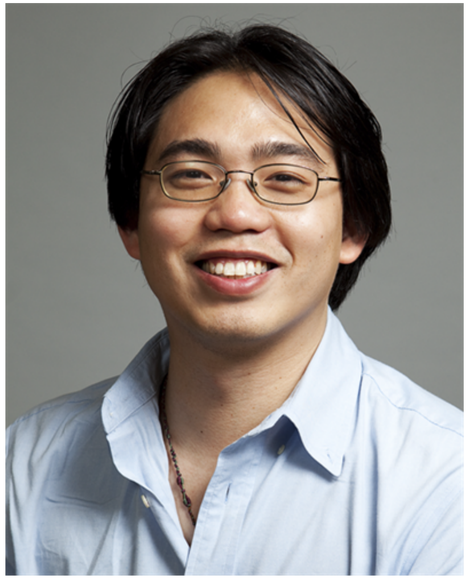
AlloyLens

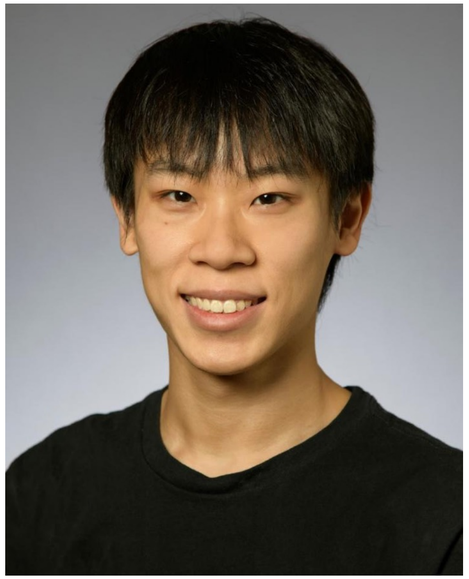
A visual analytics tool for high-throughput alloy screening and inverse design

Suyang (Susie) Li, Fernando Fajardo-Rojas, Diego Gomez-Gualdron, Remco Chang, Mingwei Li

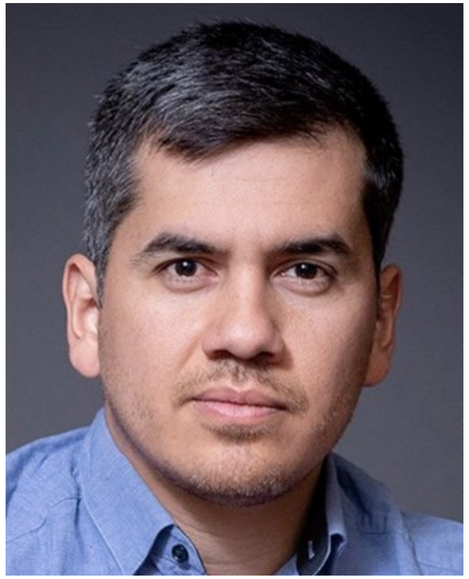
Collaborators



Advisor: Remco Chang
Professor, Computer Science, Tufts University



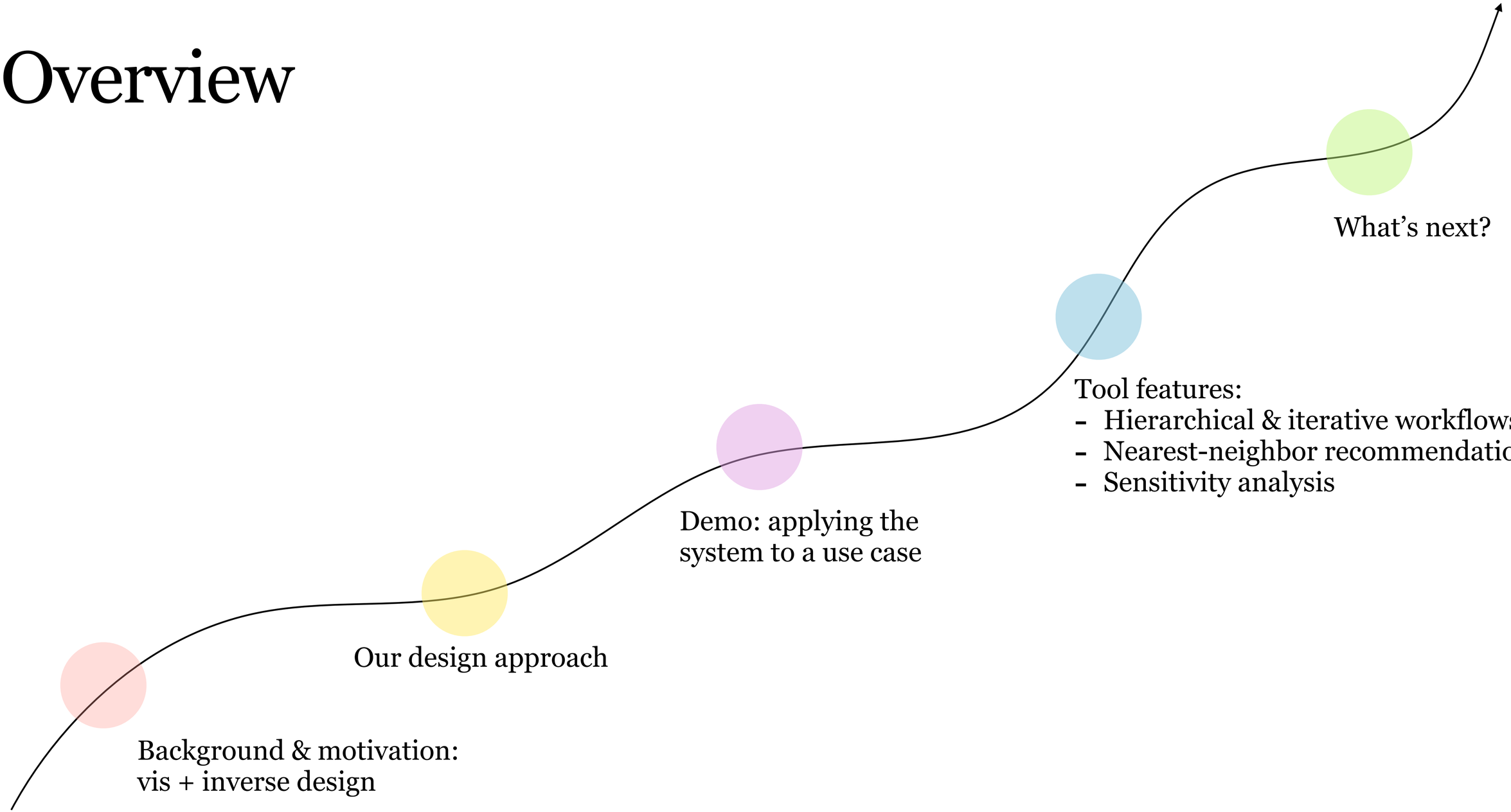
Mingwei Li
Postdoctoral Researcher, Computer Science, Tufts University



