# Distributional Symmetry-Breaking

*or: spurious correlations meets equivariance*

Hannah Lawrence, Simons Institute Lightning Talk 8/12/25

# Birdwatching 101

Brown creeper: walks up trees

Nuthatch: walks down trees

*Example credit: Walters lab*

# Birdwatching 101

**The underlying classification function is invariant**

Brown creeper: walks up trees

Nuthatch: walks down trees

*Example credit: Walters lab*

# Birdwatching 101

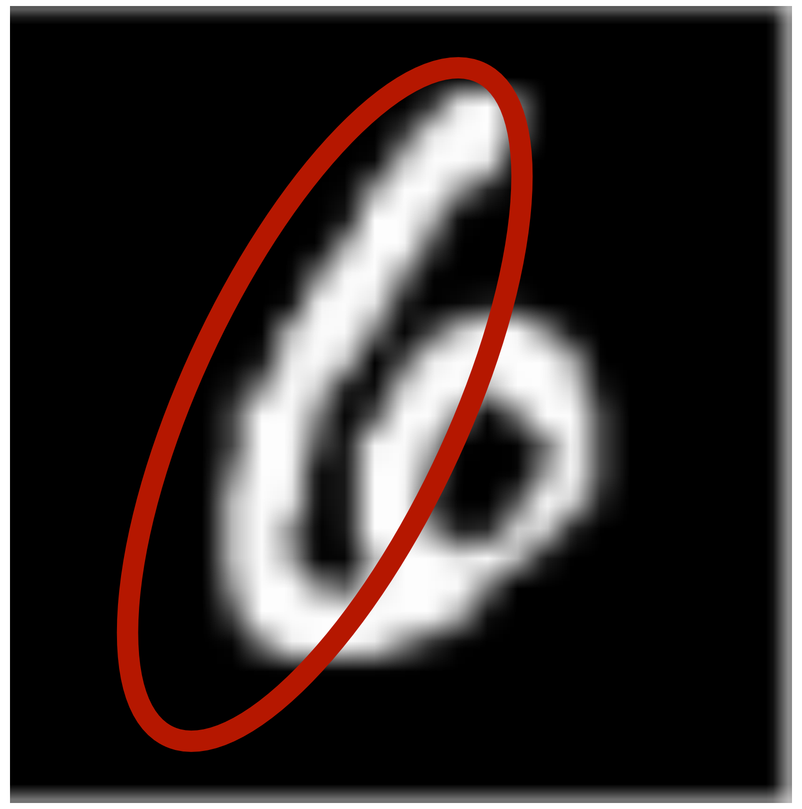
**But an equivariant model can't use this pose information!**

