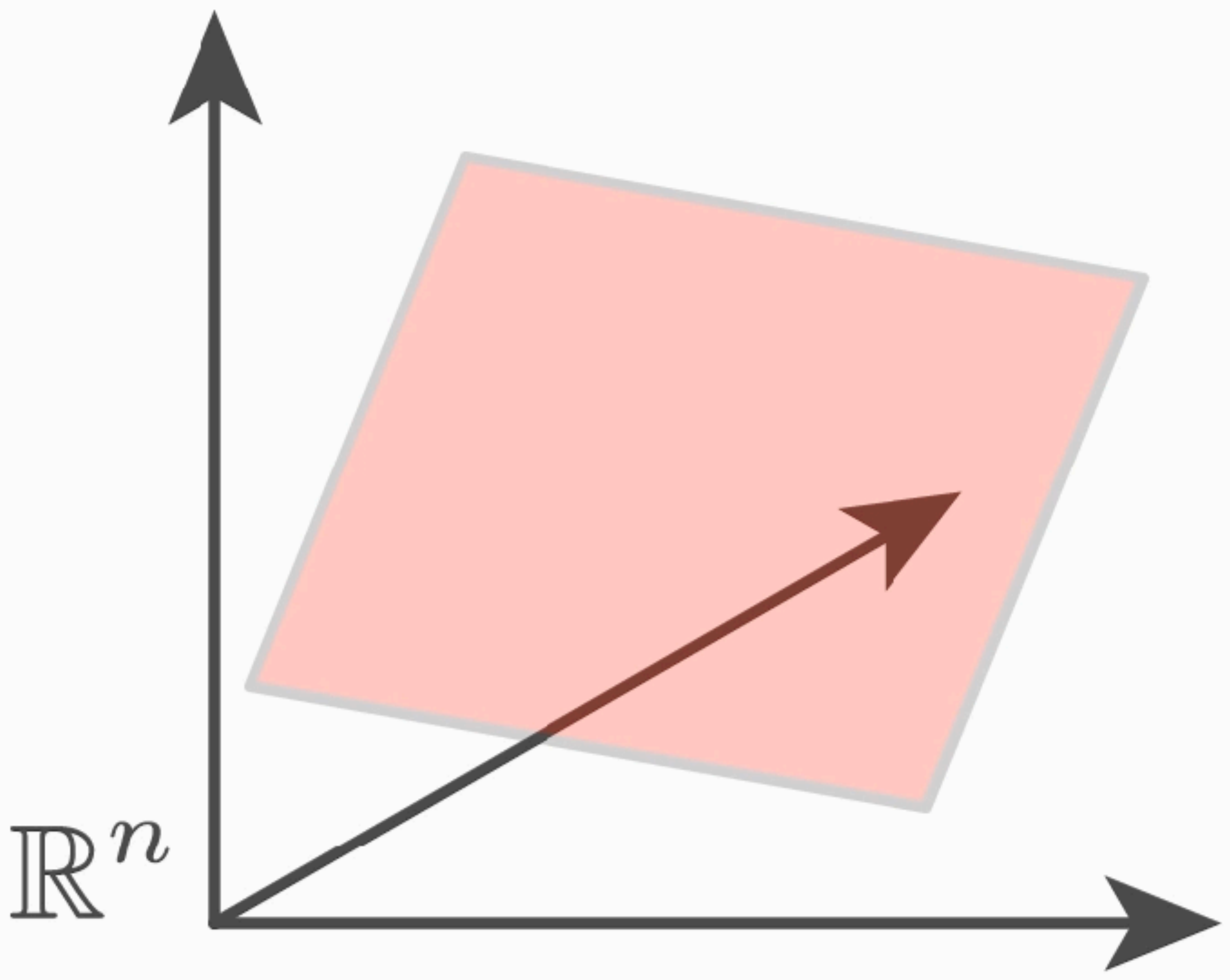
2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

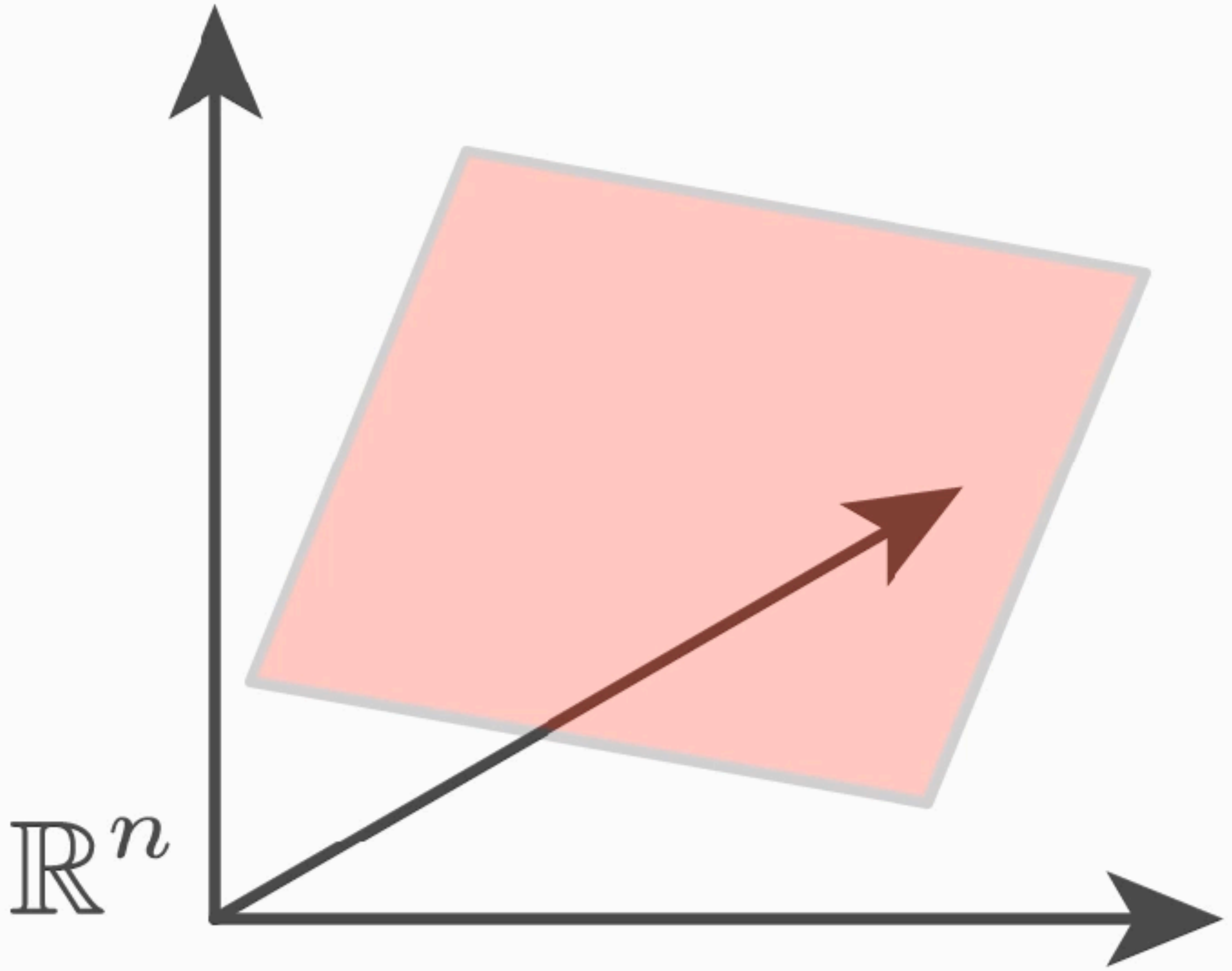
2-dimensional search subspaces (= **search planes**)
determined by the **sequential plane search** algorithm

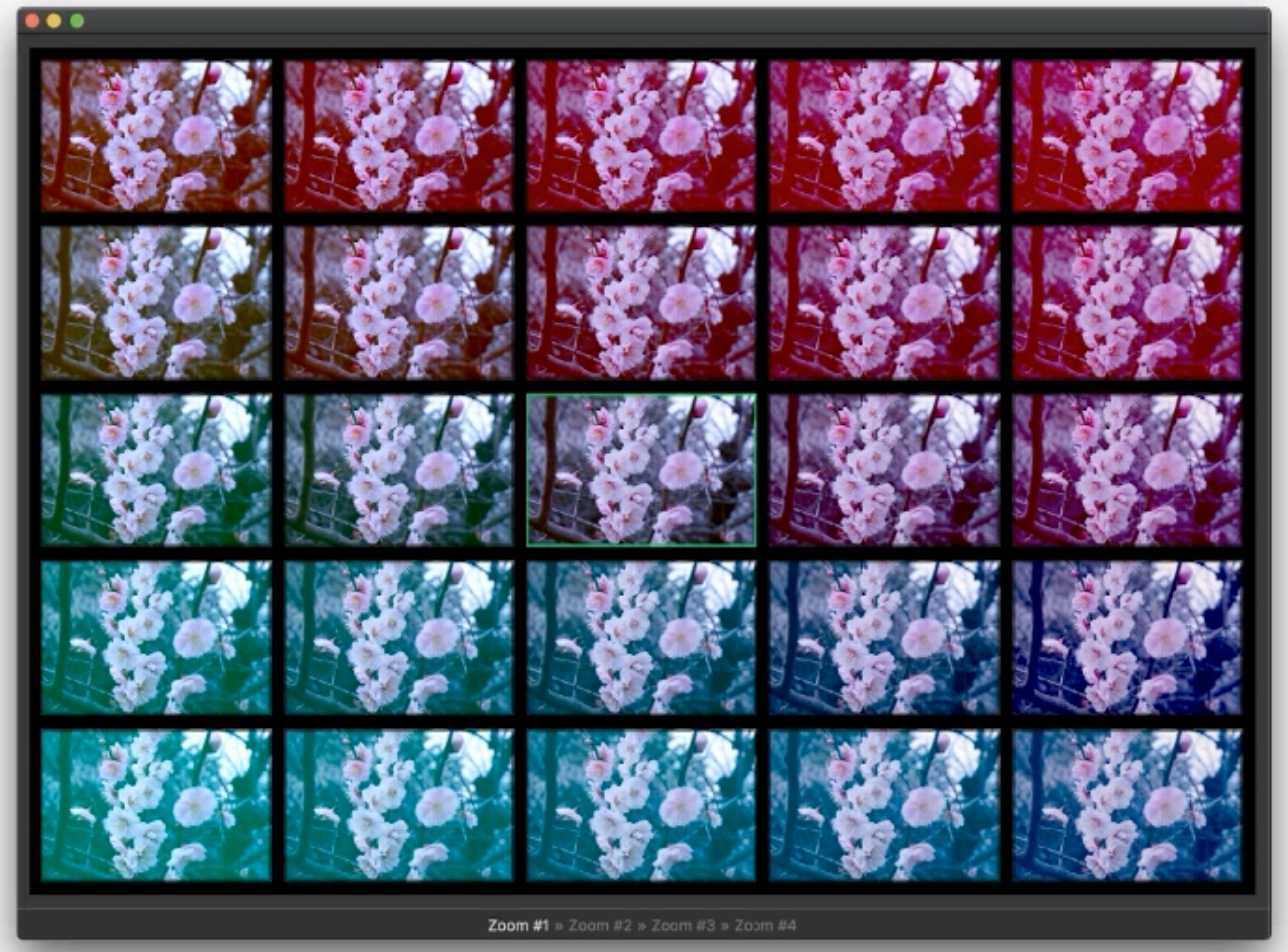
Interface: Zoomable Grid Interface for 2D Search Subtasks