Diego Gomez-Gualdron
Associate, Professor of Chemical & Biological Engineering, Colorado School of Mines

Fernando Fajardo-Rojas
Colorado School of Mines

Overview


What's next?
Tool features:
- Hierarchical & iterative workflows
- Nearest-neighbor recommendation
- Sensitivity analysis
Demo: applying the system to a use case
Our design approach
Background & motivation: vis + inverse design

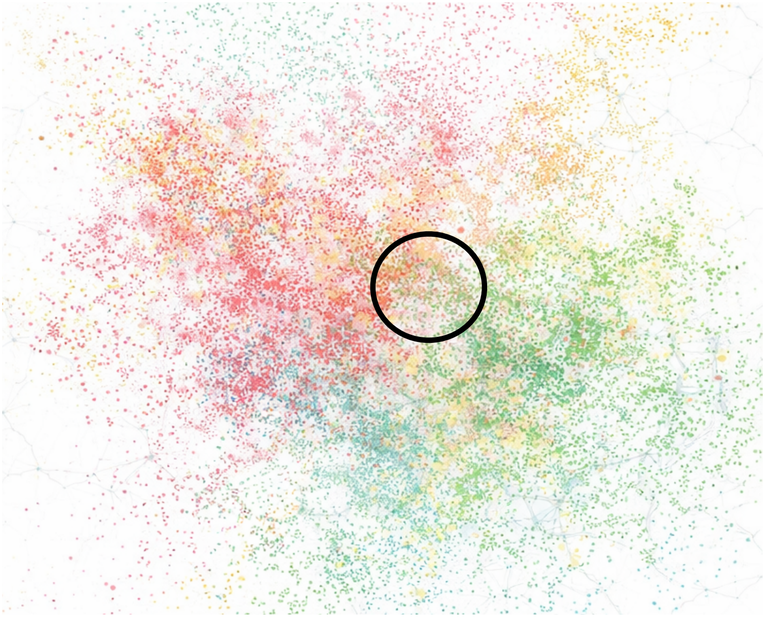
Background + Motivation

How do we go from high dimensional alloy data...

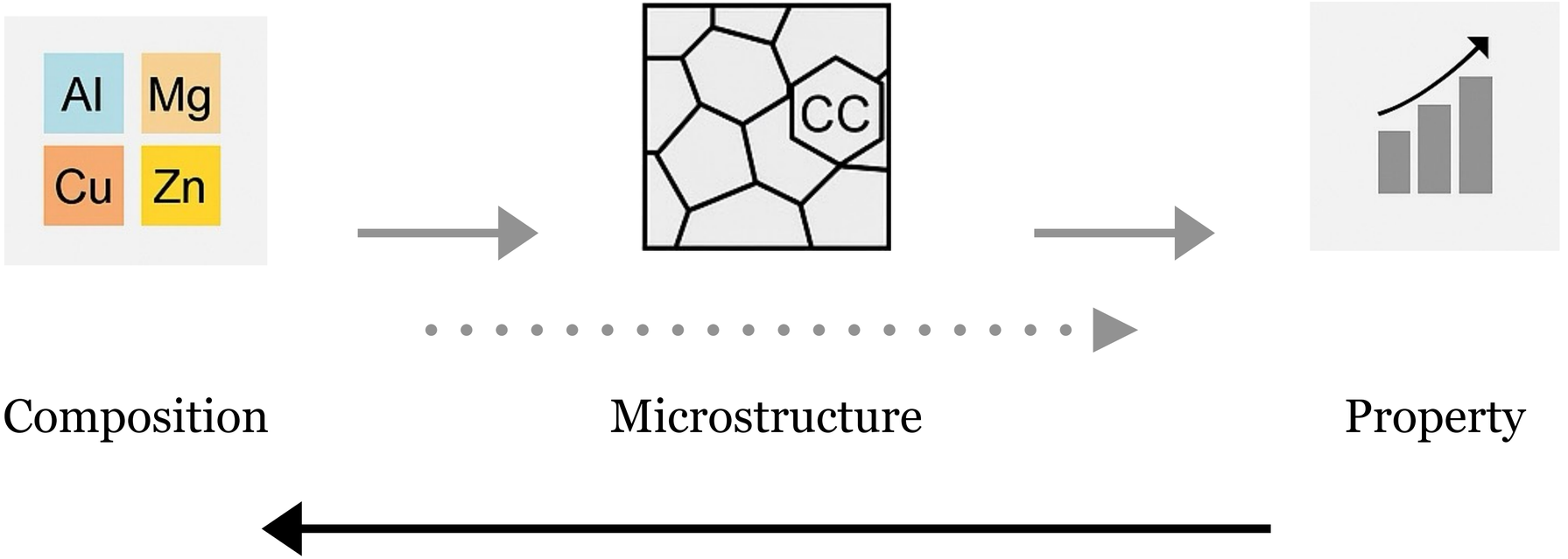
rapid_alloy_dev.txt

```
6082[%] 2024[%] bat-box[%]      3003[%] 4032[%] Al      Si      Cu      Ni      Mg      Mn      Fe      Cr
Vf_FCC_A1       Vf_DIAMOND_A4   Vf_AL15SI2M4    Vf_AL3X Vf_AL6MN        Vf_MG2ZN3       Vf_AL3NI2       Vf_AL3NI
Vf_AL2CU_C16    Vf_Q_ALCUMGSI   Vf_AL7CU2FE     Vf_MG2SI_C1     Vf_AL9FE2SI2    Vf_AL18FE2MG7SI10       eut. fra
        T_DIAMOND_A4    T_AL15SI2M4     T_AL3X  T_AL6MN T_MG2ZN3        T_AL3NI2        T_AL3NI_D011    T_AL7CU4
T_Q_ALCUMGSI    T_AL7CU2FE      T_MG2SI_C1      T_AL9FE2SI2     T_AL18FE2MG7SI10        T(liqu) T(sol)  delta_T
14      delta_T_Si      CSC     YS(MPa) hardness(Vickers)       CTEvol(1/K)(20.0-300.0∞C)       Density(g/cm3)
S/m)    El. resistivity(ohm m)  heat capacity(J/(mol K))        Therm.conductivity(W/(mK))      Therm. diffusivi
y(mK/W) Linear thermal expansion (1/K)(20.0-300.0∞C)    Technical thermal expansion (1/K)(20.0-300.0∞C)


0       0       0       100     83.675  12.25   0.9     1.3     1.05    0.025   0.55    0.05    0.025   0.025
10.29596        0.429620000000001       0.00198 0.57619 nan     0.471840000000001       2.46493 0.43028 0.01502
0.66099 85.01013        566.03237       566.87405       570.99893       661.52969       nan     195.57346
563.89033       323.21794       172.49277       403.36182       406.29139       497.92652       562.72011
542.7104        118.6967        24.16365        118.81929       28.2885299999999        0.4784382       384.672
7.11539E-5      2.65803 1.03585E-5      11302200        8.85145E-8      27.3373 159.046 6.01834E-5      0.00628
2.16178E-5


0       0       3.3     96.7    84.11885        11.86555        0.874425        1.2571  1.017   0.065425
0.025825        0.025825        0.122525        81.89618        9.41901 0.819639999999999       0.00199 0.57495
0.01388 0.16447 0.00107 0.79087 1.49454 0.61897 90.10249        567.63163       566.75619       568.61731
8       nan     542.79063       563.92944       326.69331       172.49434       403.08996       407.3476
549.00451       661.5059        542.6677        118.8382        25.20635        119.03046       25.94961
117.33693       7.13183E-5      2.65879 1.03464E-5      11414800        8.76735E-8      27.343  160.429 6.06269
2.37931E-5      2.16366E-5


0       0       6.6     93.4    84.5627 11.4811 0.84885 1.2142  0.984   0.10585 0.5599  0.0467  0.02335 0.02665
9.40406 0.875829999999999       0.00159 0.45817 nan     0.44843 2.26107 0.39524 0.00339000000000003     0.16319
90.15615        567.8873        568.04601       566.7339        661.49434       nan     239.48059       nan
328.40257       172.49771       402.8104        384.07284       466.85872       562.77306       548.75849
        26.23535        118.85564       24.0952 0.4806531       381.73599       116.80082       7.16862E-5
11489900        8.70807E-8      27.3633 161.346 6.07319E-5      0.00620277      2.3913E-5       2.17379E-5
```