Brown creeper: walks up trees

Nuthatch: walks down trees

*Example credit: Walters lab*

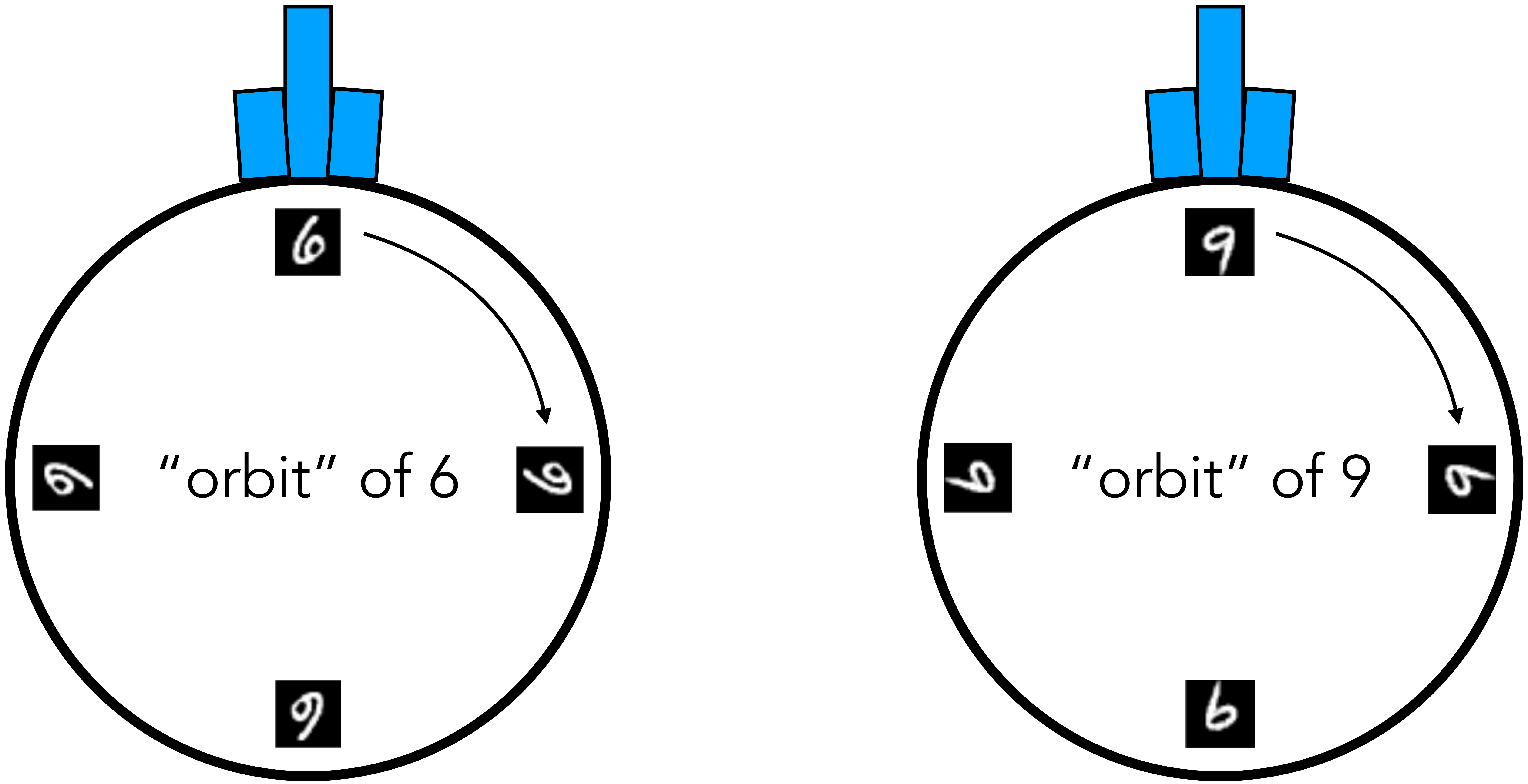
# MNIST Example



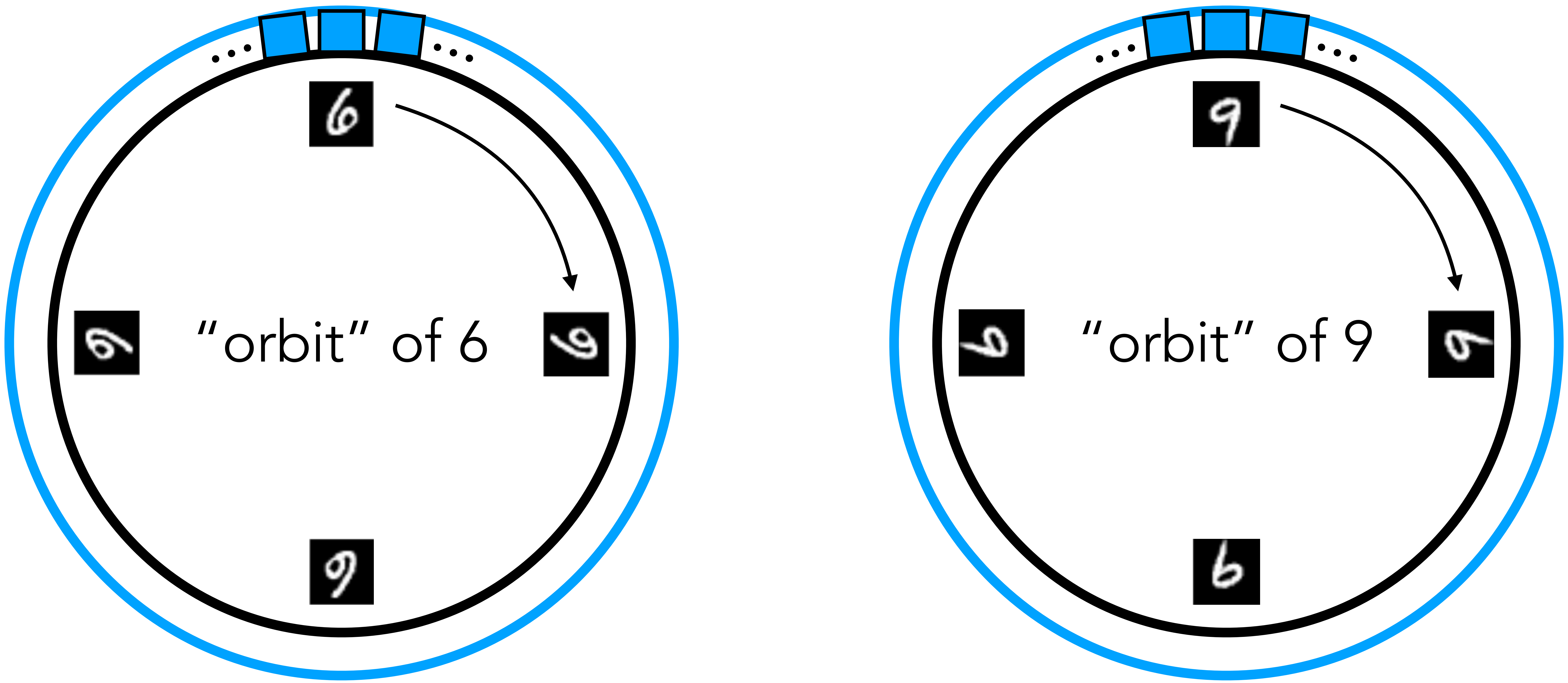
Digits have preferred orientation, which conveys useful information