Interface: Zoomable Grid Interface for 2D Search Subtasks

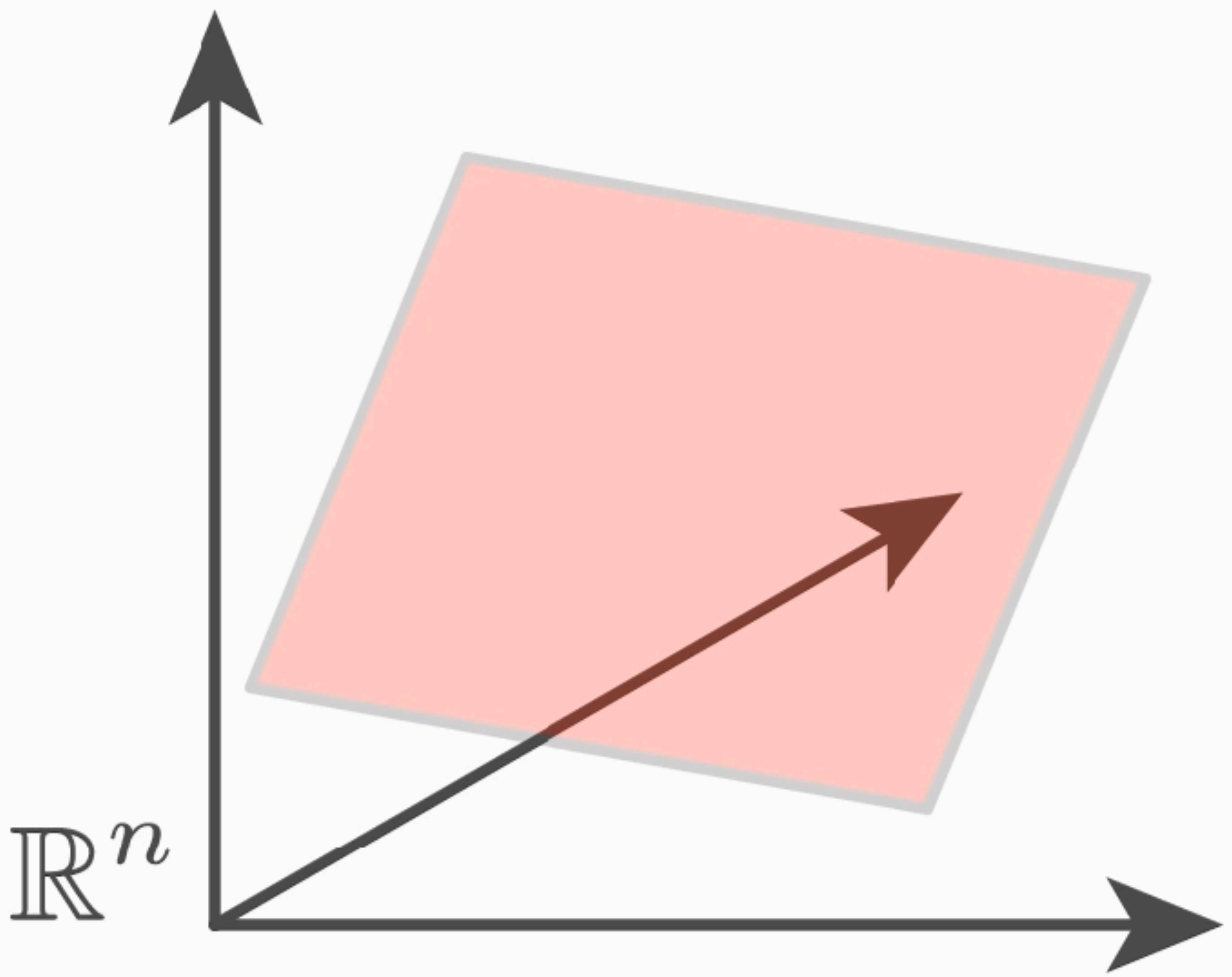


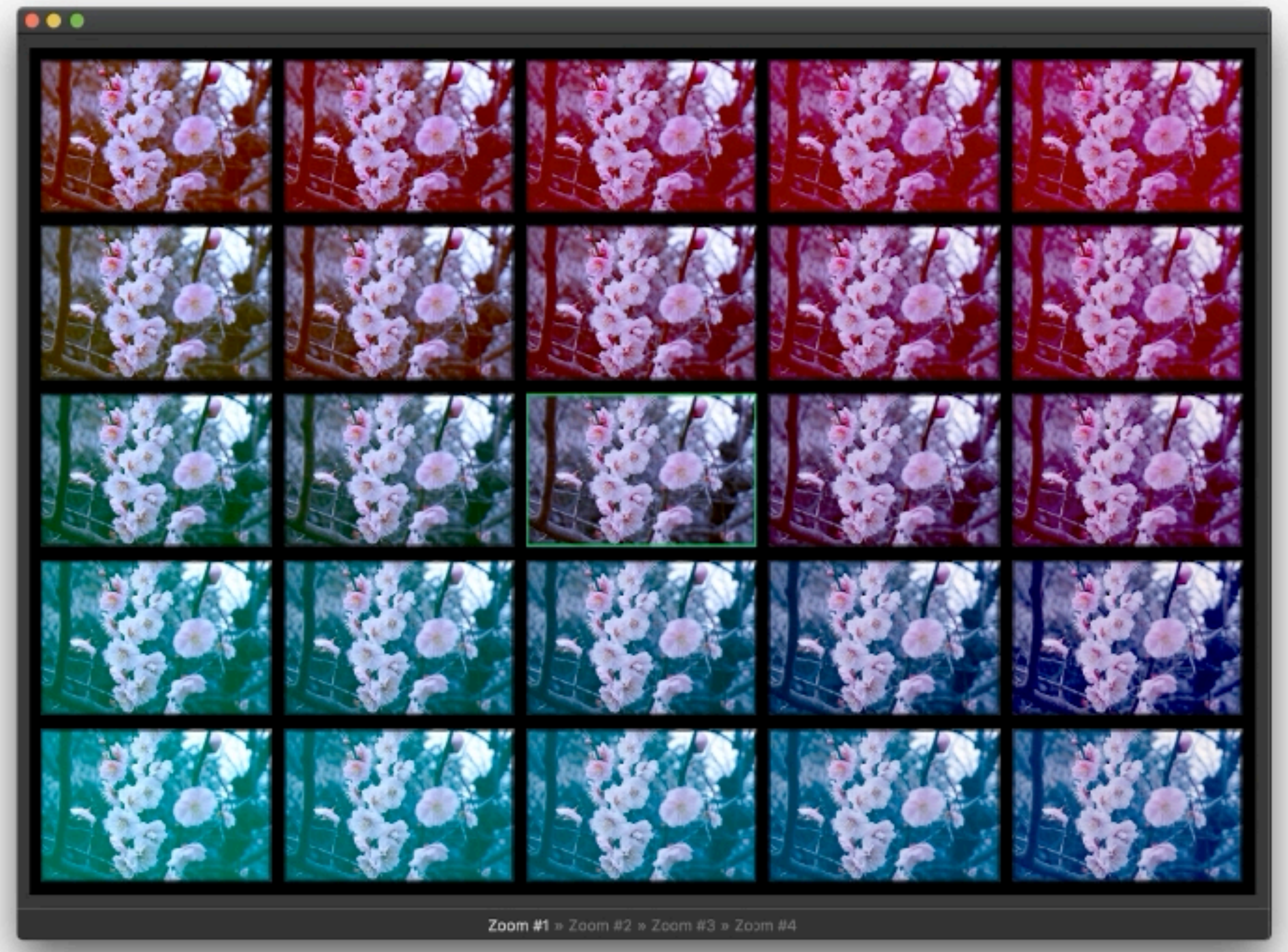
Interface: Zoomable Grid Interface for 2D Search Subtasks





Interface: Zoomable Grid Interface for 2D Search Subtasks

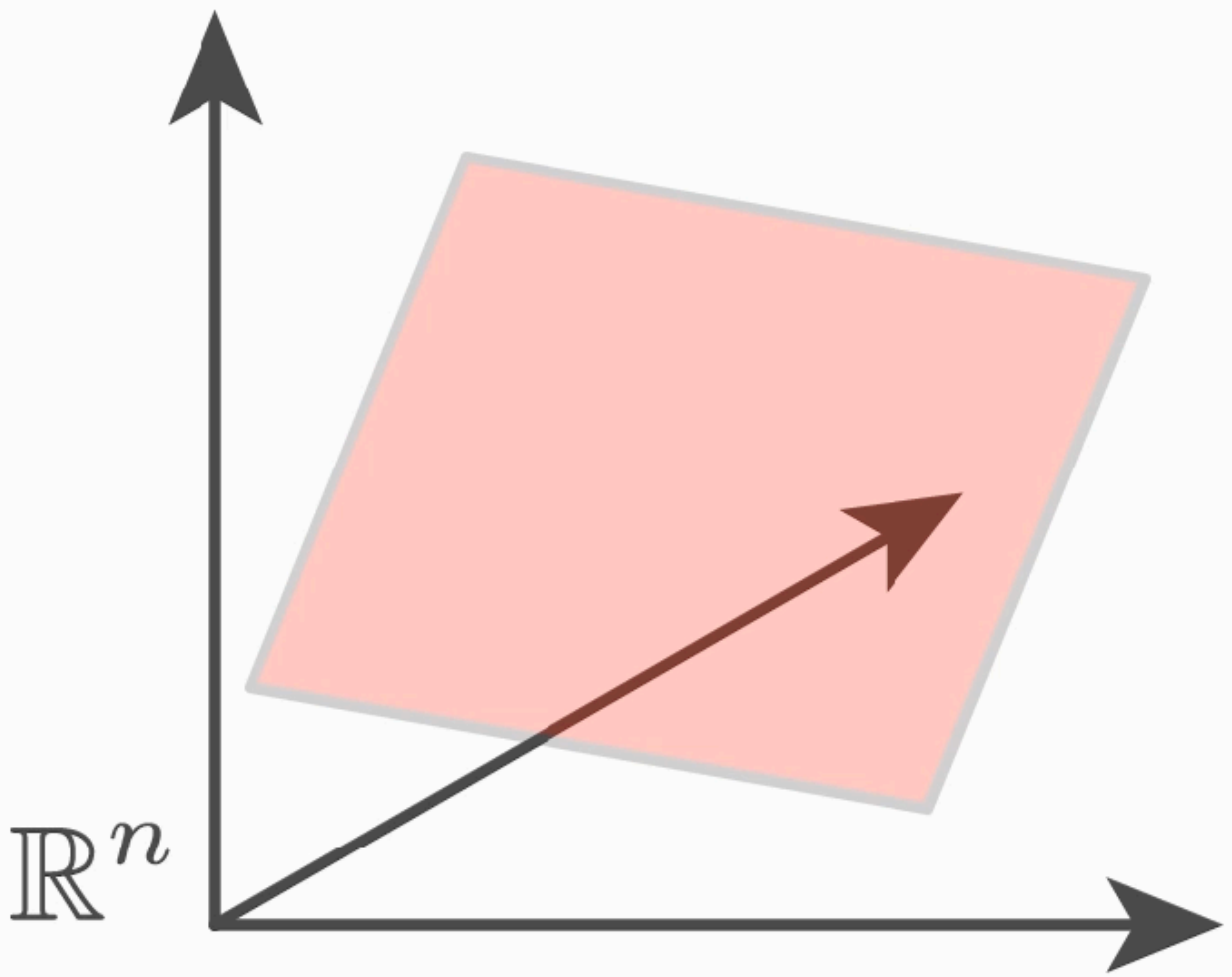


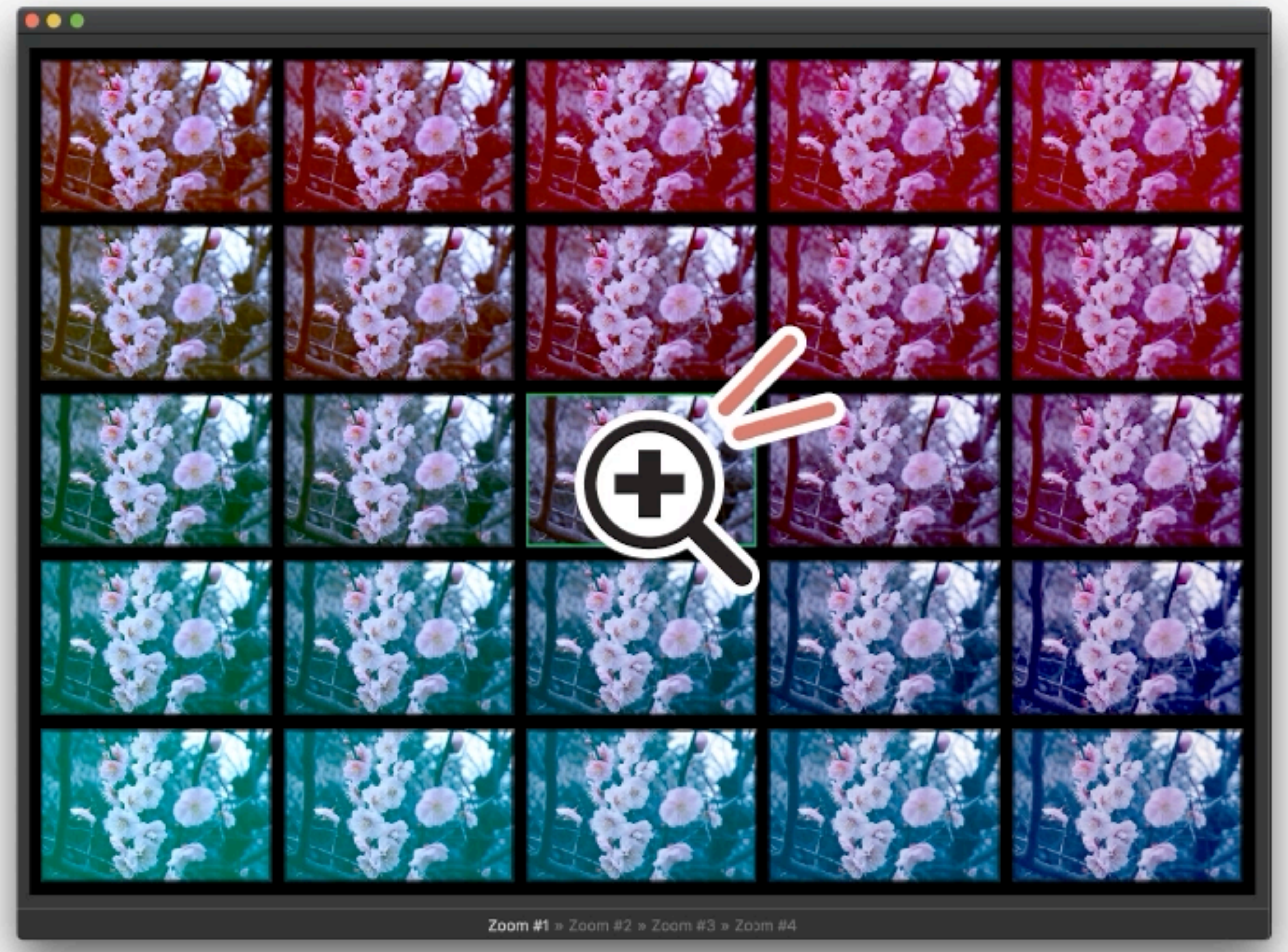


Current zoom level: 1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 4

[coarse] [fine]

Interface: Zoomable Grid Interface for 2D Search Subtasks

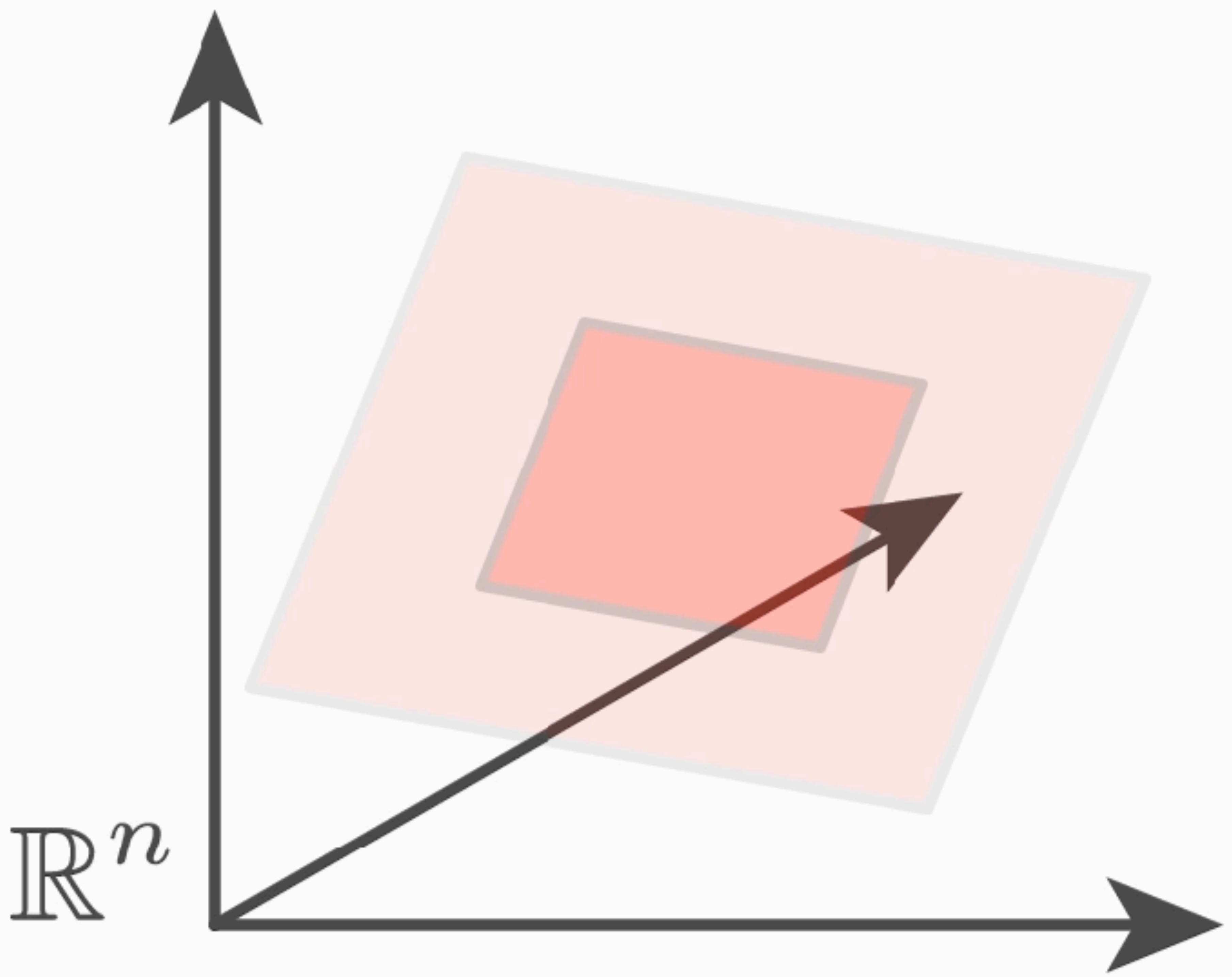




Current zoom level: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

[coarse] [fine]