To a smaller subset of ideal candidates with target properties?



Inverse design



How can visualization support *inverse design*—going from desired properties to a set of viable alloy compositions?

Outline

Background + motivation

Our design approach

Demo: applying the system to a use case

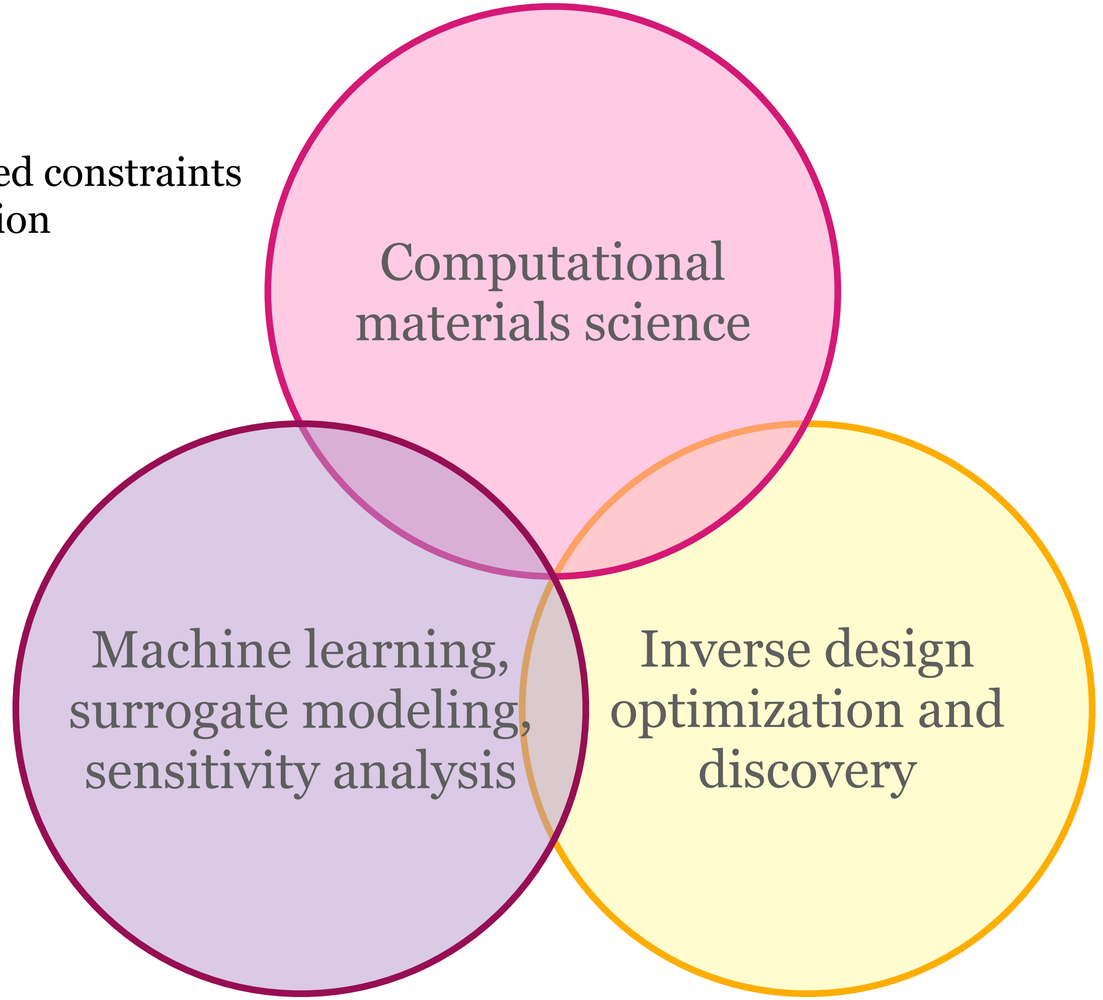
System features:
- Hierarchical & iterative workflows
- Optimal materials screening
- Nearest-neighbor recommendation

What's next?