…even if the actual digits are technically distinguishable

# The data distribution as orbits



# What an equivariant method sees

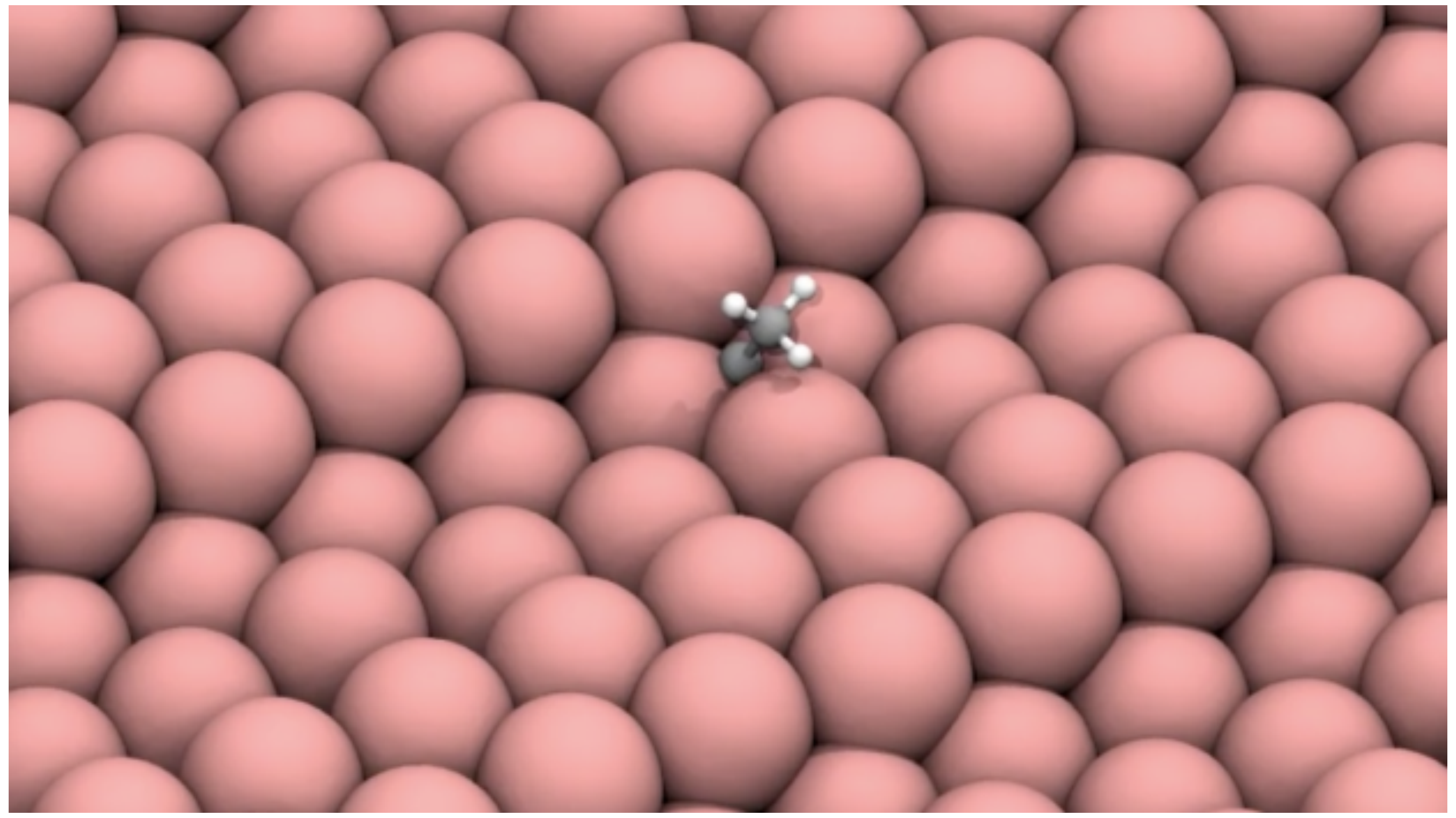



As if all mass is equally distributed on the orbit

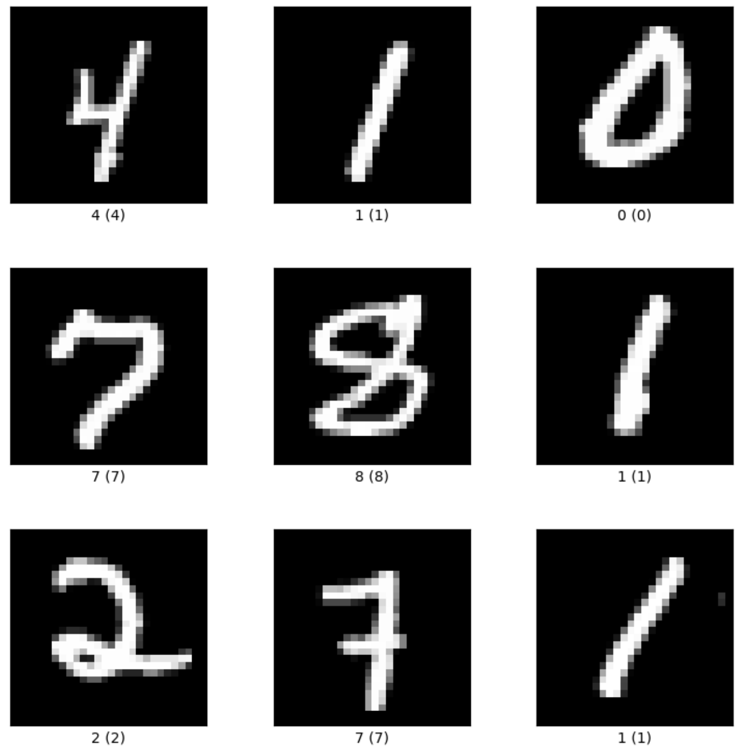
# What’s happening here?

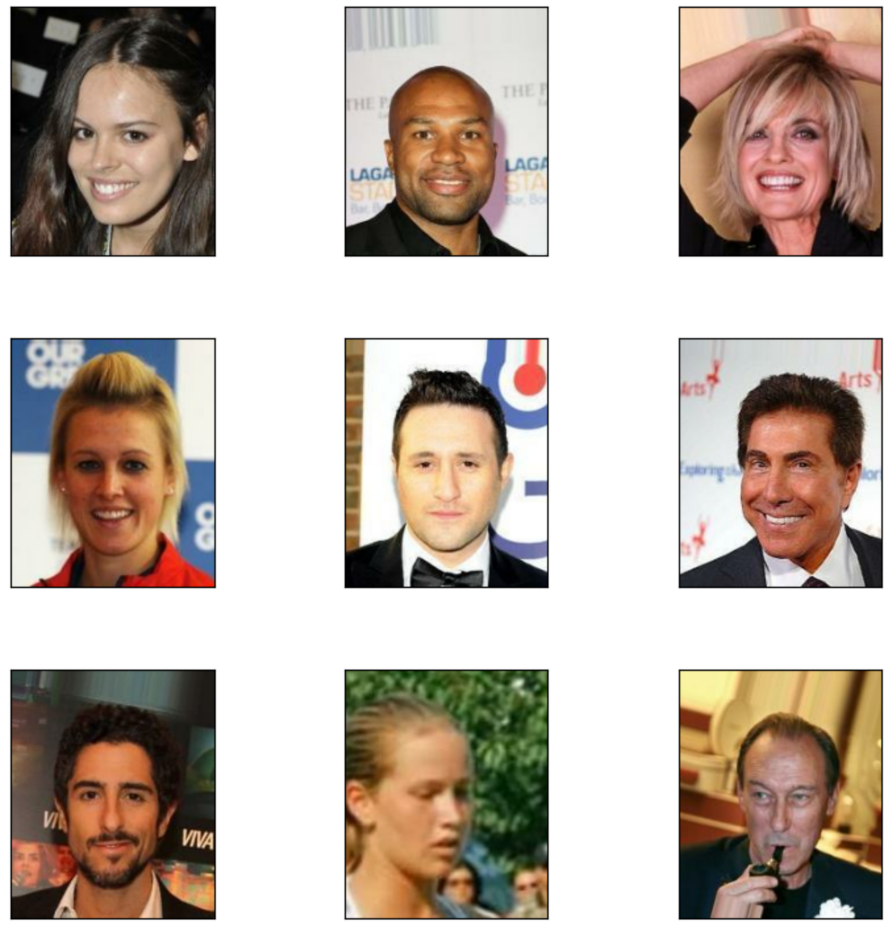
- The underlying function is rotation invariant
- However, the data distribution is NOT uniform over orbits: $x$ (bird pointed up) and $gx$ (bird pointed down) are not equally likely
  - Different species of birds — or digits — have preferred orientations, which they’re more likely to be found in
- Theoretically, the best invariant model can still tell the difference between the birds, but it is forced to use potentially **“hard”** features, like feather coloration, rather than the **“easy”** non-invariant feature of orientation

# Actually, lots of data has preferred orientations



**Superficial:** user-defined; due to convention in how the data was generated/stored





**Fundamental:** some more fundamental, replicable process in the data generation

# Why do we care?

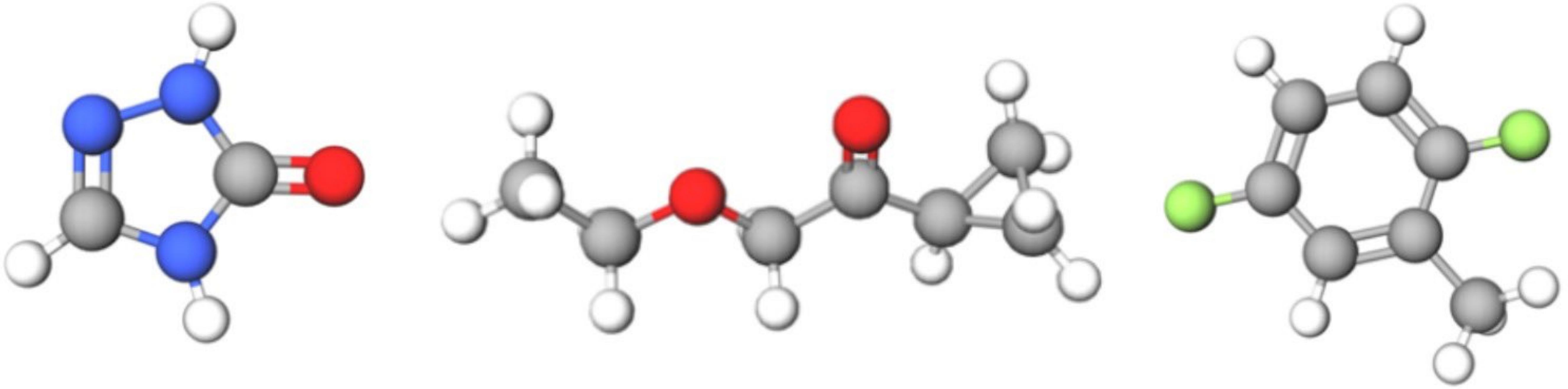
- Understanding your data is important!
- **If you only care about generalizing in-distribution…**
  - by using an invariant method, you might be throwing away useful information without realizing it
- **If you care about generalizing out of distribution, to new orientations…**
  - by using a non-invariant method, you might be led astray by overoptimistic validation performance