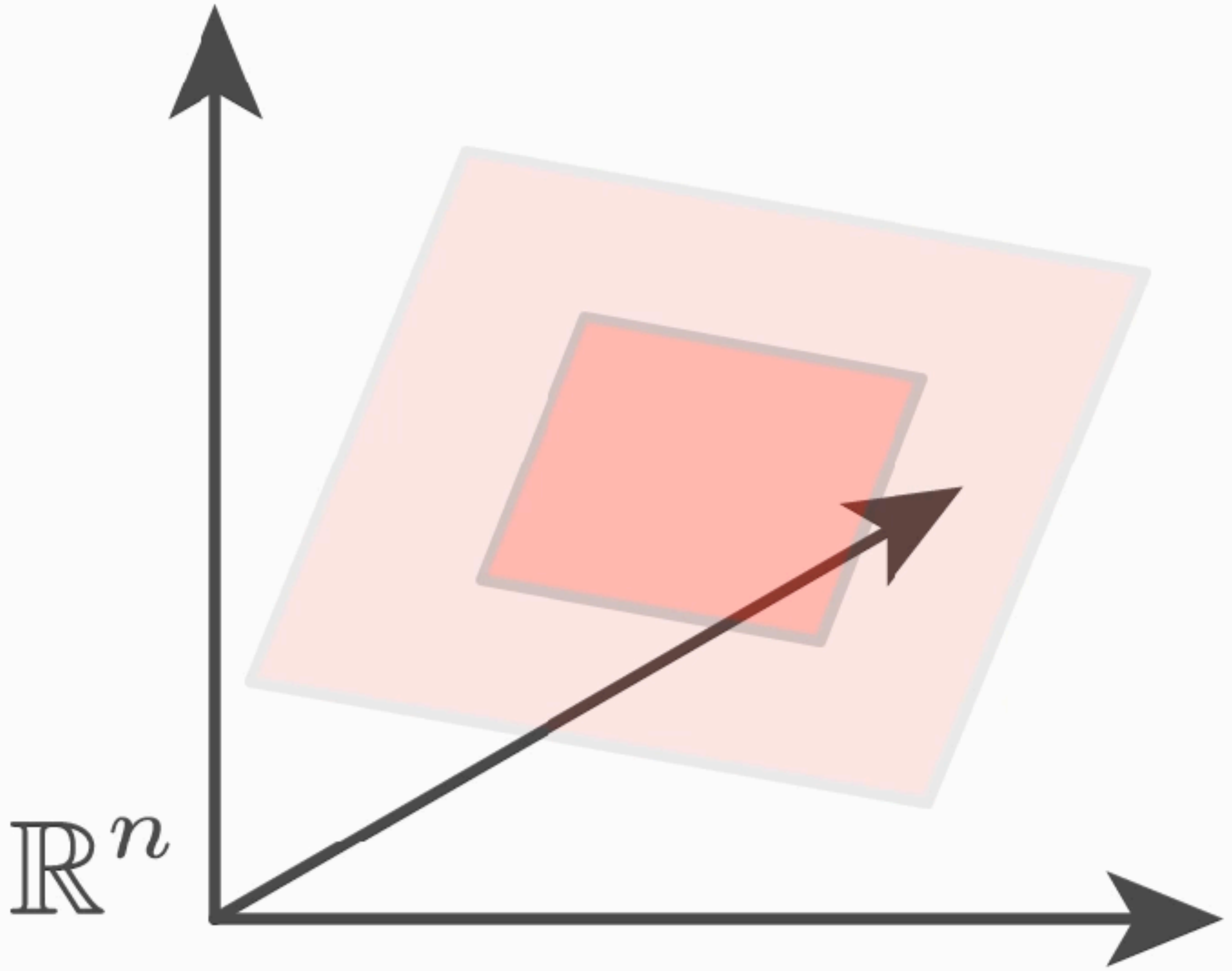
Interface: Zoomable Grid Interface for 2D Search Subtasks

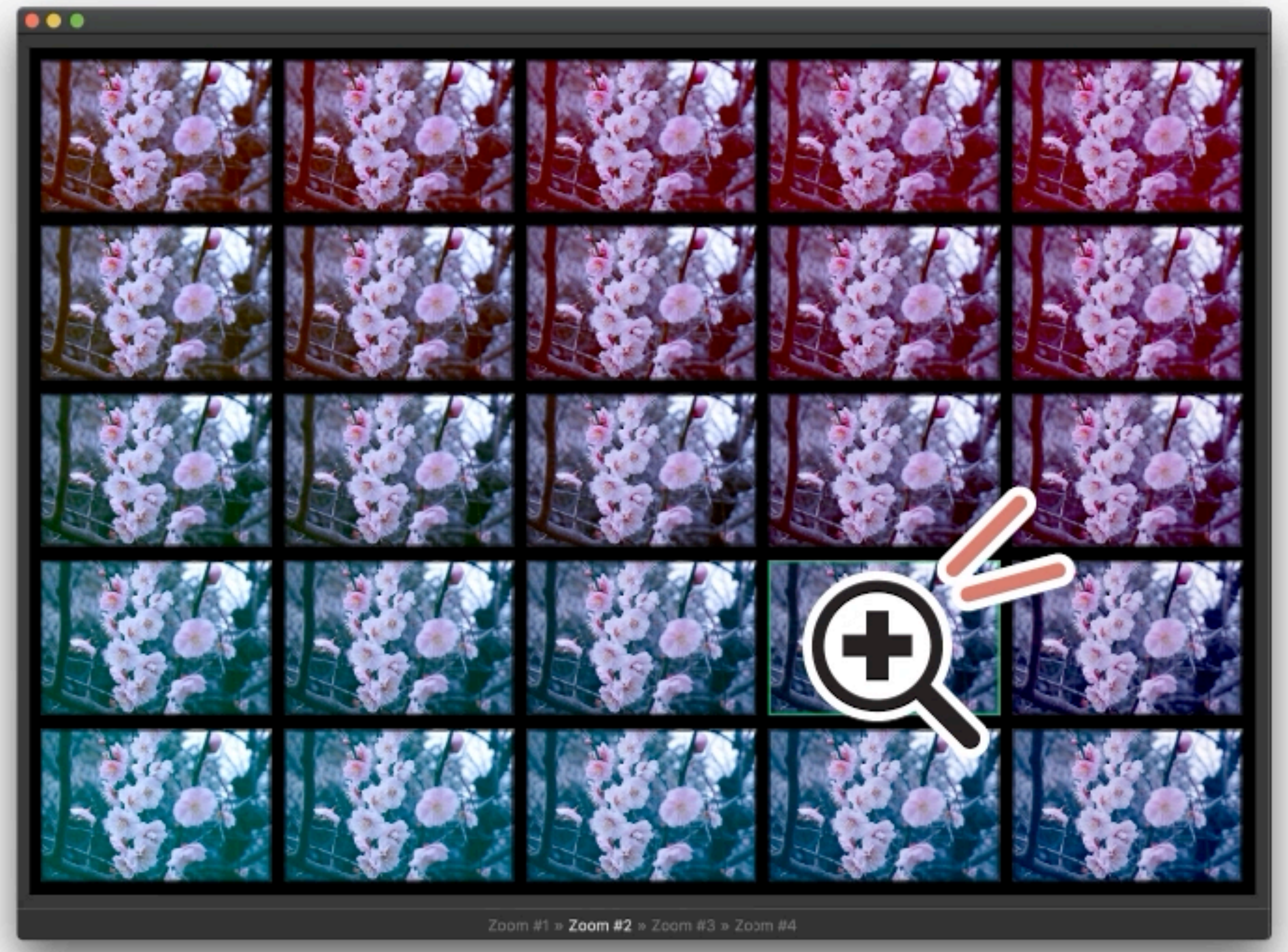


Current zoom level: 1 ⟶ **2** ⟶ 3 ⟶ 4

[coarse] [fine]

Interface: Zoomable Grid Interface for 2D Search Subtasks

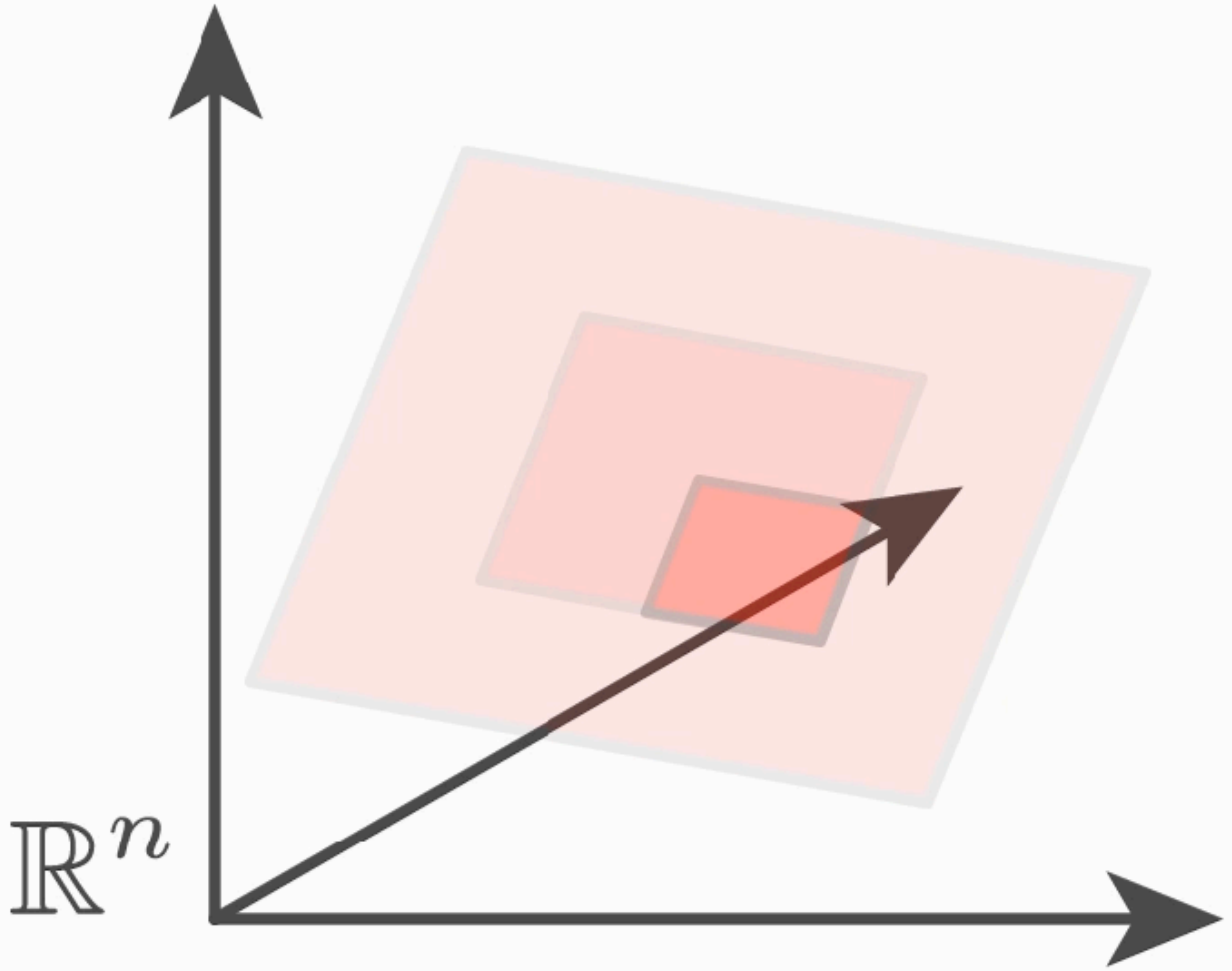


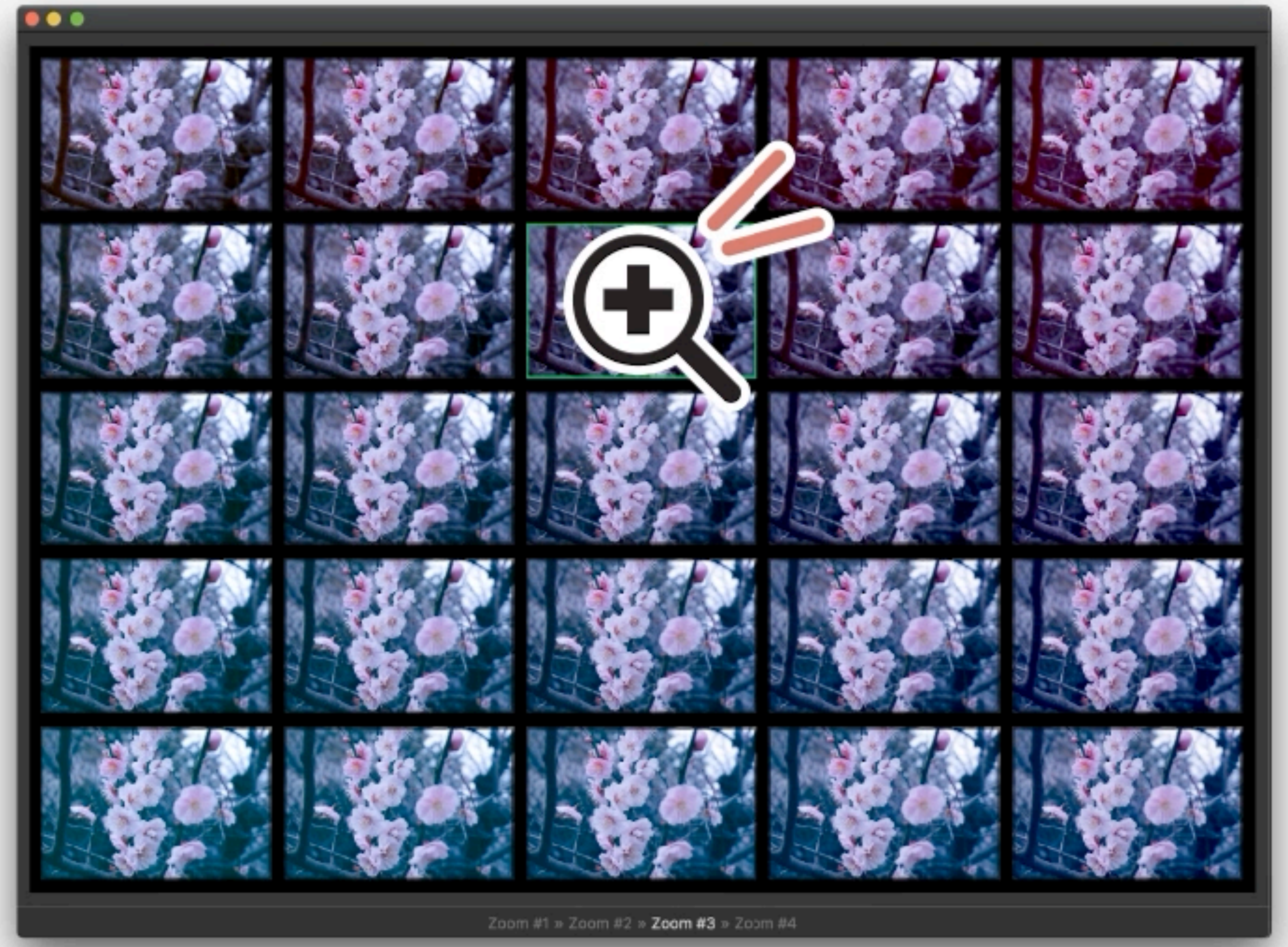


Current zoom level: 1 → **2** → 3 → 4

[coarse] [fine]