High-Throughput Alloy Design (HTAD)

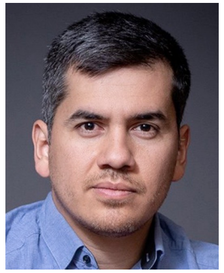
Physics-based constraints and simulation

Data-driven



Domain users, material scientists

COLORADO SCHOOL OF MINES



Dataset for AlloyLens

	KS1295[%]	6082[%]	2024[%]	bat-box[%]	3003[%]	4032[%]	Al	Si	Cu	Ni	...	Density(g/cm3)	Volume(m3/mol)	El.con
0	0.0	0.0	0.0	0.0	0.0	100.0	83.675000	12.250000	0.900000	1.300000	...	2.65803	0.00001	
1	0.0	0.0	0.0	0.0	3.3	96.7	84.118850	11.865550	0.874425	1.257100	...	2.65879	0.00001	
2	0.0	0.0	0.0	0.0	6.6	93.4	84.562700	11.481100	0.848850	1.214200	...	2.65935	0.00001	
3	0.0	0.0	0.0	0.0	9.9	90.1	85.006550	11.096650	0.823275	1.171300	...	2.66111	0.00001	
4	0.0	0.0	0.0	0.0	13.2	86.8	85.450400	10.712200	0.797700	1.128400	...	2.66318	0.00001	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
324627	95.7	0.0	0.0	0.0	3.3	1.0	80.533928	12.296200	3.381765	1.946140	...	2.72509	0.00001	
324628	95.7	0.0	0.0	3.3	0.0	1.0	80.539868	12.297322	3.397440	1.946140	...	2.72496	0.00001	
324629	95.7	0.0	3.3	0.0	0.0	1.0	80.521553	12.309400	3.379290	1.946965	...	2.72497	0.00001	
324630	95.7	3.3	0.0	0.0	0.0	1.0	80.358533	12.297025	3.531090	1.946965	...	2.72756	0.00001	
324631	99.0	0.0	0.0	0.0	0.0	1.0	79.966460	12.695500	3.493800	2.012800	...	2.72671	0.00001	

324632 rows × 70 columns

324632 samples, 70 properties

6 input scraps, 12 elements, 38 microstructures, 14 properties

Human–in–the–Loop Alloy Design Workflow

- Needs and gaps?

Overview of patterns in dataset

Apply constraints: filter, specify ranges, or fix target properties.

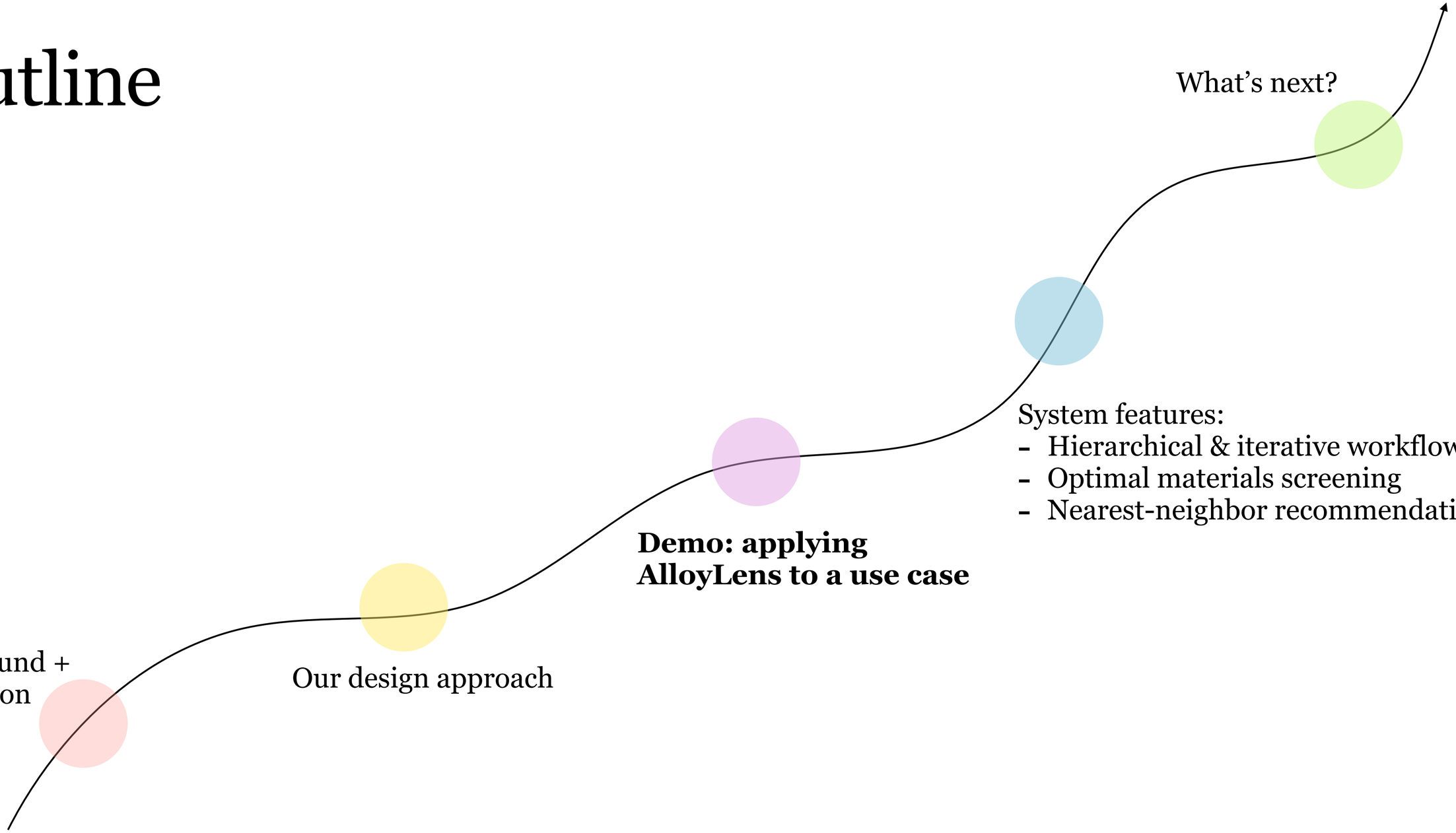
Analyze sensitivity: inspect pairwise correlations and influence of variables

Handle empty solution cases

Downstream tasks

- run more physical experiments
- collect more data from candidates of interest

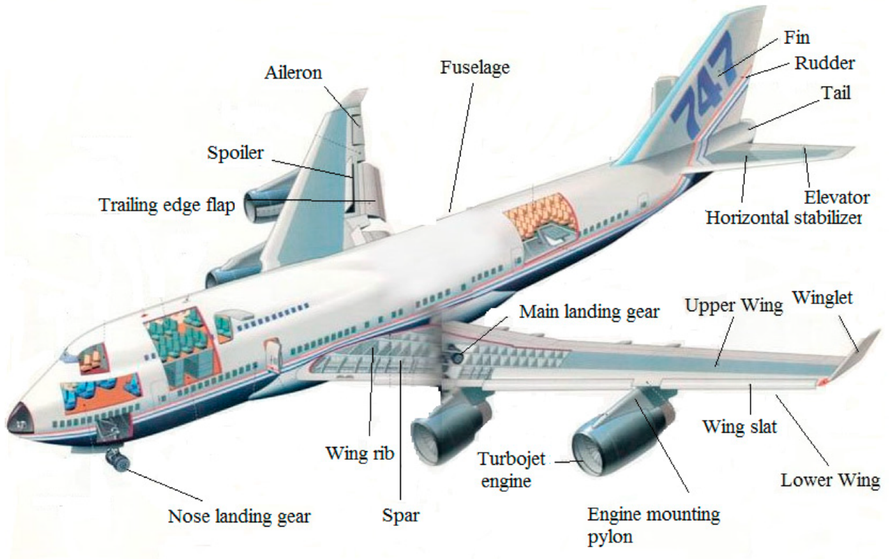
Outline


What's next?
System features:
- Hierarchical & iterative workflows
- Optimal materials screening
- Nearest-neighbor recommendation
Demo: applying AlloyLens to a use case
Our design approach
Background + motivation