# Goals of this study

- **Quantify** the extent to which benchmark point cloud datasets are distributed evenly over rotations
    - Build on current metrics (e.g. kernel-based hypothesis test framework, Chiu & Bloem-Reddy 2024)
- **Empirically** test what the implications are for learning with/without augmentations

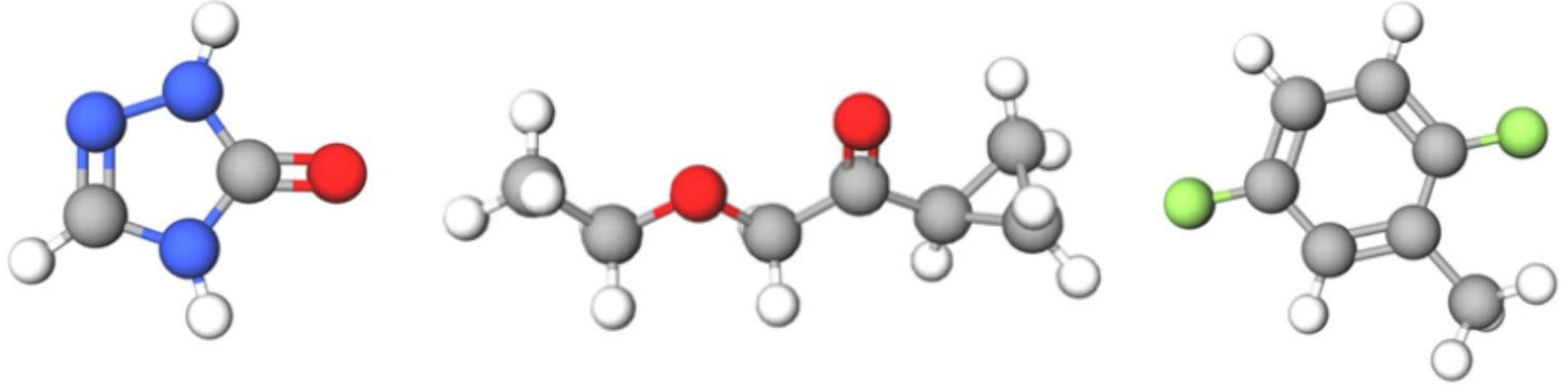
*Note: distinct from the symmetry discovery problem!*