Interface: Zoomable Grid Interface for 2D Search Subtasks





Current zoom level: $1 \longrightarrow 2 \longrightarrow \mathbf{3} \longrightarrow 4$

[coarse] [fine]

Benefits of Zoomable Grid Interface

- **Allows the user to easily grasp the available options** in the 2D subspace by just seeing the grid view
- **WYSIWYG** (What-You-See-Is-What-You-Get); do not need to be aware of raw parameters
- Compatibility with the sequential-plane-search task (i.e., 2D search)
- Minimum quantization errors (thanks to zooming operations)

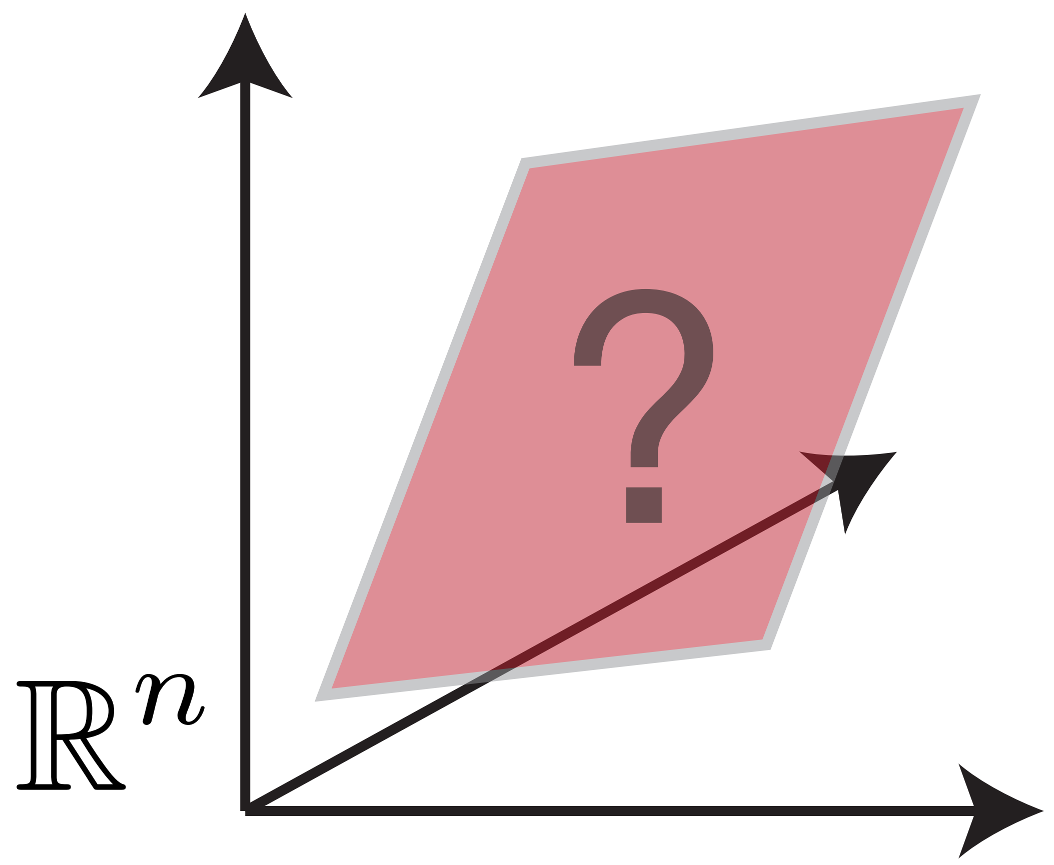
Sequential Plane Search

How to Determine the Next Plane in Each Iteration

Criterion to Select 2D Search Planes

We want to select the most **effective** 2D subspace (= search plane) for the next query

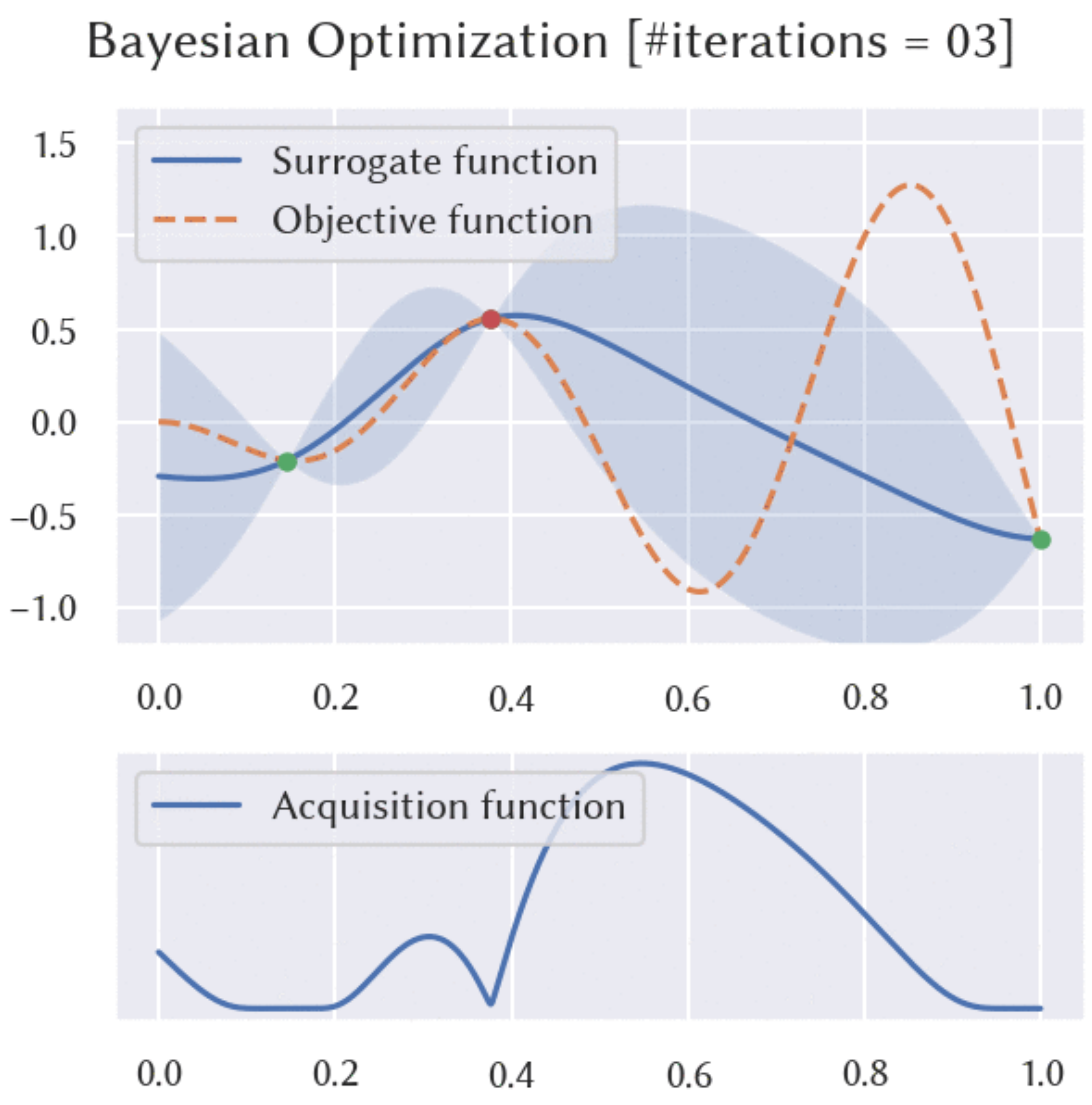
- **Effectiveness** here means the degree to which it is worth observing in the next iteration to find the optimal solution



Question: How can we define the effectiveness?

Acquisition Function in (Standard) BO

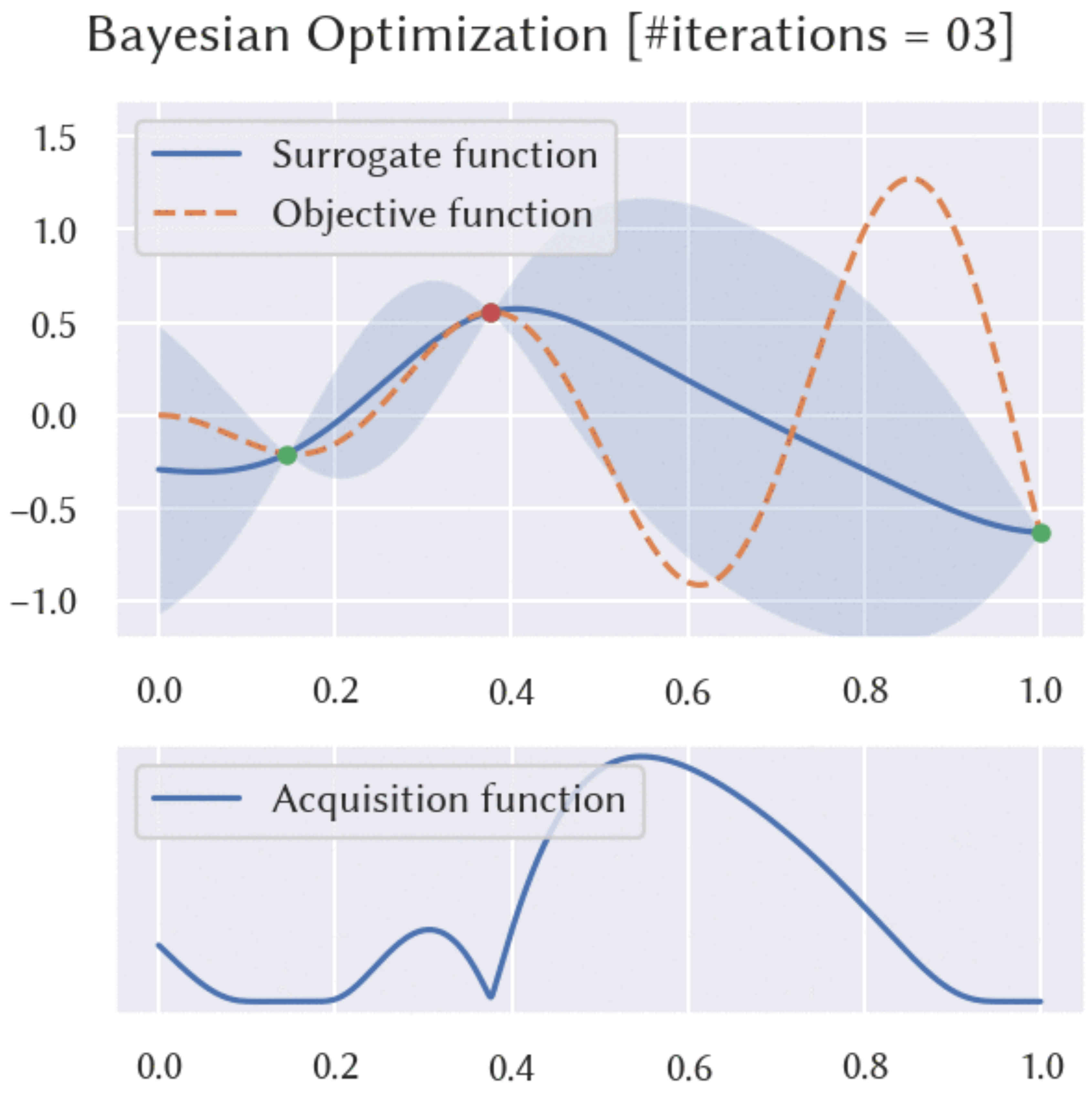
- Standard BO uses **acquisition functions**
  - An acquisition function evaluates **how effective a point is** as the next query
  - The maximizer of the acquisition function is selected as the next point



Note: There are many variants in acquisition function definitions (e.g., "expected improvement"). Most of them can automatically balance the "exploration" (i.e., favor unobserved regions) and "exploitation" (i.e., favor promising regions) strategies, which is the magic to enable BO to be successful. See [Shahriari+, Proc. IEEE 2016] for details.

Acquisition Function in (Standard) BO

- Standard BO uses **acquisition functions**
  - An acquisition function evaluates **how effective a point is** as the next query
  - The maximizer of the acquisition function is selected as the next point
- We extend this point-wise acquisition function to evaluate **how effective a search plane is** as the next query (next slide)



Note: There are many variants in acquisition function definitions (e.g., “expected improvement”). Most of them can automatically balance the “exploration” (i.e., favor unobserved regions) and “exploitation” (i.e., favor promising regions) strategies, which is the magic to enable BO to be successful. See [Shahriari+, Proc. IEEE 2016] for details.