Demo: Applying AlloyLens to a Real Use Case

Automotive and Aerospace systems

- Strength: 200 MPa (auto), > 300 MPa (aero)
- Hardness (Vickers): 80–130 HV
- Density (g/cm³): < 2.75
- FCC phase fraction: >80%
- CSC: < 0.5 (low risk)
- delta_T: < 100°C



Gloria, A., Montanari, R., Richetta, M., & Varone, A. (2019). Alloys for Aeronautic Applications: State of the Art and Perspectives. *Metals*, *9*(6), 662. https://doi.org/10.3390/met9060662

ui.ipynb

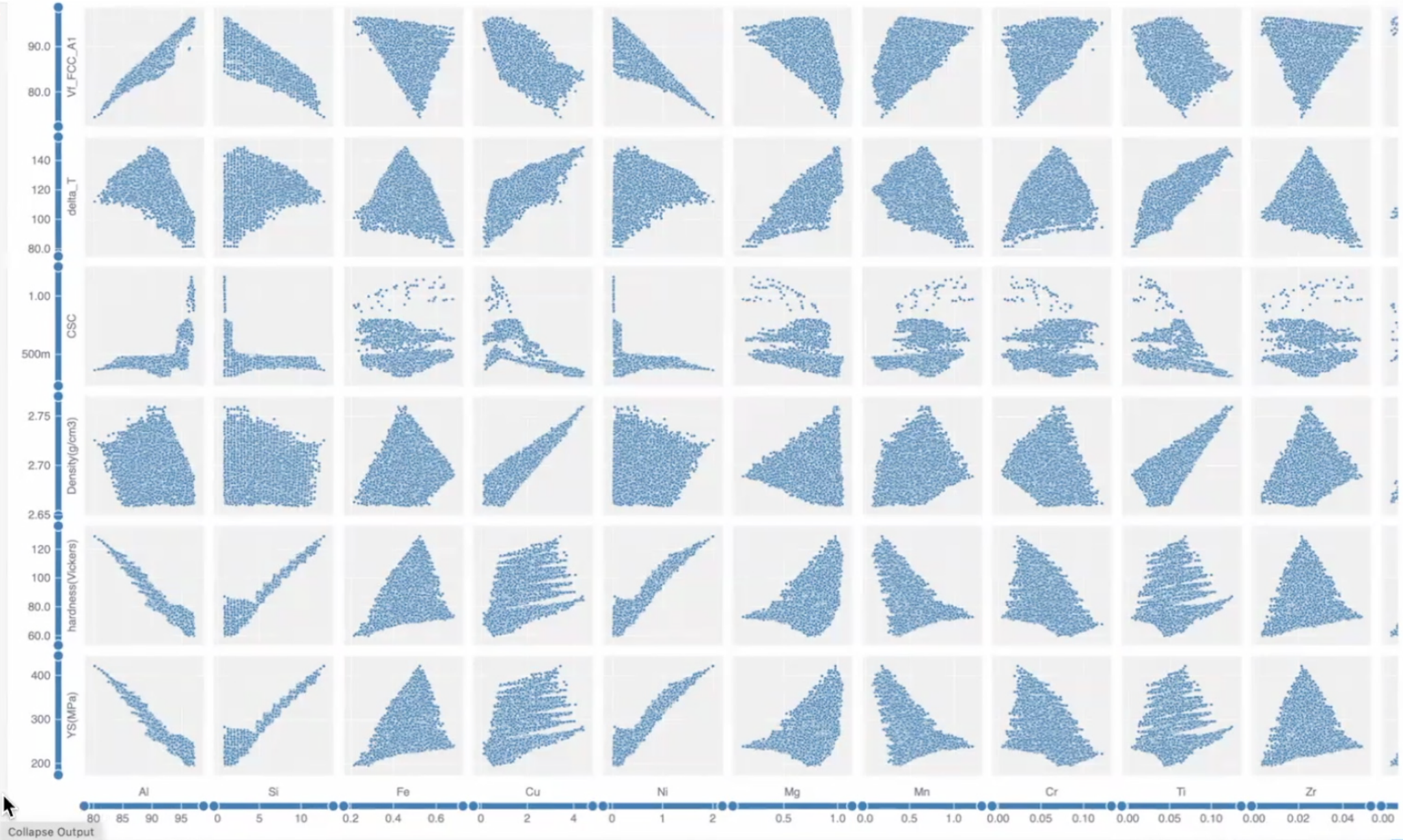
File Edit View Run Kernel Tabs Settings Help

Code

Open in... Python 3 (ipykernel)