# Proposal #1: measuring canonicalization



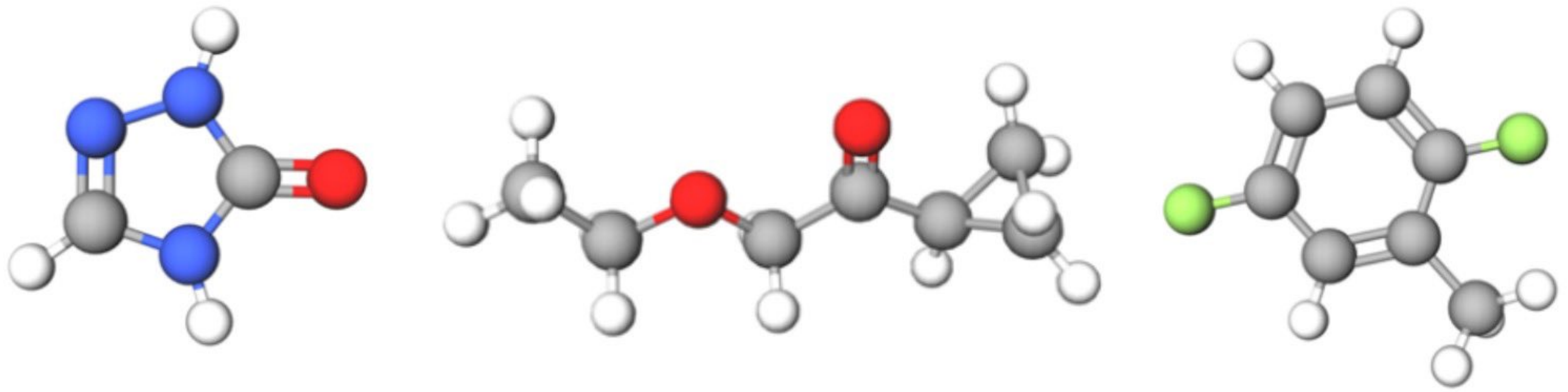
Start with your natural dataset

# Proposal #1: measuring canonicalization

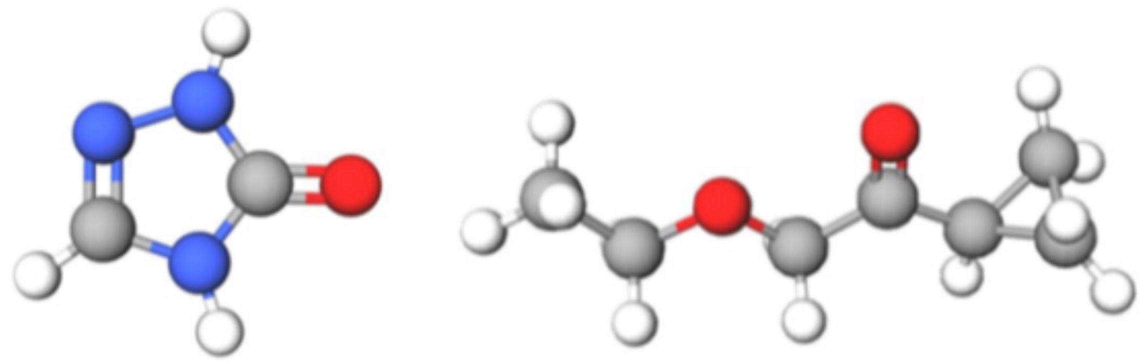


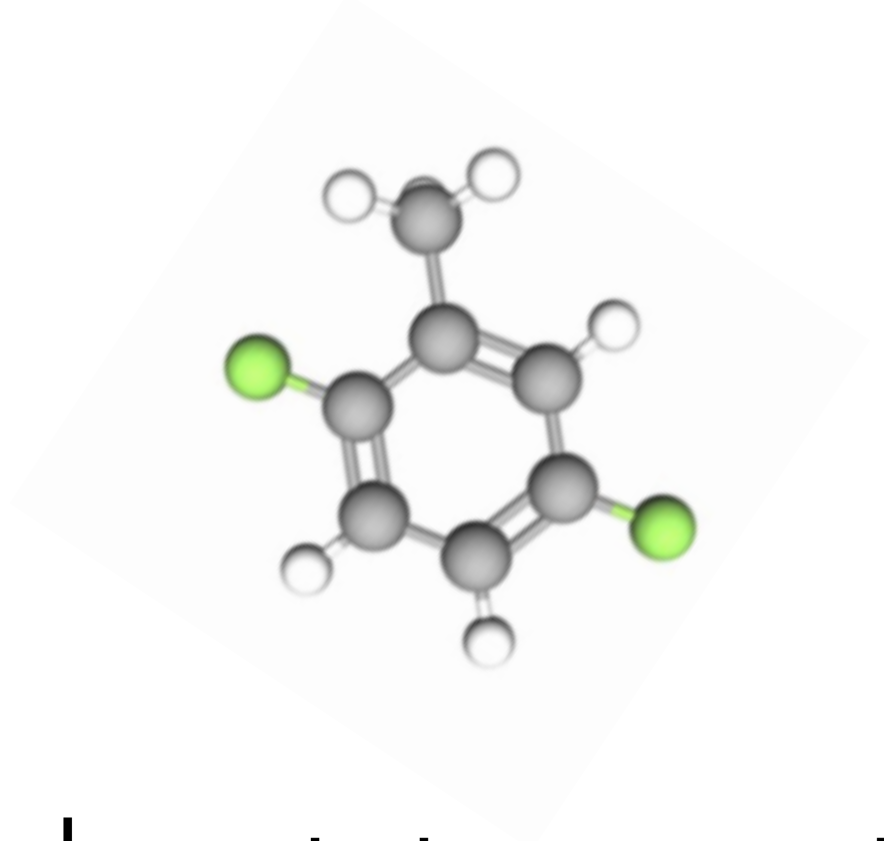
Split it into two halves

# Proposal #1: measuring canonicalization



Split it into two halves

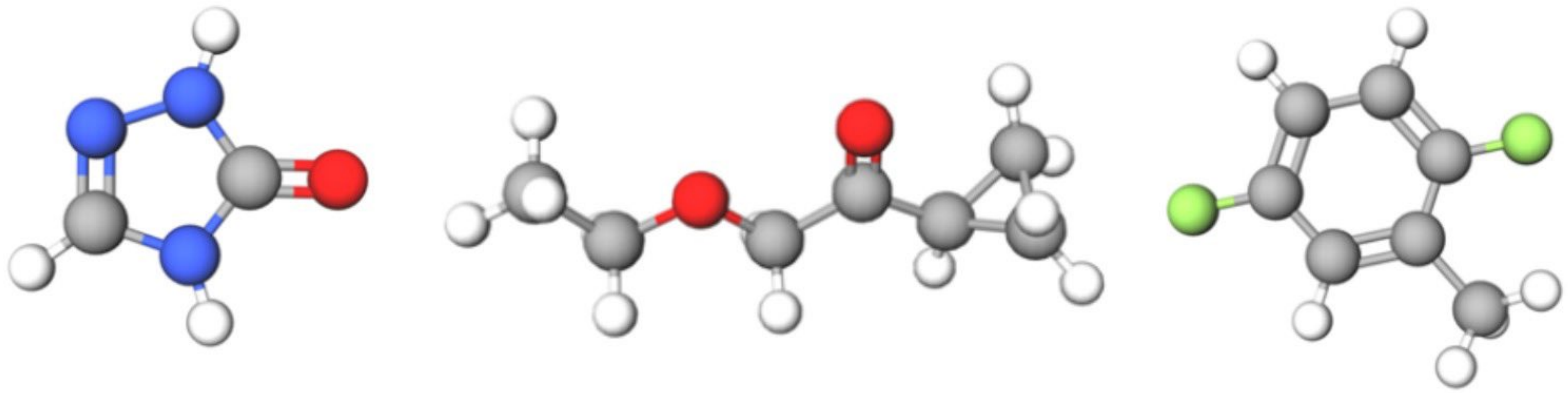