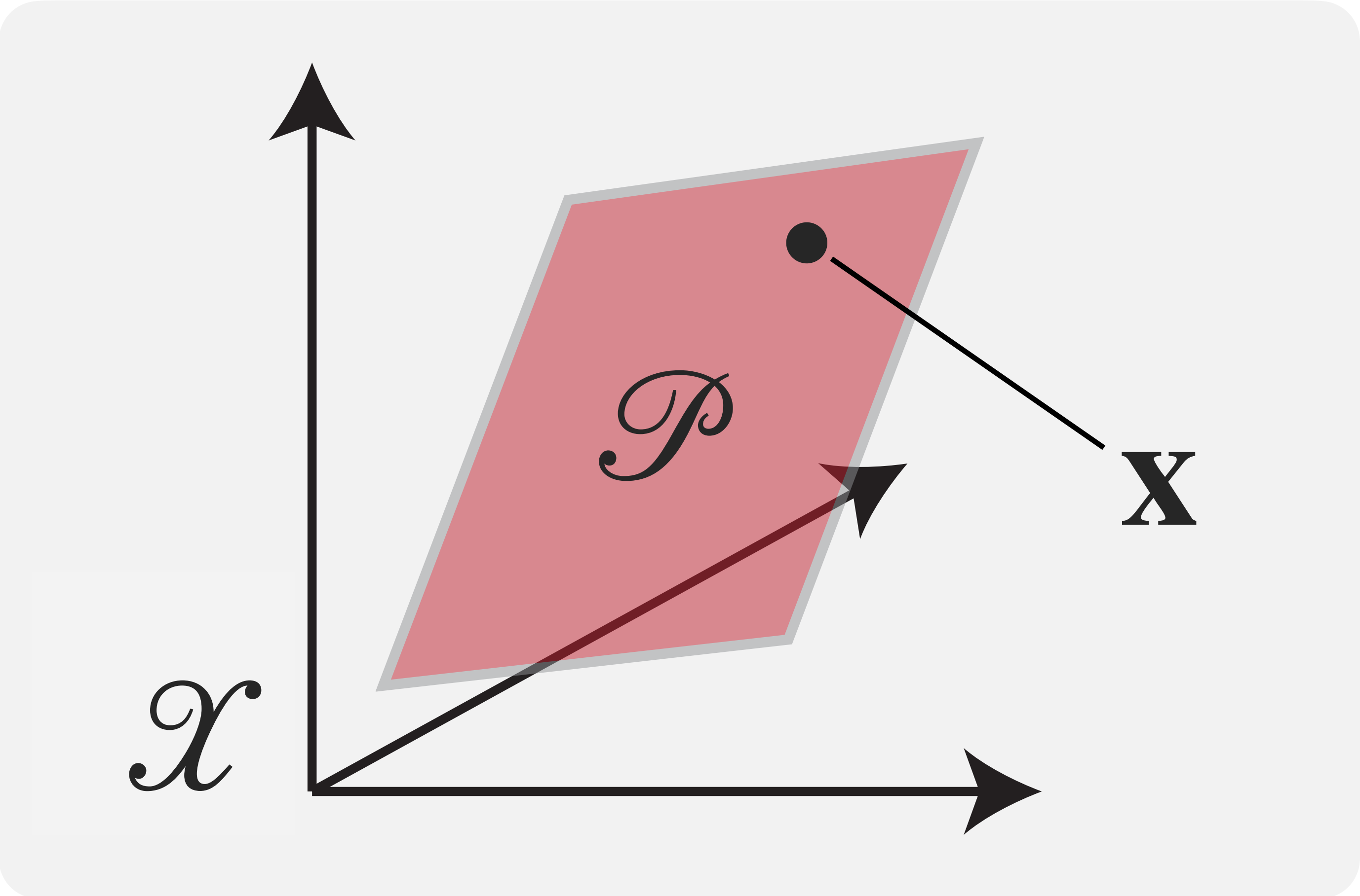
Acquisition Function for Determining the Next Plane

- Let $\mathscr{D}$ be the accumulated preferential data so far obtained
- **Point-wise** acquisition function (used in standard BO):

$$a^{\mathrm{point}}(\mathbf{x}; \mathscr{D})$$ [Note 1]

- **Plane-wise** acquisition function (proposed):

$$a^{\mathrm{plane}}(\mathscr{P}; \mathscr{D})$$



[Note 1]: Refer to [Koyama+, SIGGRAPH 2017] for the definition of a^{point} in PBO; in short, it is based on the use of Gaussian process regression to estimate the goodness function landscape from the available preferential data.

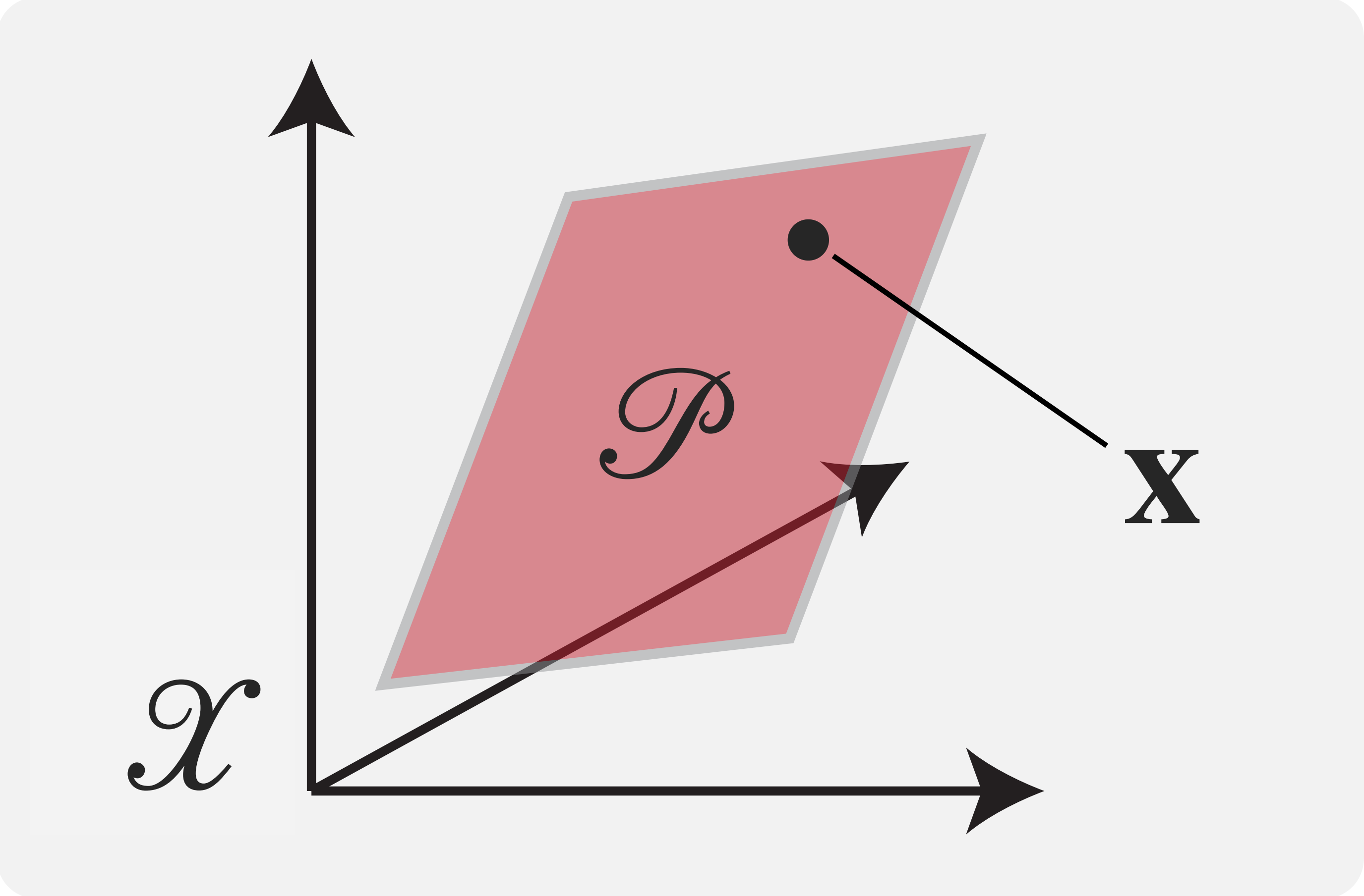
Acquisition Function for Determining the Next Plane

- Let $\mathscr{D}$ be the accumulated preferential data so far obtained
- **Point-wise** acquisition function (used in standard BO):

$$a^{\mathrm{point}}(\mathbf{x}; \mathscr{D})$$ [Note 1]

- **Plane-wise** acquisition function (proposed):

$$a^{\mathrm{plane}}(\mathscr{P}; \mathscr{D}) = \int_{\mathscr{P}} a^{\mathrm{point}}(\mathbf{x}; \mathscr{D}) dS$$



[Note 1]: Refer to [Koyama+, SIGGRAPH 2017] for the definition of a^{point} in PBO; in short, it is based on the use of Gaussian process regression to estimate the goodness function landscape from the available preferential data.

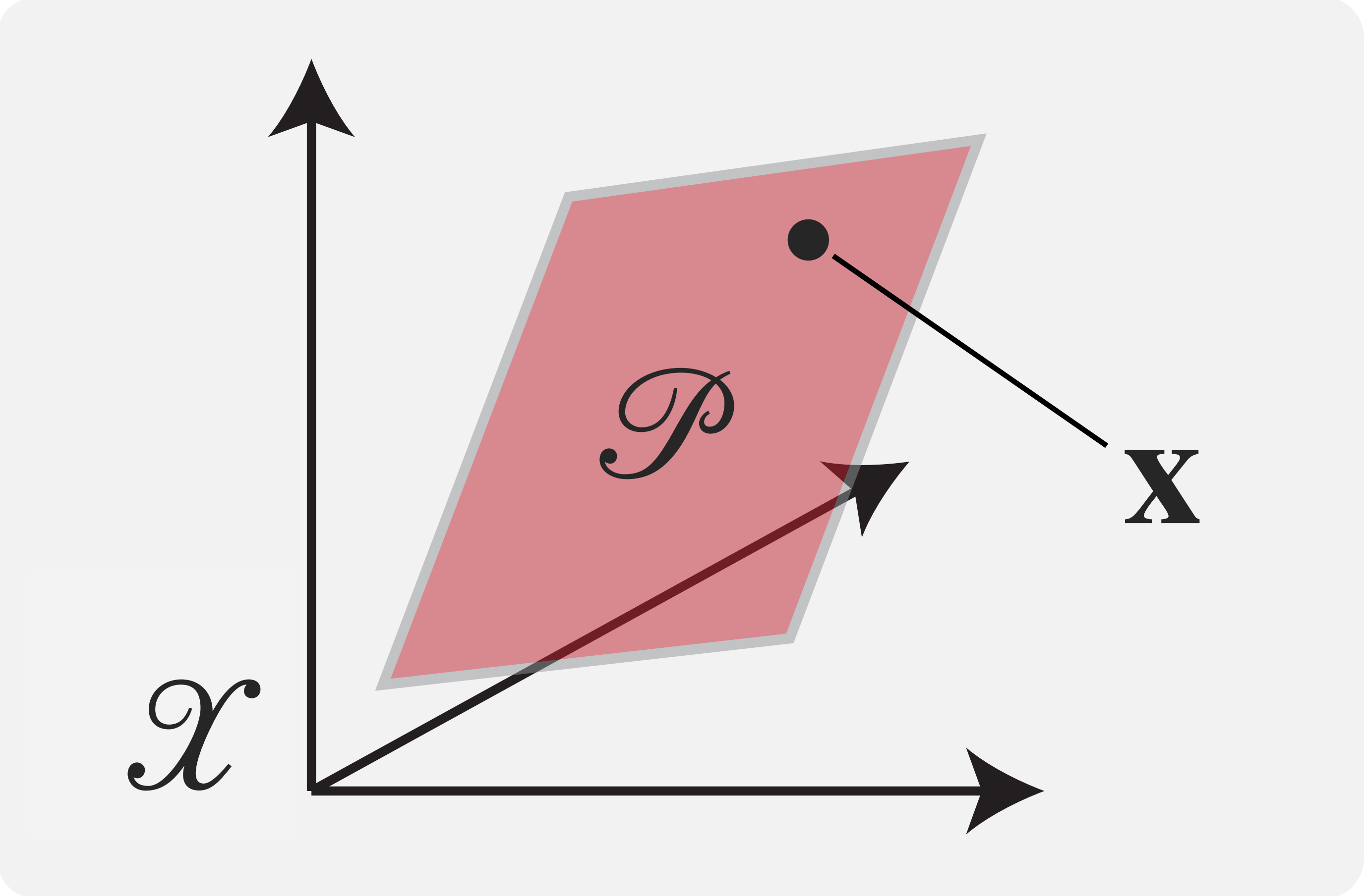
Acquisition Function for Determining the Next Plane

- Let $\mathscr{D}$ be the accumulated preferential data so far obtained
- **Point-wise** acquisition function (used in standard BO):

$$a^{\text{point}}(\mathbf{x}; \mathscr{D})$$ [Note 1]

- **Plane-wise** acquisition function (proposed):

$$a^{\text{plane}}(\mathscr{P}; \mathscr{D}) = \int_{\mathscr{P}} a^{\text{point}}(\mathbf{x}; \mathscr{D}) dS$$



- We choose the next plane $\mathscr{P}^{\text{next}}$ by solving a maximization problem:

$$\mathscr{P}^{\text{next}} = \mathop{\text{argmax}}_{\mathscr{P} \in \mathscr{X}} a^{\text{plane}}(\mathscr{P}; \mathscr{D})$$ [Note 2]

[Note 1]: Refer to [Koyama+, SIGGRAPH 2017] for the definition of a^{point} in PBO; in short, it is based on the use of Gaussian process regression to estimate the goodness function landscape from the available preferential data.

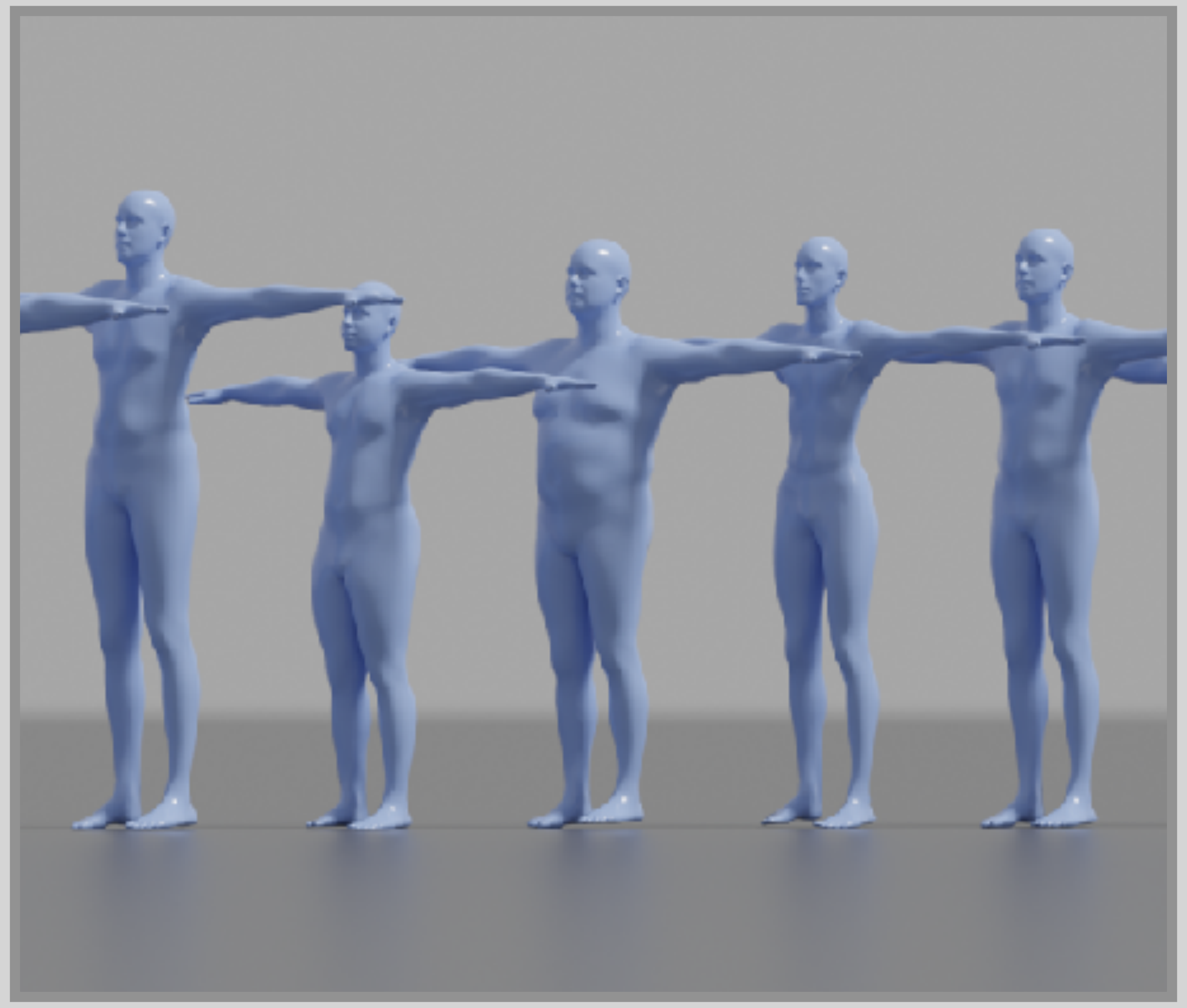
[Note 2]: This maximization problem is differentiable, so L-BFGS can be used. Since it can have multiple local maxima, we solve this problem multiple times with random initialization. It takes less than one sec. in most cases, which is acceptable from the interaction viewpoint.