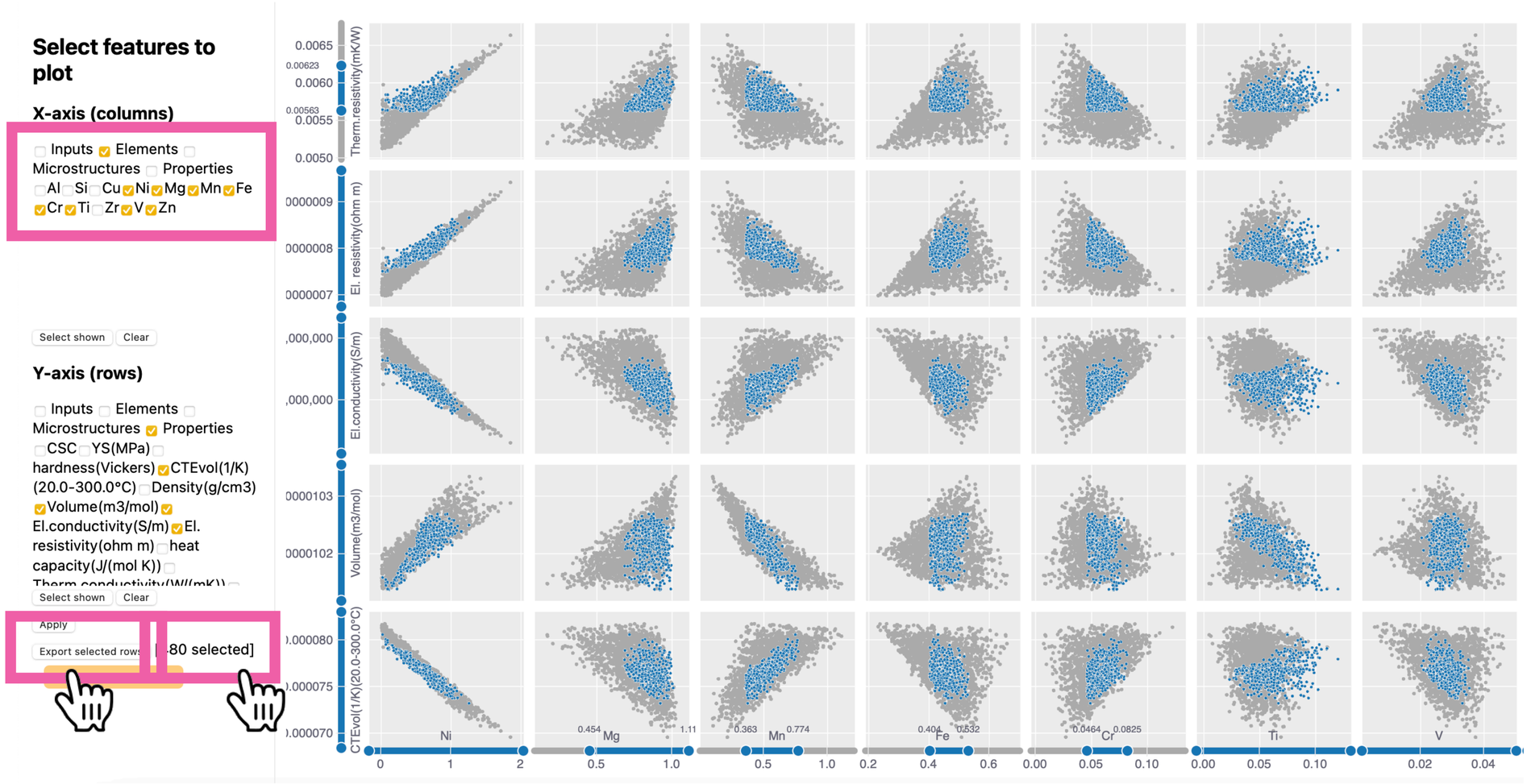
```
ui
```

[51]:



Collapse Output

AlloyLens Web



Outline

Background + motivation

Our design approach

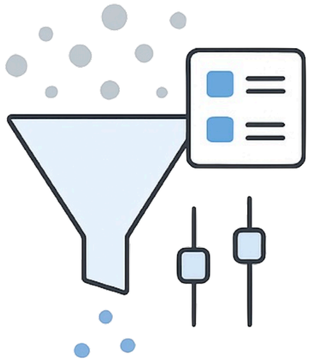
Demo: applying the system to a use case

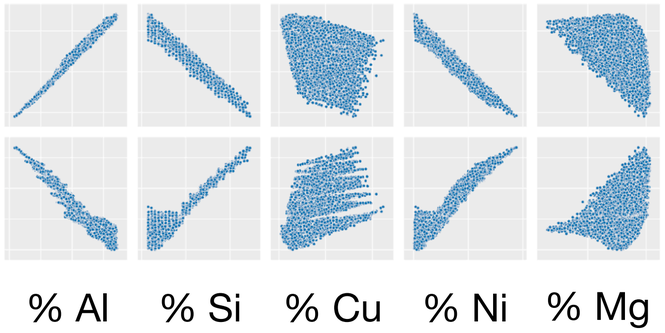
System features:
- Hierarchical & iterative workflows
- Optimal materials screening
- Nearest-neighbor recommendation

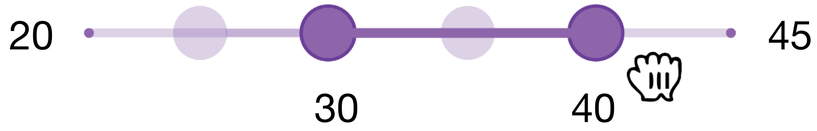
What's next?

System features

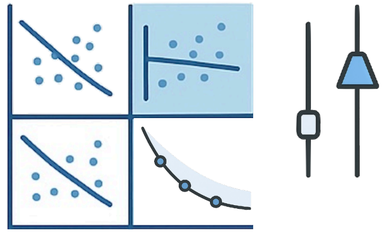
1. Overview, selection, + real-time steering