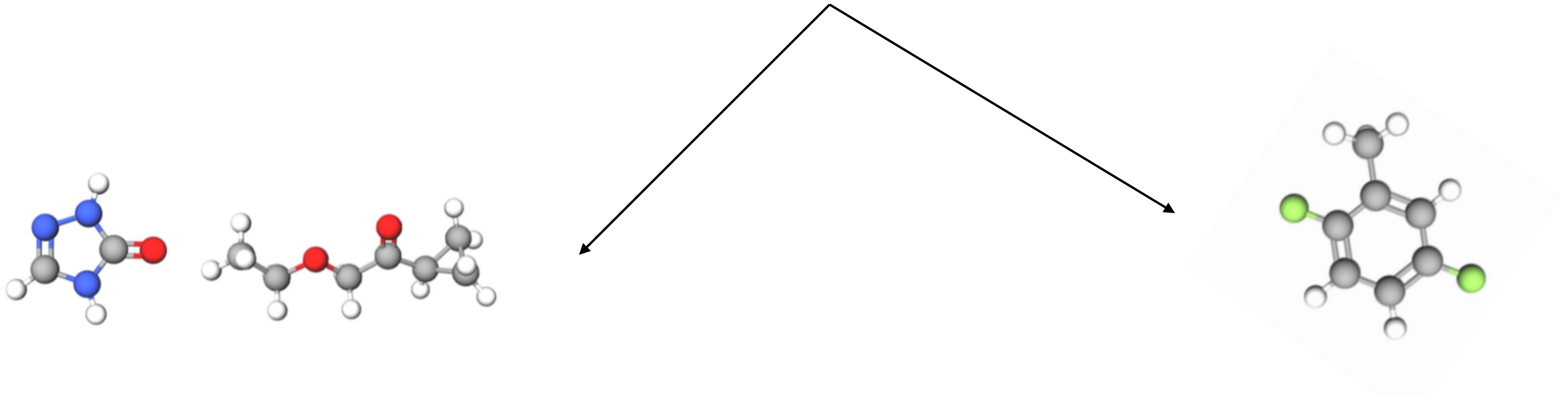
Don’t change anything

Randomly rotate everything

# Proposal #1: measuring canonicalization



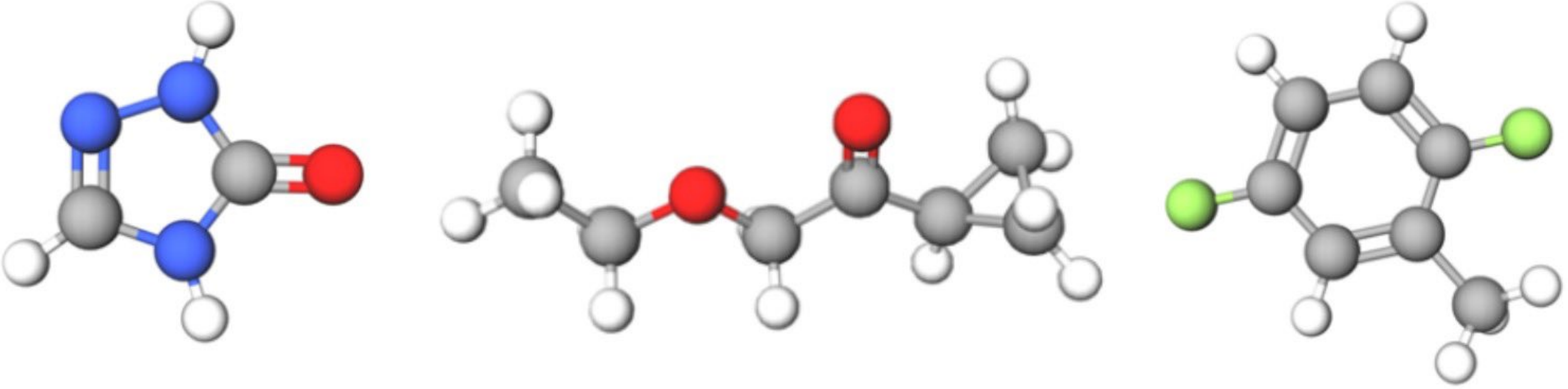
Split it into two halves



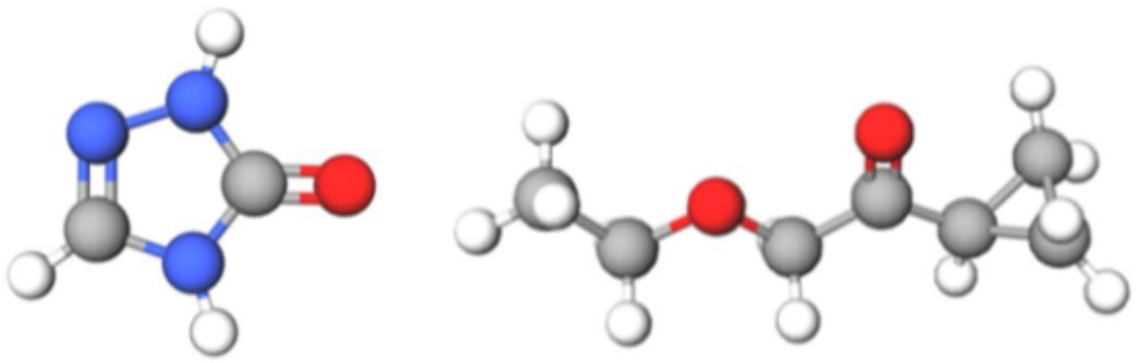
Class label 0

Class label 1

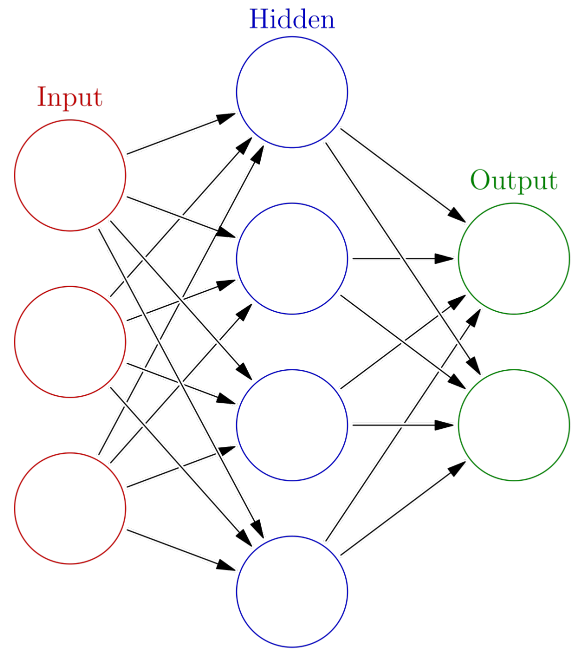
# Proposal #1: measuring canonicalization