Applications:
Possible Scenarios and Demonstrations

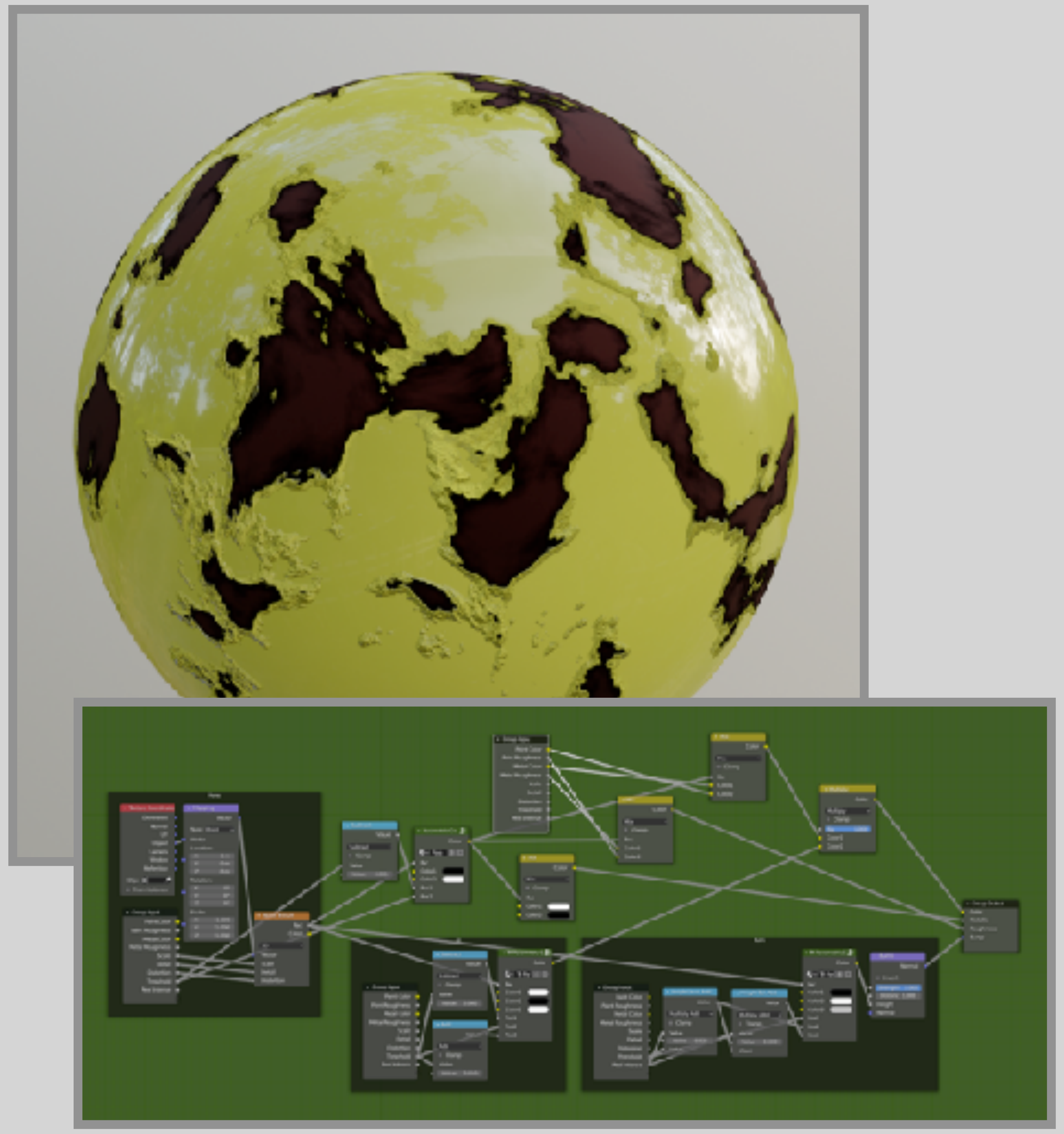
Potential Applications

Photo color enhancement

Generative modeling



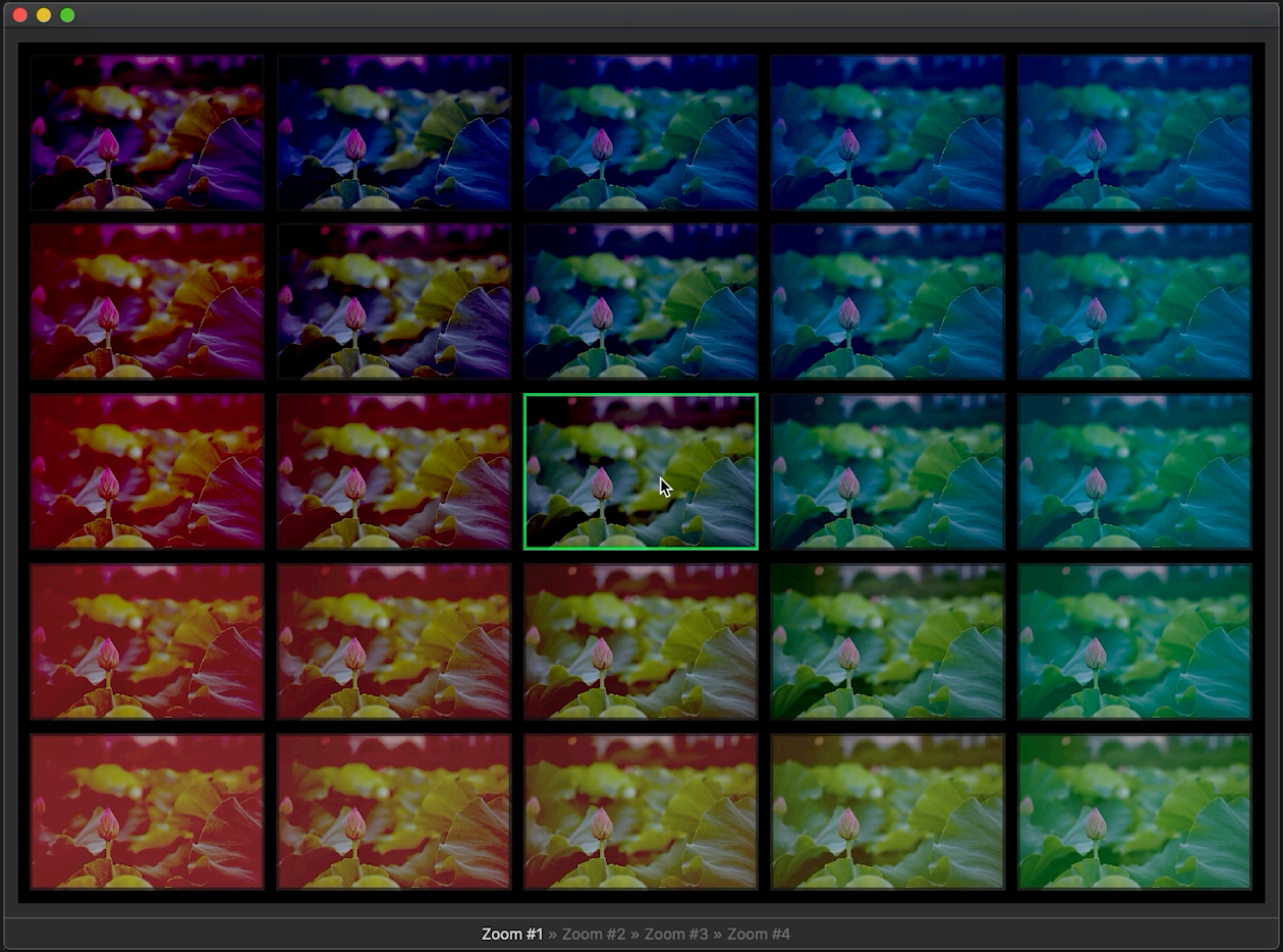
Procedural texturing



… and many other parametric design scenarios

Photo Color Enhancement (12D)

Brightness, contrast, saturation, shadows (RGB), midtones (RGB), and highlights (RGB)



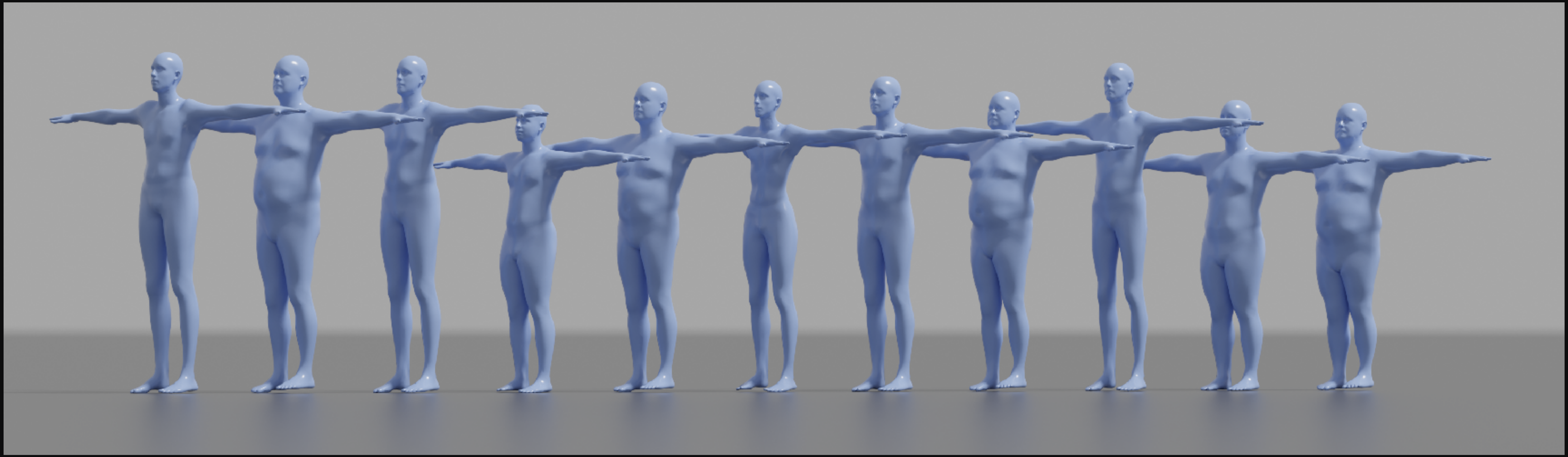
x2 speed

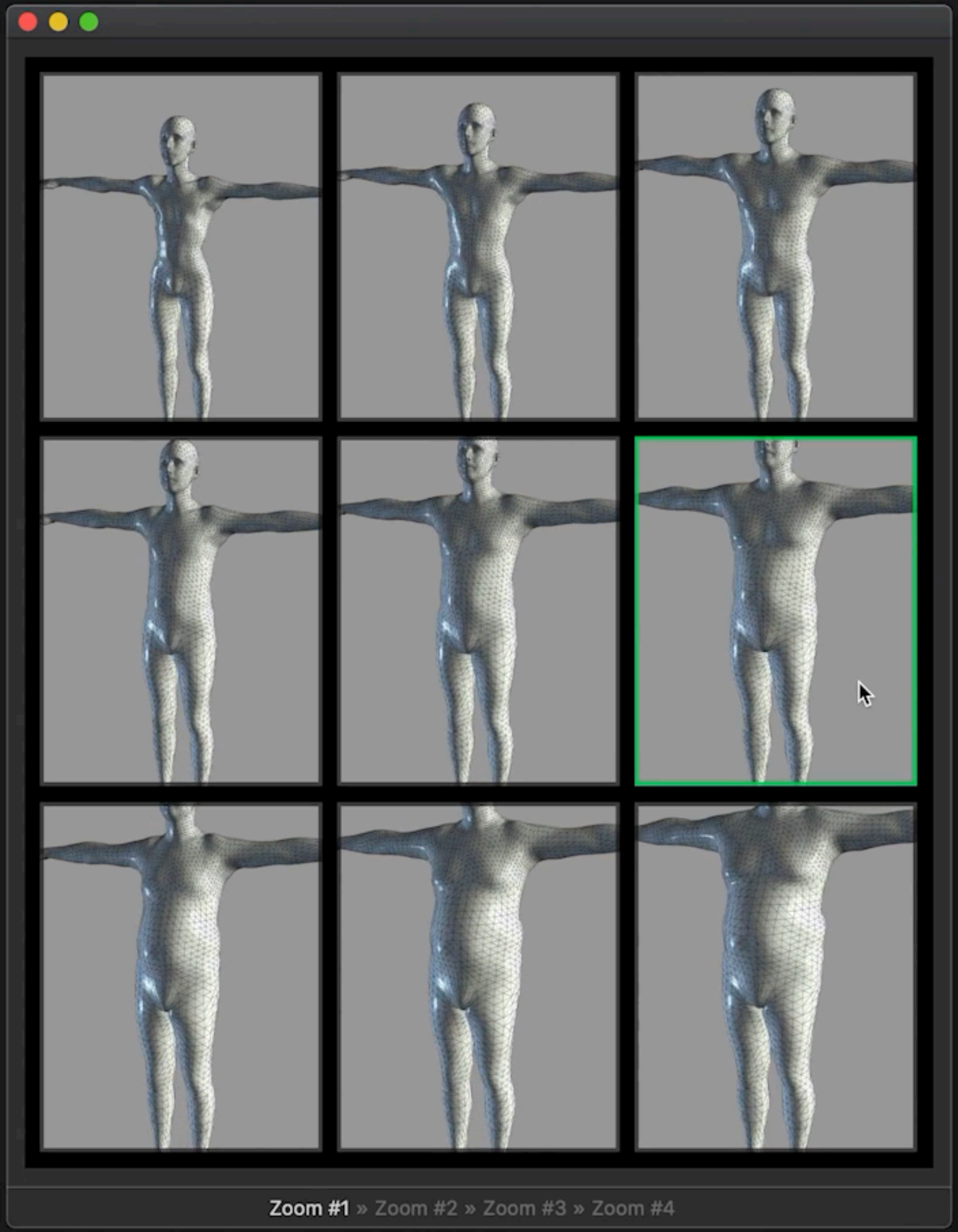
Original photograph

Enhanced photograph
(after 5 iterations)

Body Shaping (10D)

Using the SMPL model [Loper+15] (the first 10 principal components)





x1.5 speed

Goal: Body shaping from a character description

"He was of medium height, solidly built, wide in the shoulders, thick in the neck, with a jovial heavy-jawed red face [...]"