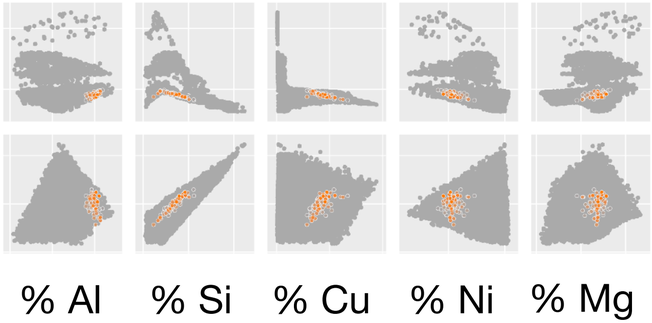




2. Sensitivity analysis

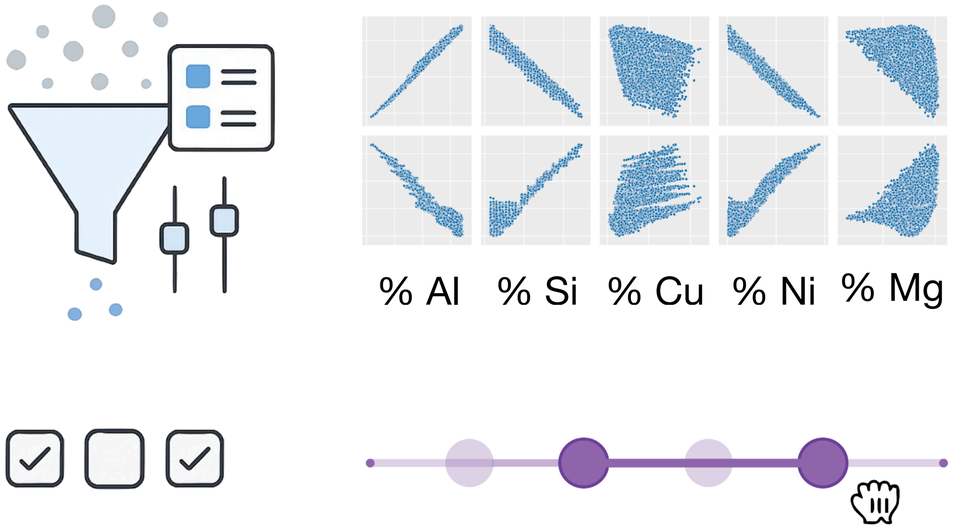


3. Nearest-neighbor recommendation



💡1: Real-time steering & adjustment

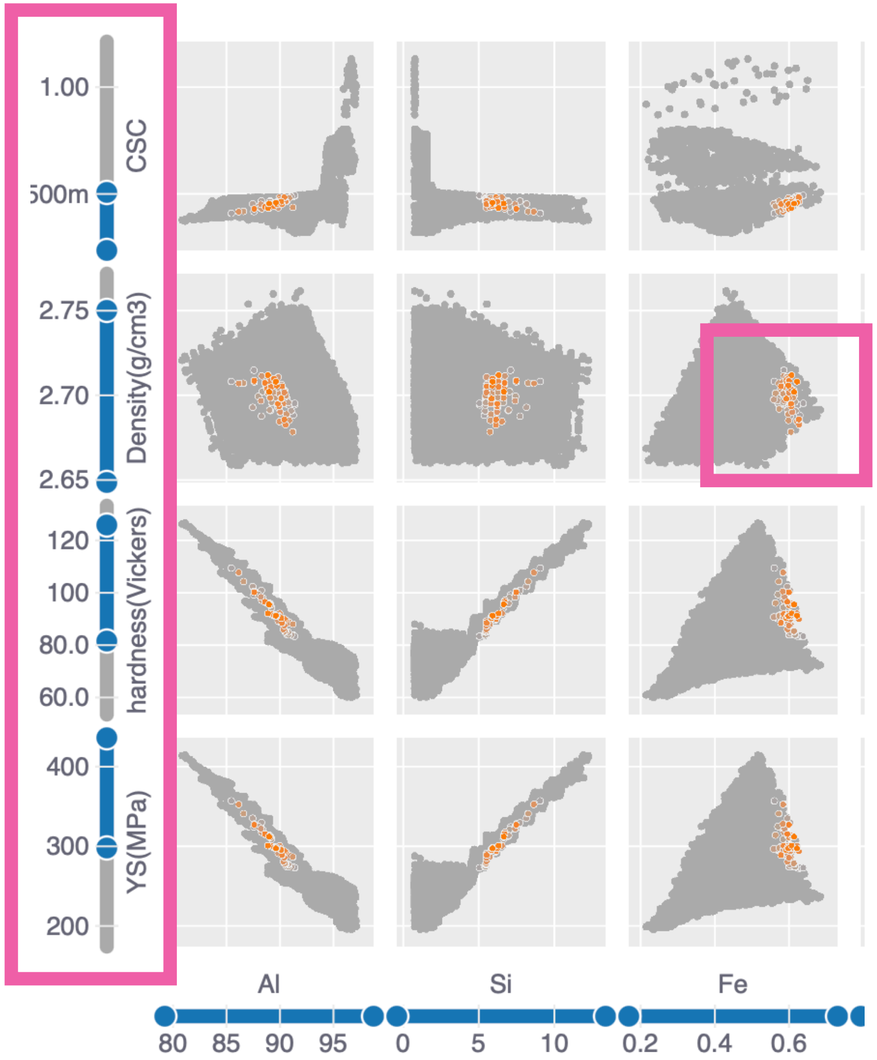
- Create a tight feedback loop between the user and the system
- Starts with an initial set of target properties
- Adjusts target specification
- But sometimes even after adjusting constraints, no perfect alloy exists — and that's where our second mechanism comes in.



💡2: Nearest neighbor: recommend next-best solution

- Case: no candidate fits all the required properties.
- AlloyLens recommends the top *k* nearest-neighbors, as measured by the Euclidean distance in feature space