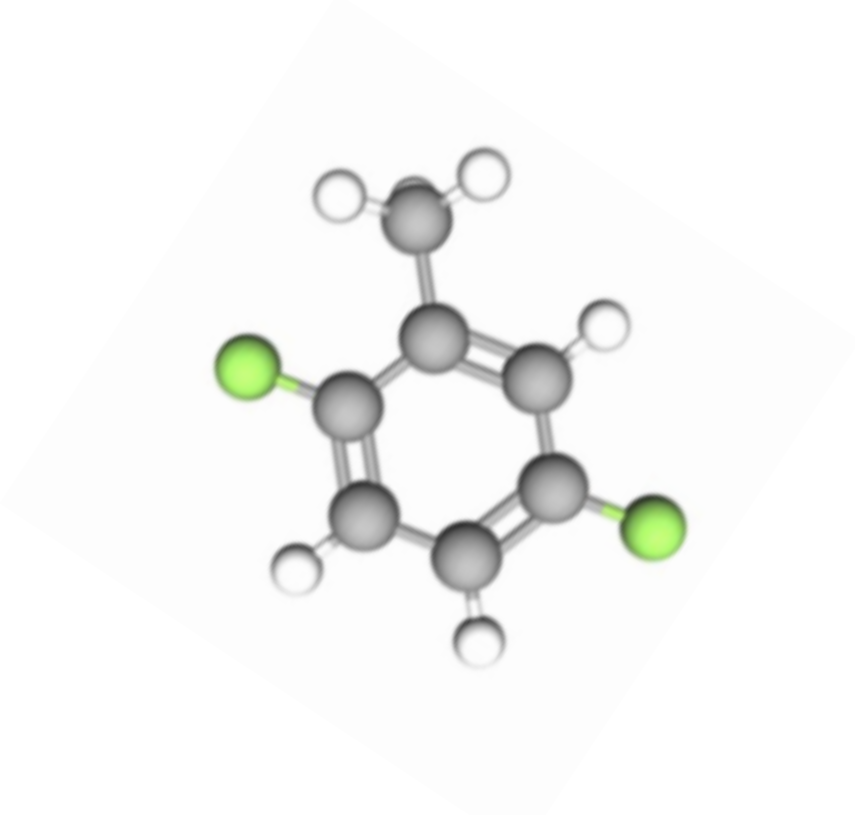


Split it into two halves



Class label 0

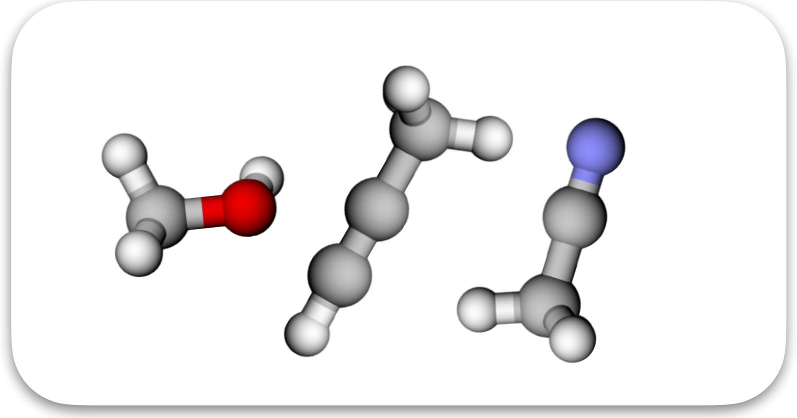


Class label 1

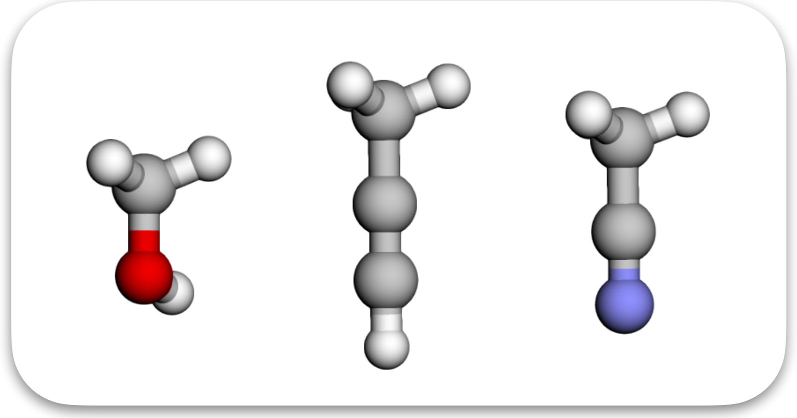
# Interpretation of the test accuracy

Hard



**If all rotations are equally likely in the dataset** → 50% accuracy

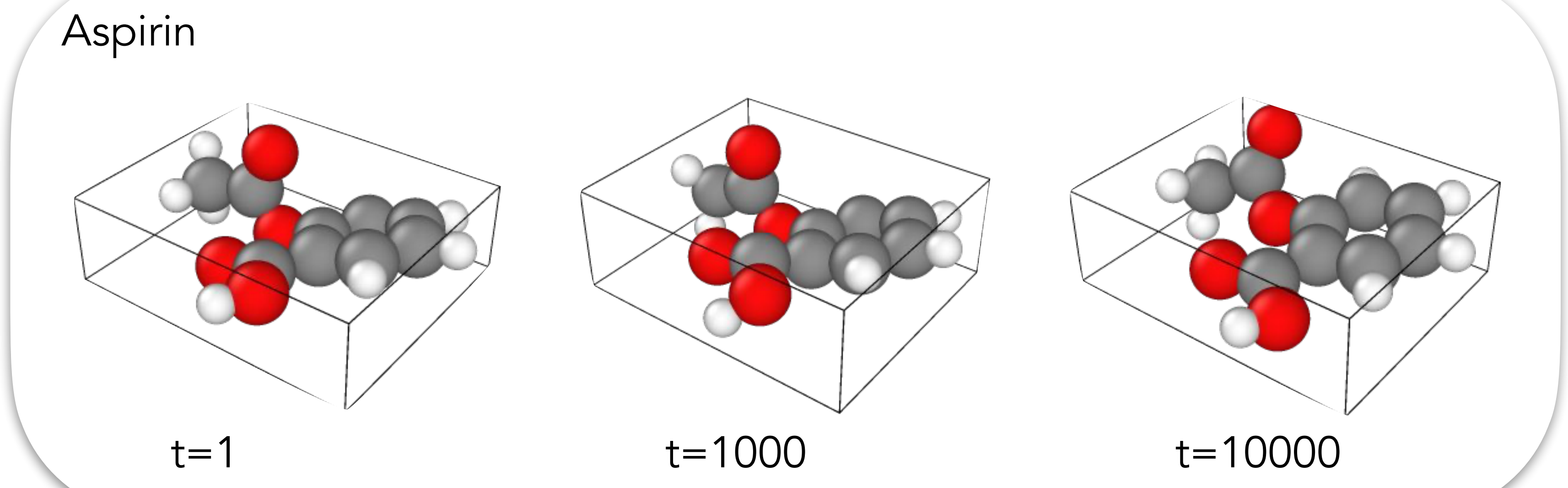
Easy



**If the dataset is fully canonicalized** → almost 100% accuracy

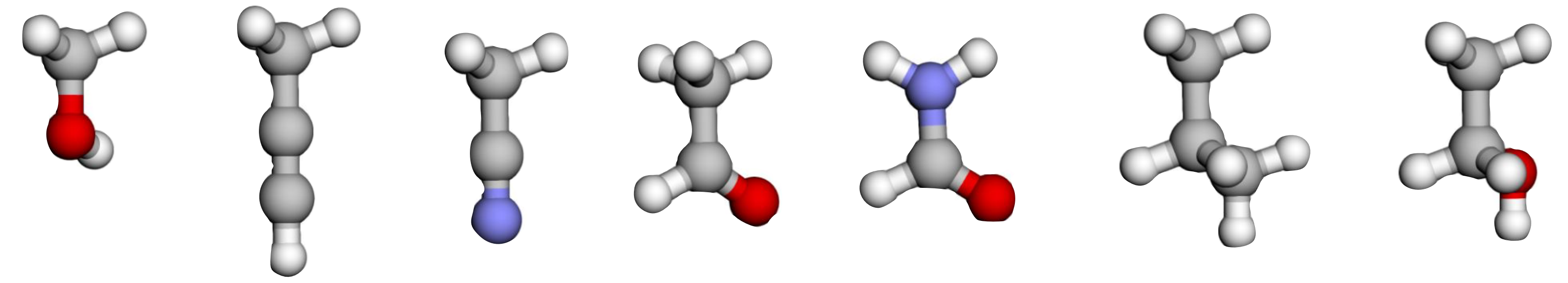
# What do we find in practice?

**MD17**: 79%-97% test accuracy (by molecule)



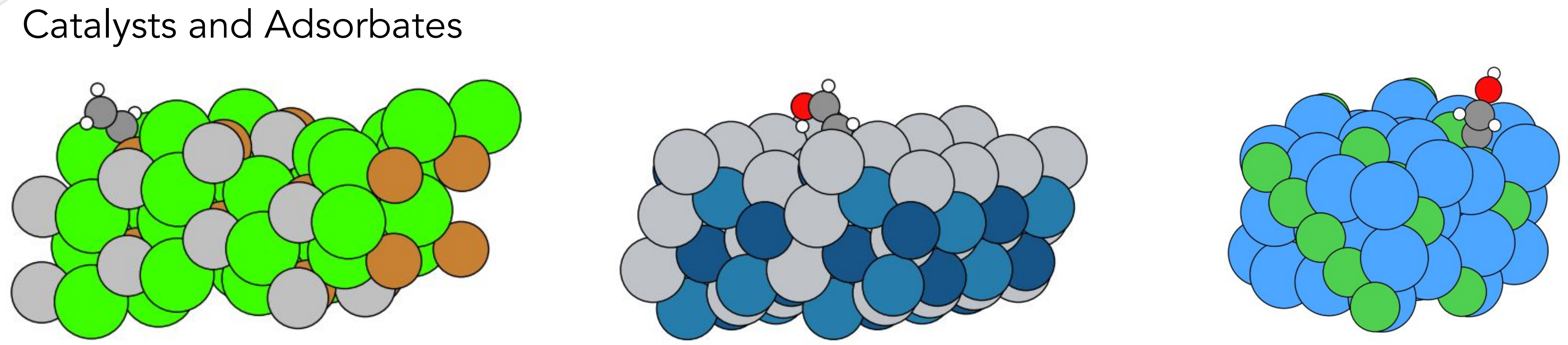

Relaxation trajectory determined by $t = 0$ orientation and physical structure of molecule