Dashiell Hammett. 1930. *The Maltese Falcon.*

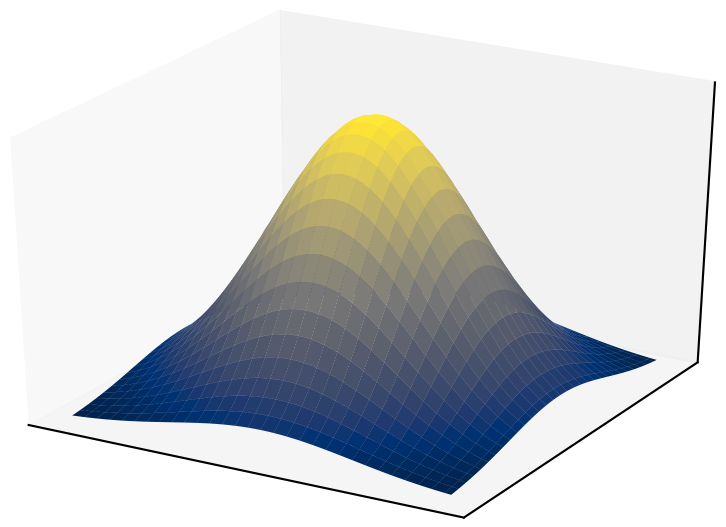
“He was of medium height, solidly built, wide in the shoulders, thick in the neck, with a jovial heavy-jawed red face […]”

Dashiell Hammett. 1930. *The Maltese Falcon.*

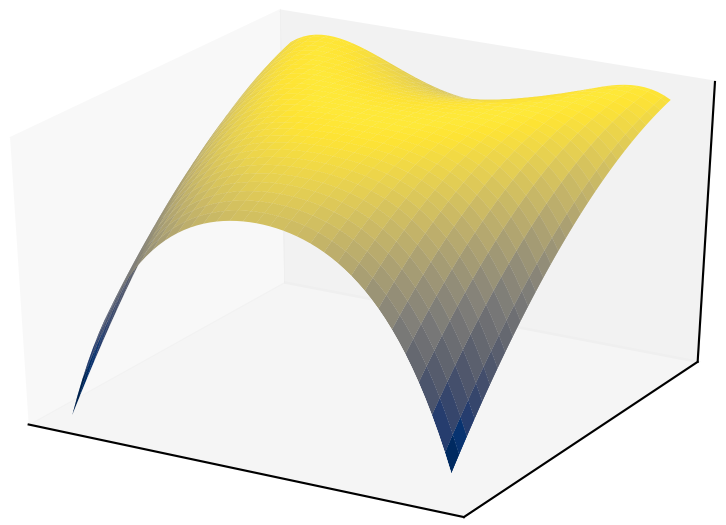
Evaluation [1/2]:
Optimization Performance Comparison

Performance Comparison Using Synthetic Functions

- **Goals:**
  - Evaluate the efficiency of our sequential plane search compared to the previous work [Koyama+17]
  - Confirm that BO (i.e., the use of the acquisition function) contributes to the optimization performance
- **How:**
  - Use **synthetic objective functions** to simulate human responses (shown on the right)
- **Algorithms to be compared:** (next slide)



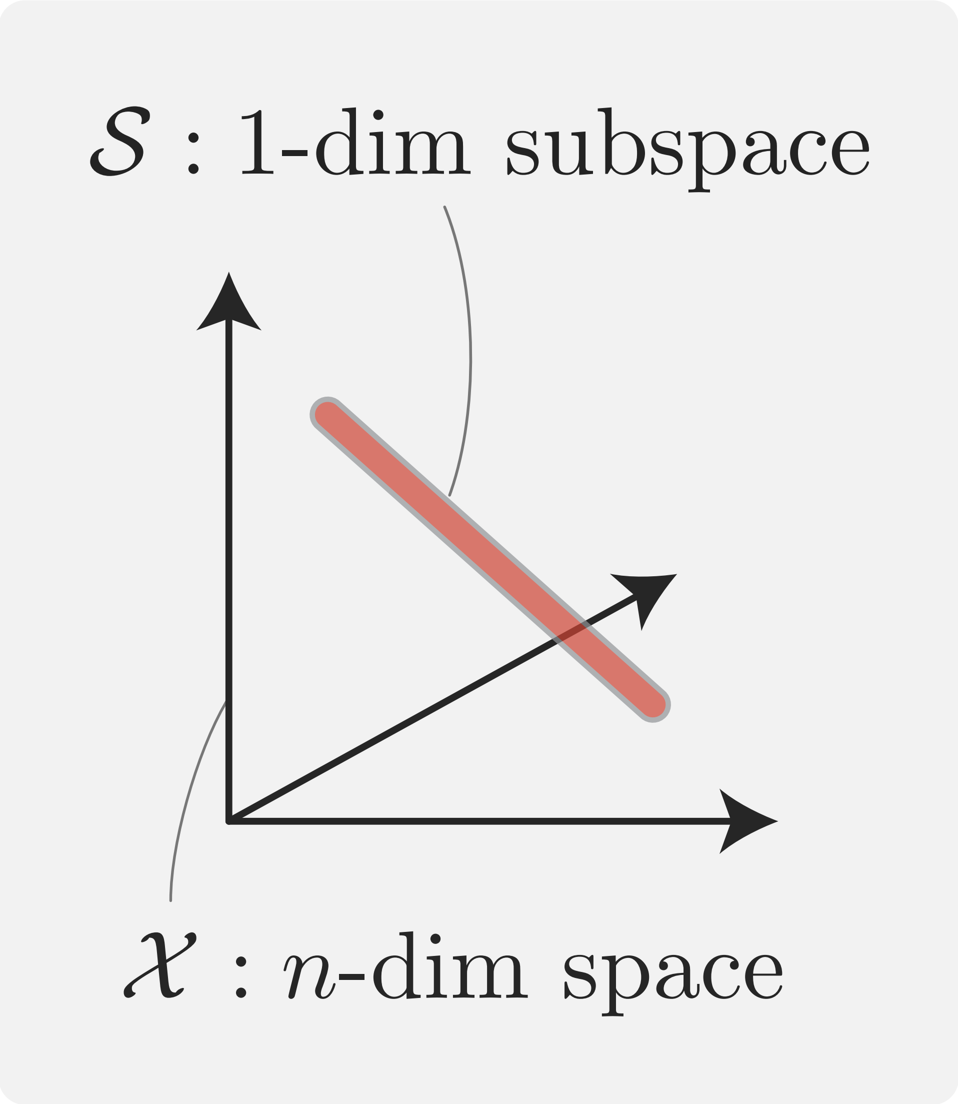
Gaussian (5D/15D)



Rosenbrock (10D/20D)

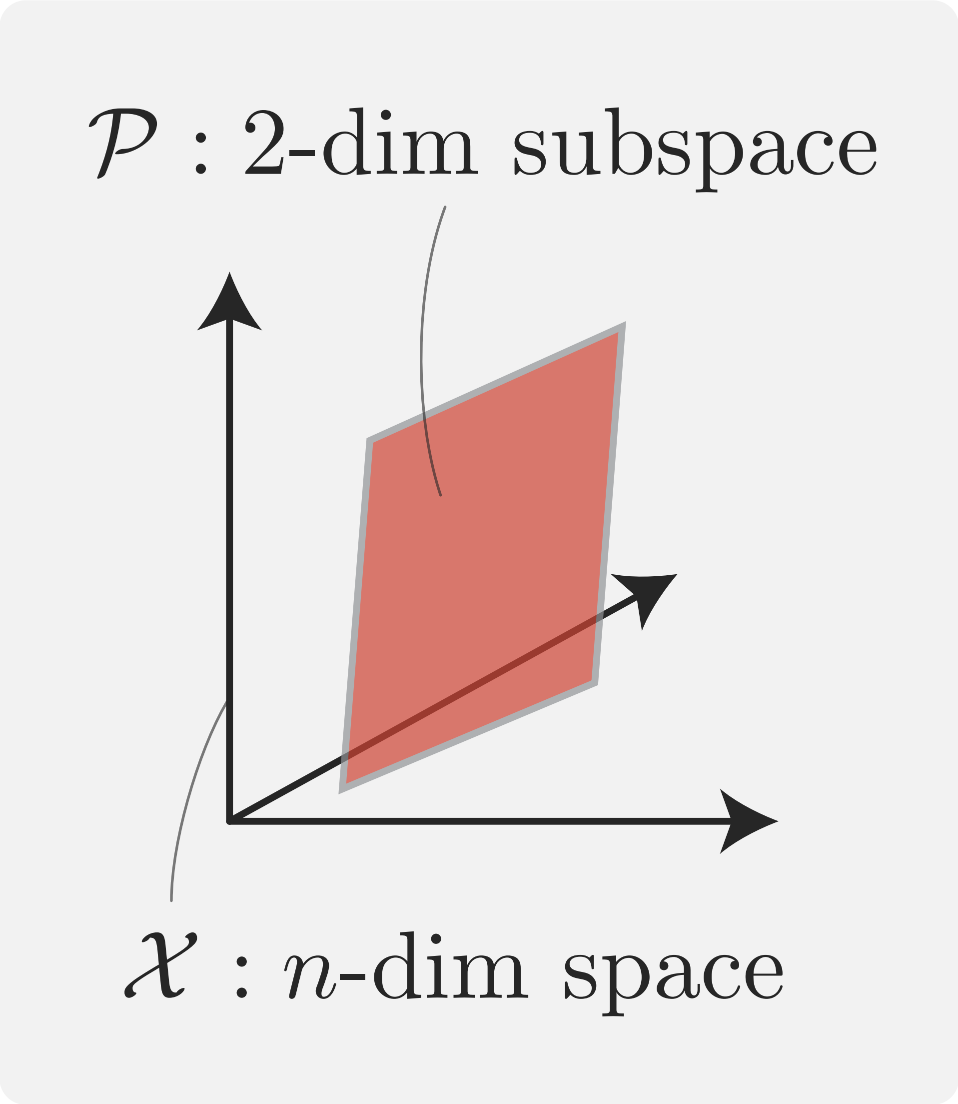
Algorithms to be Compared

Baseline 1:
SLS



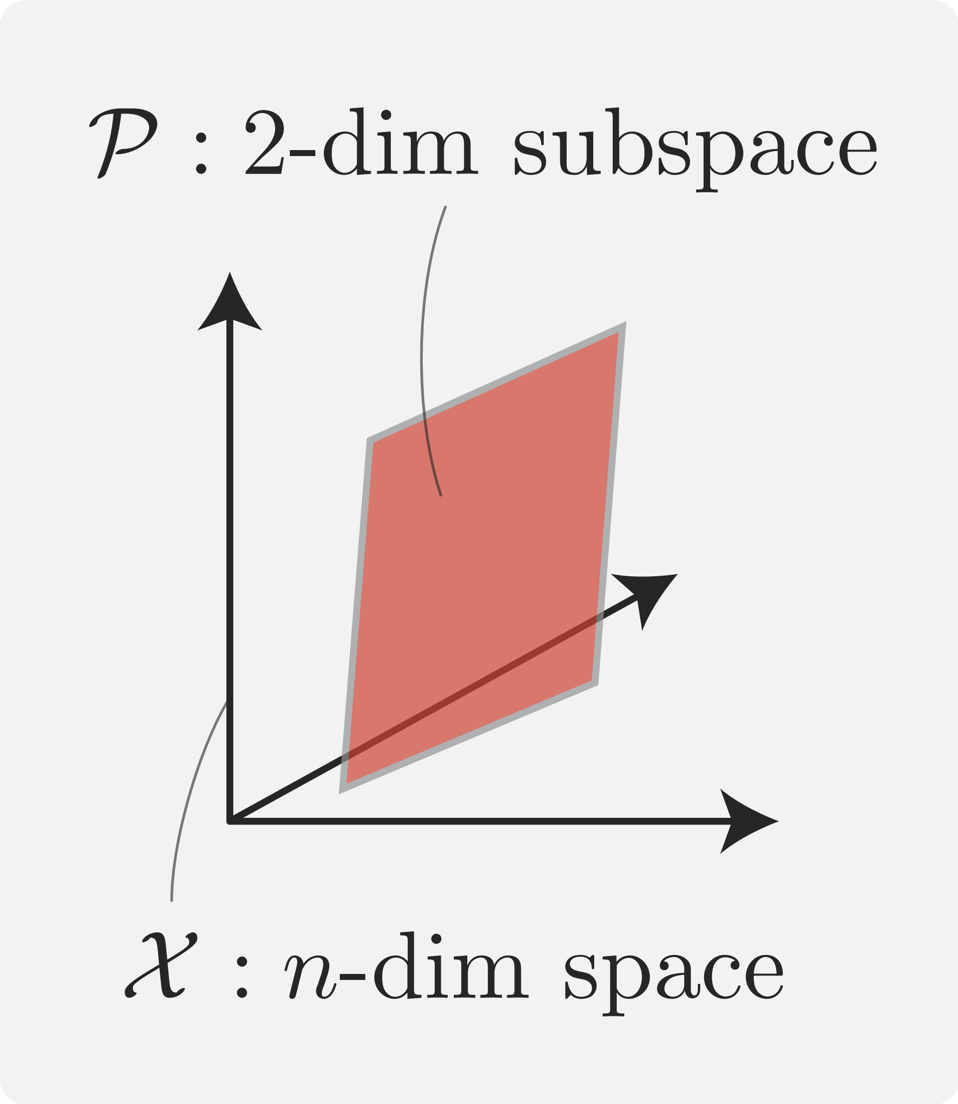
Sequential line search [Koyama+17]

Baseline 2:
SPS (Random)



Sequential plane search, but the plane is randomly chosen (instead of using BO)