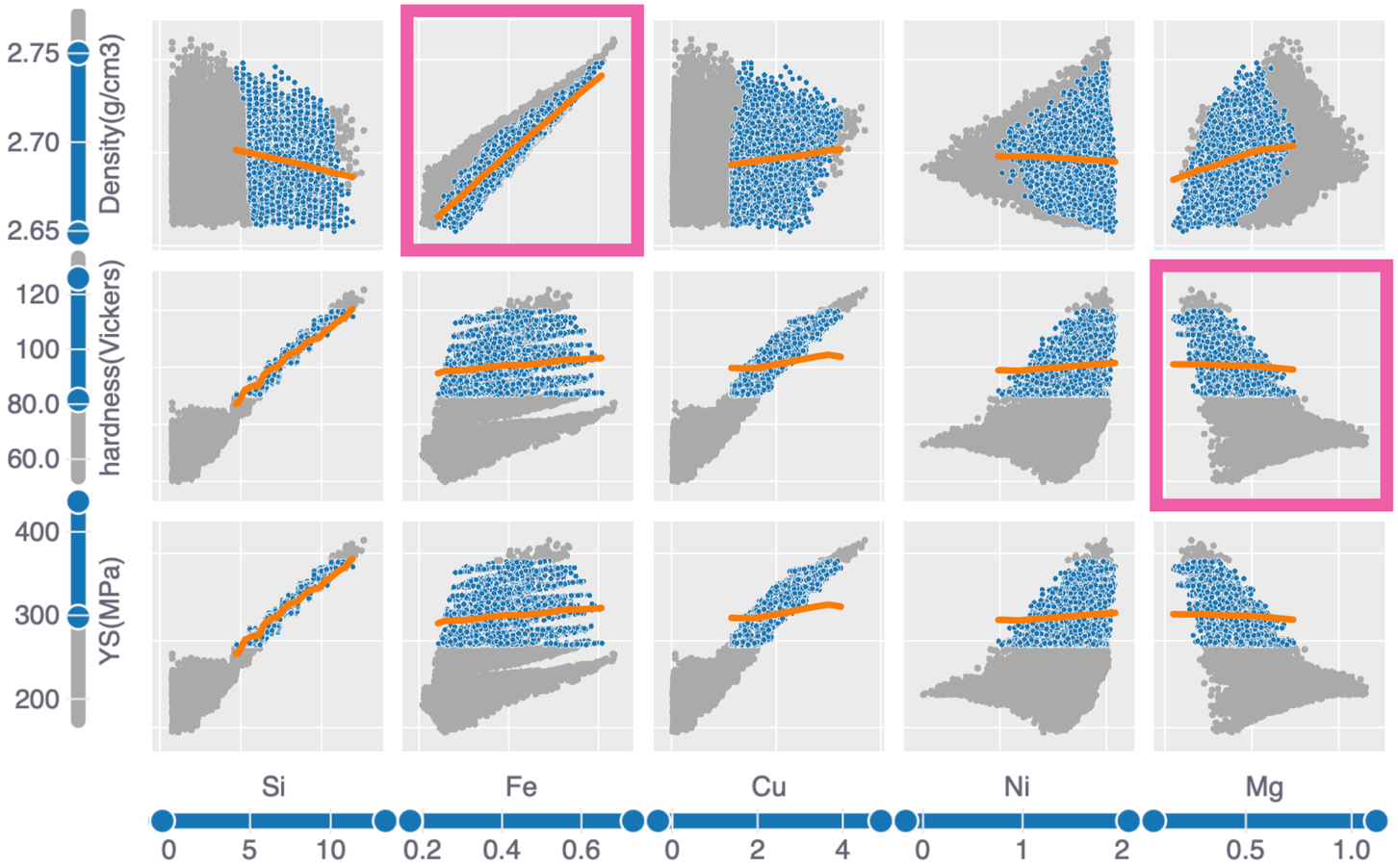
User target



top-k

💡 3: Sensitivity analysis

Steeper slope:
Changes in %Fe corresponds to a larger change in density



Flat curve:
% Mg barely affects hardness

Using AlloyLens in Jupyter

```
# Import AlloyLens
import alloylens

# Load alloy dataset (CSV/TSV/txt/etc…)
df = pd.read_csv(“SciVis_Rapid_Alloy_development.txt", sep=“\t")

# Define column groups in dataset
groups = Groups(
    elements=["Al", "Cu", "Mg", "Mn", "Si", "Zn"],
    properties=[“Yield_strength", “Therm_conductivity”],
)

# Launch interactive visualization
app = alloylens.AlloyLensApp(df, groups, x=“elements”, y=“properties”)
app # Visualize inside Jupyter notebook
```

```
groups = {"elements": element_cols, "microstructures": microstructure_cols, "properties": property_cols} # column groups

app = al.AlloyLensApp(df, groups, x="elements", y="properties") # choose feature group for x- and y- axes
app
```

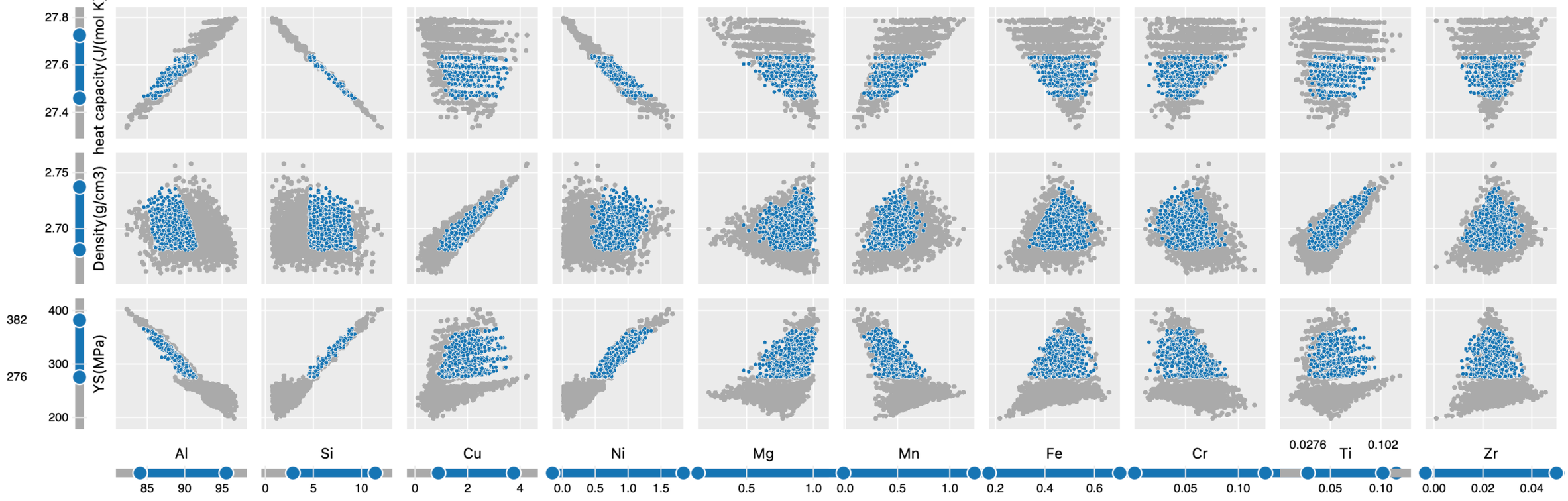
✓

Select features to plot:

☐ CSC	☑ YS(MPa)	☐ hardness(Vickers)	☐ CTEvol(1/K)(20.0-30...	☑ Density(g/cm3)	☐ Volume(m3/mol)	☐ El.conductivity(S/m)
☐ El. resistivity(ohm m)	☑ heat capacity(J/(mol ...	☐ Therm.conductivity(...	☐ Therm. diffusivity(m...	☐ Therm.resistivity(mK...	☐ Linear thermal expan...	☐ Technical thermal ex...

Select all | Clear all | Apply

You selected: ['YS(MPa)', 'Density(g/cm3)', 'heat capacity(J/(mol K))']



Sensitivity calculations

Trained a surrogate model (multi-layer perception) that predicts a property value (Y) given an elemental composition

Use the centroid of the selected points as input to the surrogate model.

Increase and decrease the X value of the centroid, plot the corresponding Y values

```python
import torch.nn as nn

# Define a simple MLP
class PropertyPredictor(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super().__init__()
        self.net = nn.Sequential(
            # nn.BatchNorm1d(input_dim),
            nn.Linear(input_dim, hidden_dim),
            nn.PReLU(),
            nn.Linear(hidden_dim, hidden_dim),
            nn.PReLU(),
            nn.Linear(hidden_dim, output_dim),
        )

    def forward(self, x):
        return self.net(x)
```

Outline

Background + motivation

Our design approach

Demo: applying the system to a use case

System features:
- Hierarchical & iterative workflows
- Optimal materials screening
- Nearest-neighbor recommendation

What's next?

What’s next

- beyond Jupyter environment: deploy as web app
- integrate surrogate predictions
- Allow UI specification for the nearest neighbor algorithm:
  - Weighting of different features (e.g. prioritize closeness in conductivity)
  - Tolerance level (delta)

Thank you!

Jupyter Widget:

github.com/susiesyli/alloylens

Source: github.com/susiesyli/alloylens

Web: susiesyli.com/alloylens-web