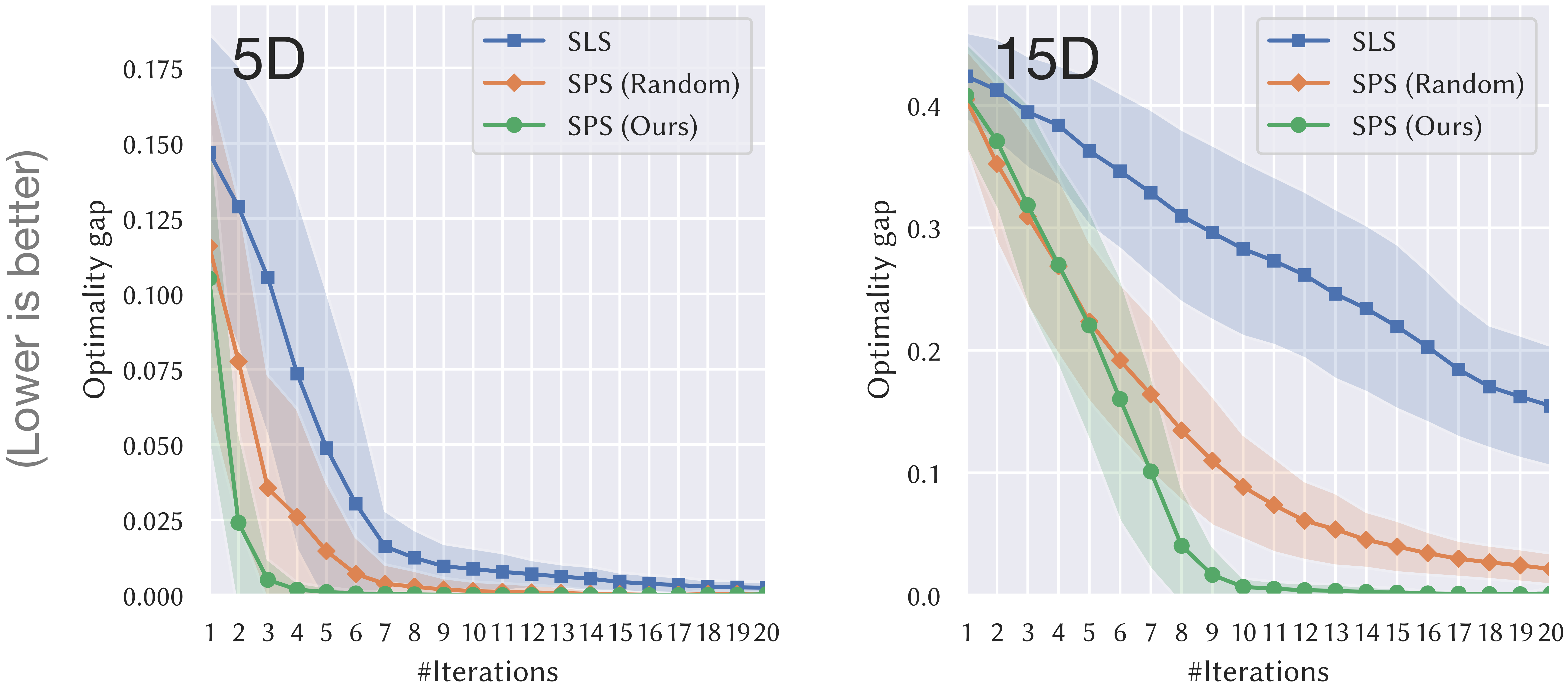
SPS (Ours)



Sequential plane search (using BO)

Result: Performance Comparison [1/2]

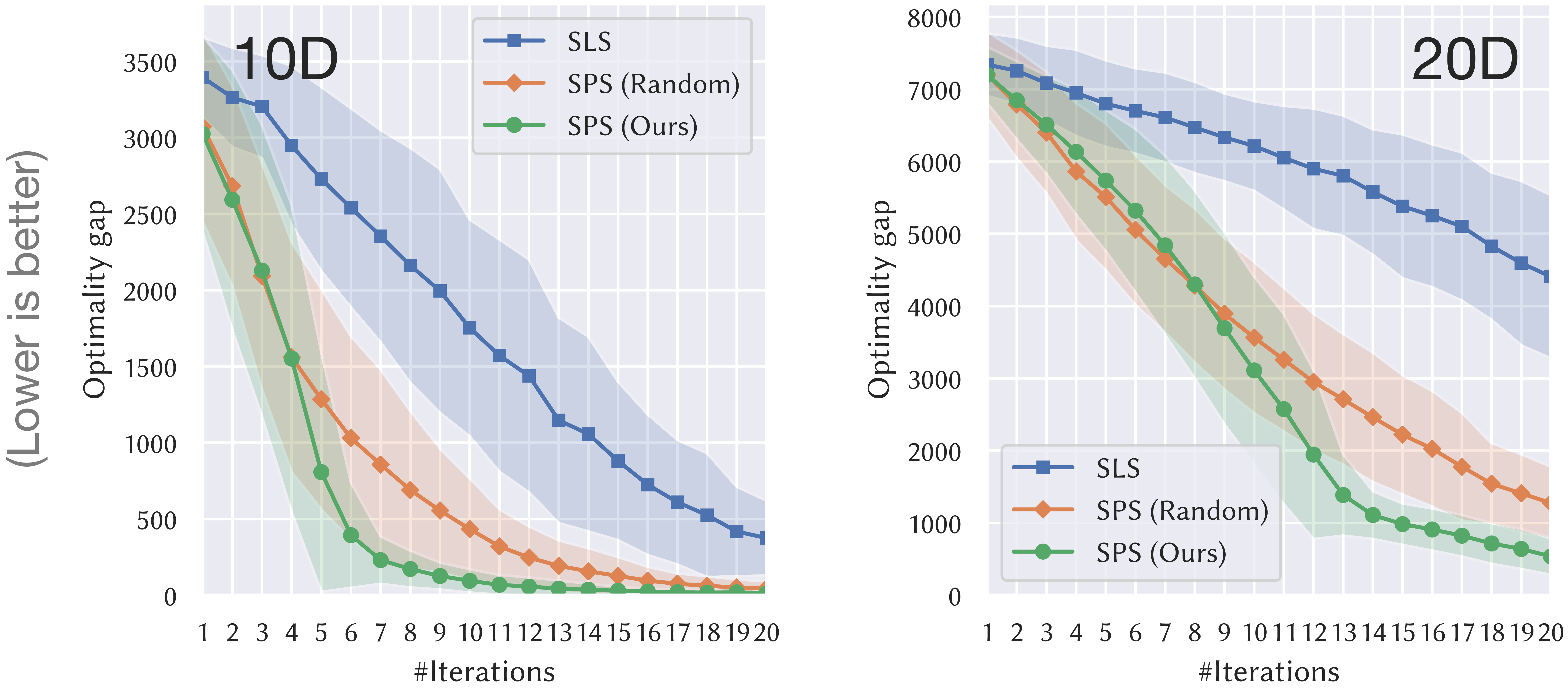
Synthetic objective function: Gaussian function



Performance: SLS < SPS (random) < SPS (ours)

Result: Performance Comparison [2/2]

Synthetic objective function: Rosenbrock function



Performance: SLS < SPS (random) < SPS (ours)

Evaluation [2/2]:
Informal User Study

Informal User Study

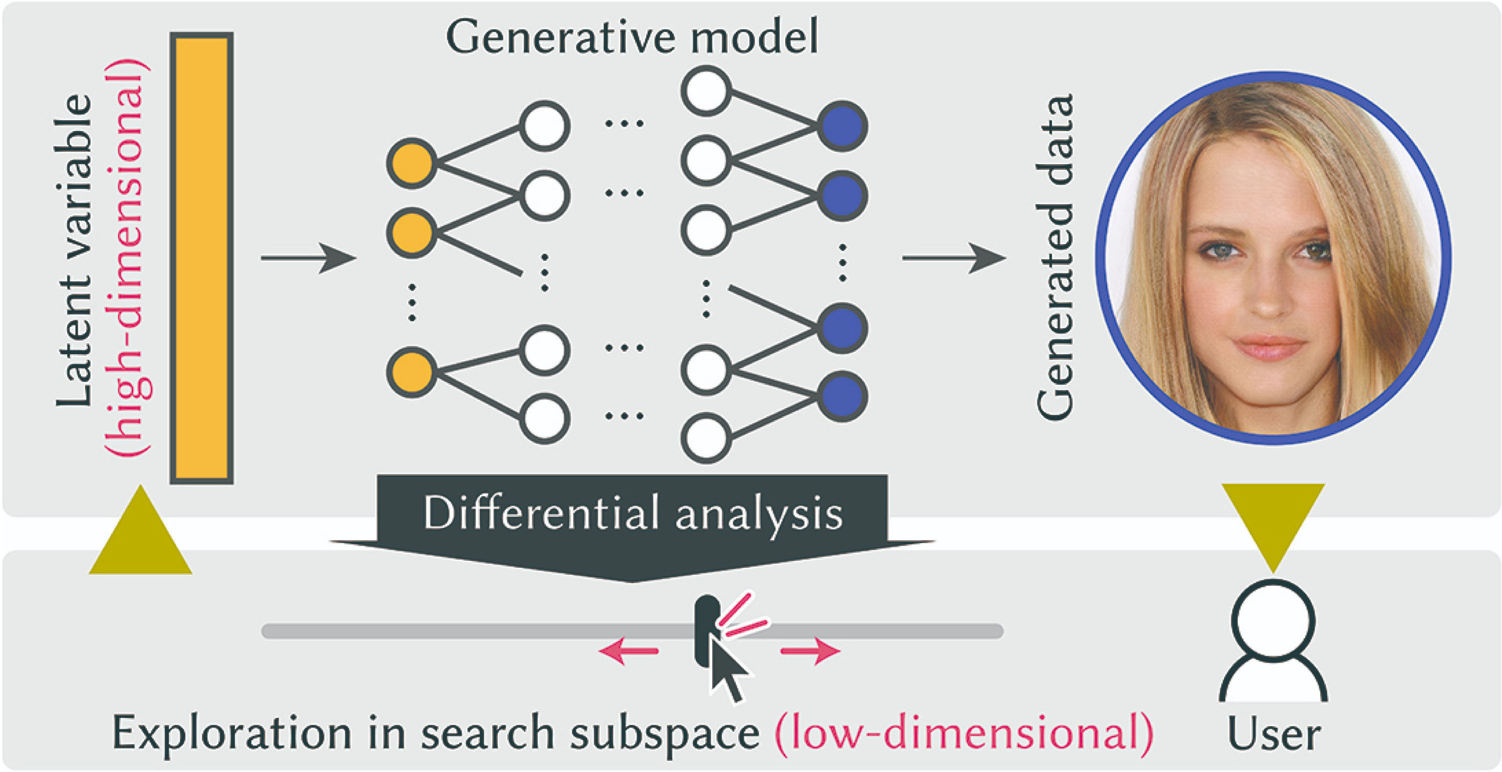
- **Participants**: Six students/researchers (five novices and one expert)
- **Task**: Enhance photographs using our system (12D)
- **Results**:
    - All the participants **could successfully perform optimization** and find satisfactory results in **5.36 iterations** in average (SD = 2.69)
    - Participants appreciated the grid view since they **could get inspiration** for possible color variations easily and quickly

Discussions

Limitations and Future Work

Discussions: High Dimensionality

- **BO (and thus PBO, too) is inherently not good at handling very high-dimensional problems** [Wang+16]
  - We assumed the dimensionality is **at most 20**
- To overcome the dimensionality issue, **application-specific extensions** should be made
  - E.g., for generative modeling (e.g., GANs), [Chiu+, SIGGRAPH 2020] is a possible option to combine with Sequential Gallery



C.f., A method specific to generative modeling [Chiu+, SIGGRAPH 2020]

Discussion: Others

- **Initial plane selection**: we randomly choose initial planes, but other strategies are possible (e.g., use Design Gallery [Marks+97] for the first iteration and then start the sequential plane search iterations)
- **Time-changing preference**: we assume that the user's preference does not change during the iteration, but this is not true in some cases; supporting time-changing preference is an interesting future work
- **Grid resolution and zooming levels**: currently, we manually select the grid resolution and the number of levels, but this may be adjusted dynamically by analyzing the just-noticeable difference (JND)
- **Prior knowledge**: when some prior data is available, we could build a rough approximation of the goodness function and then use it as a prior of the Bayesian inference; this would make the optimization even more efficient


Sequential Gallery for Interactive Visual Design Optimization

YUKI KOYAMA, National Institute of Advanced Industrial Science and Technology (AIST), Japan
ISSEI SATO, The University of Tokyo, Japan
MASATAKA GOTO, National Institute of Advanced Industrial Science and Technology (AIST), Japan

Refer to the paper for details

Summary

Summary

- **Sequential Gallery** is an interactive system for **user-in-the-loop** visual design optimization
- Its efficiency is enabled by **sequential plane search**, which is a new variant of **preferential Bayesian optimization** and is able to find the solution with only a minimal number of iterations

References

- **[Akiba+, KDD 2019]** Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proc. KDD '19. 2623–2631. DOI:https://doi.org/10.1145/3292500.3330701
- **[Brochu+, NIPS 2007]** Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. 2007. Active Preference Learning with Discrete Choice Data. In Proc. NIPS '07. 409–416. http://papers.nips.cc/paper/3219-active-preference-learning-with-discrete-choice-data.pdf
- **[Brochu+, SCA 2010]** Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian Interactive Optimization Approach to Procedural Animation Design. In Proc. SCA '10. 103–112. https://doi.org/10.2312/SCA/SCA10/103-112
- **[Chiu+, SIGGRAPH 2020]** Chia-Hsing Chiu, Yuki Koyama, Yu-Chi Lai, Takeo Igarashi, and Yonghao Yue. 2020. Human-in-the-Loop Differential Subspace Search in High-Dimensional Latent Space. ACM Trans. Graph. 39, 4, pp.85:1–85:15 (July 2020). DOI:https://doi.org/10.1145/3386569.3392409
- **[Koyama+, SIGGRAPH 2017]** Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential line search for efficient visual design optimization by crowds. ACM Trans. Graph. 36, 4, Article 48 (July 2017), 11 pages. DOI:https://doi.org/10.1145/3072959.3073598
- **[Koyama+, Computational Interaction (2018)]** Yuki Koyama and Takeo Igarashi. 2018. Computational Design with Crowds. In Computational Interaction (Eds. Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes), Oxford University Press, pp.153—184. https://arxiv.org/abs/2002.08657
- **[Shahriari+, Proc. IEEE (2016)]** Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proc. IEEE 104,1(January 2016),148–175. https://doi.org/10.1109/JPROC.2015.2494218
- **[Wang+, JAIR (2016)]** Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. 2016. Bayesian Optimization in a Billion Dimensions via Random Embeddings. J. Artif. Intell. Res. 55 (February 2016), 361–387. https://doi.org/10.1613/jair.4806

Sequential Gallery for Interactive Visual Design Optimization

Yuki Koyama[1], Issei Sato[2], and Masataka Goto[1]

1. National Institute of Advanced Industrial Science and Technology (AIST)
2. The University of Tokyo

SIGGRAPH THINK BEYOND
2020 19-23 JULY WASHINGTON DC