**QM9**: 99% test accuracy



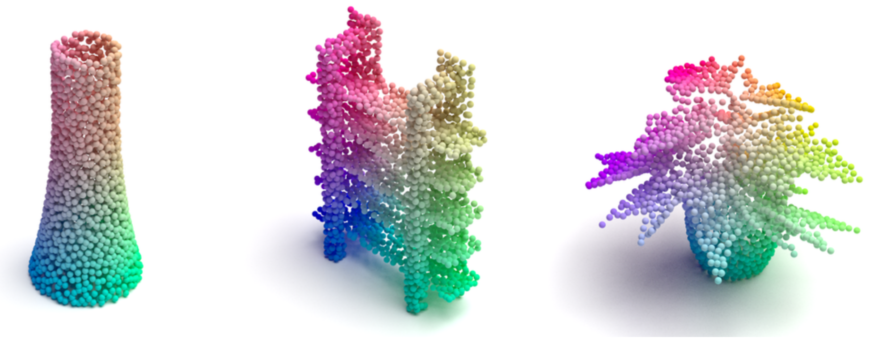
CORINA [5] used to generate conformers → may perform alignment

**OC20**: 97% test accuracy w/o surface, 99% with




Catalyst slab aligned with $xy$ plane

**ModelNet40**: 92% test accuracy



Objects have preferred orientations in reality

# What are the implications for learning?

Test-time augmentation

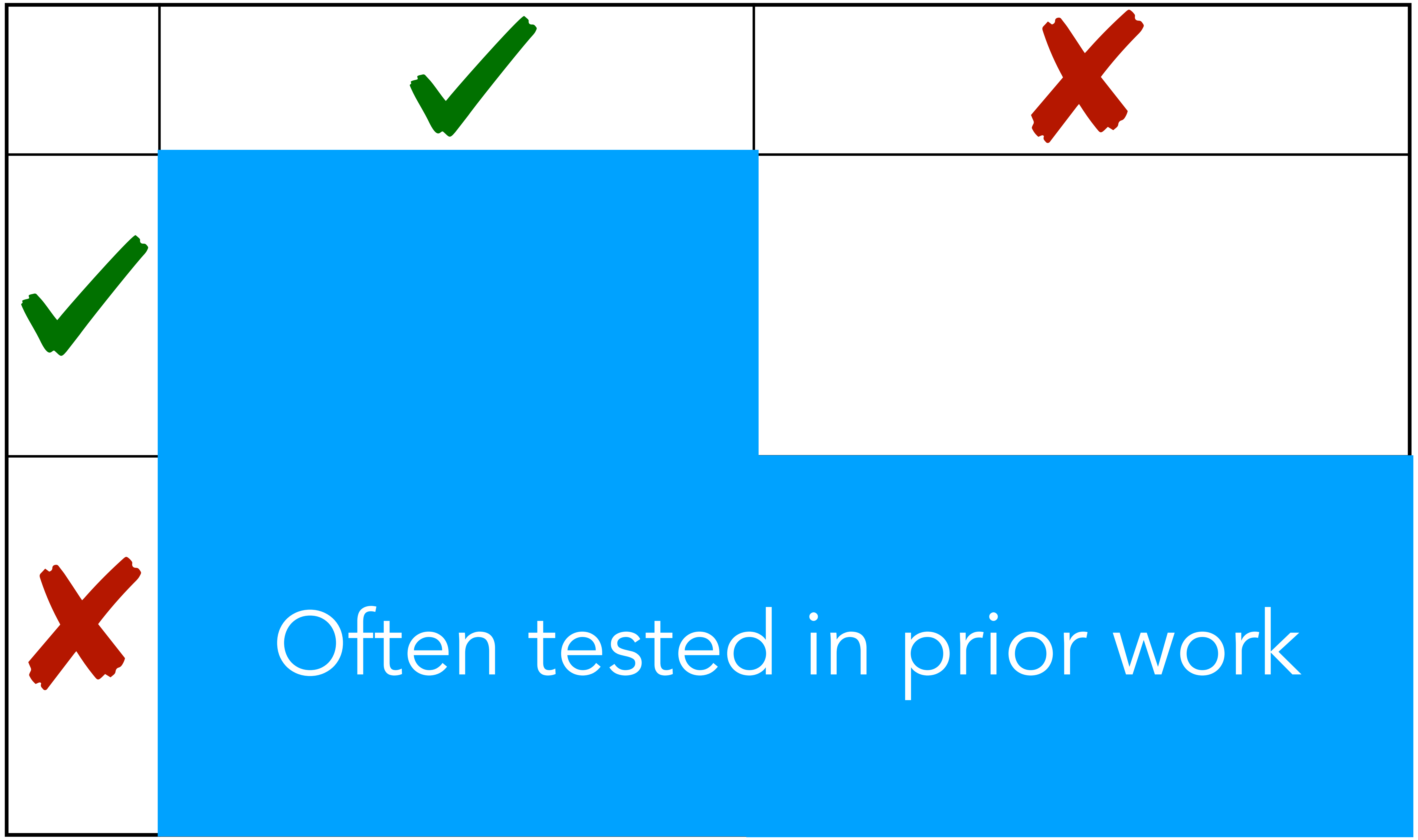
Train-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | | |
| ✗ | | |

# What are the implications for learning?

Test-time augmentation

Train-time augmentation



# What are the implications for learning?

Test-time augmentation

Train-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | Preferred orientations of inputs is lost | |
| ✗ | Preferred orientations of inputs is retained | |

# What are the implications for learning?

Test-time augmentation

Train-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | In-distribution | **Distribution shift** |
| ✗ | **Distribution shift** | In-distribution |

# What are the implications for learning?

Test-time augmentation

Train-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | Expected preferred setting of column | |
| ✗ | | Expected preferred setting of column |

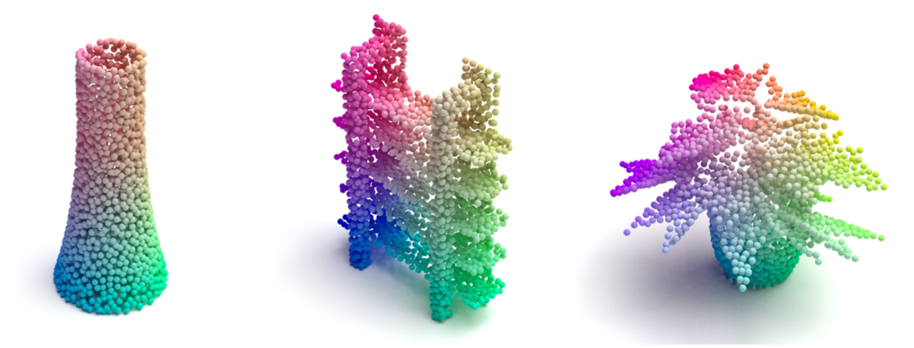
# What are the implications for learning?

**ModelNet40 Dataset**

Test-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | **75.7%** | 76.8% |
| ✗ | 7.9% | **84.5%** |

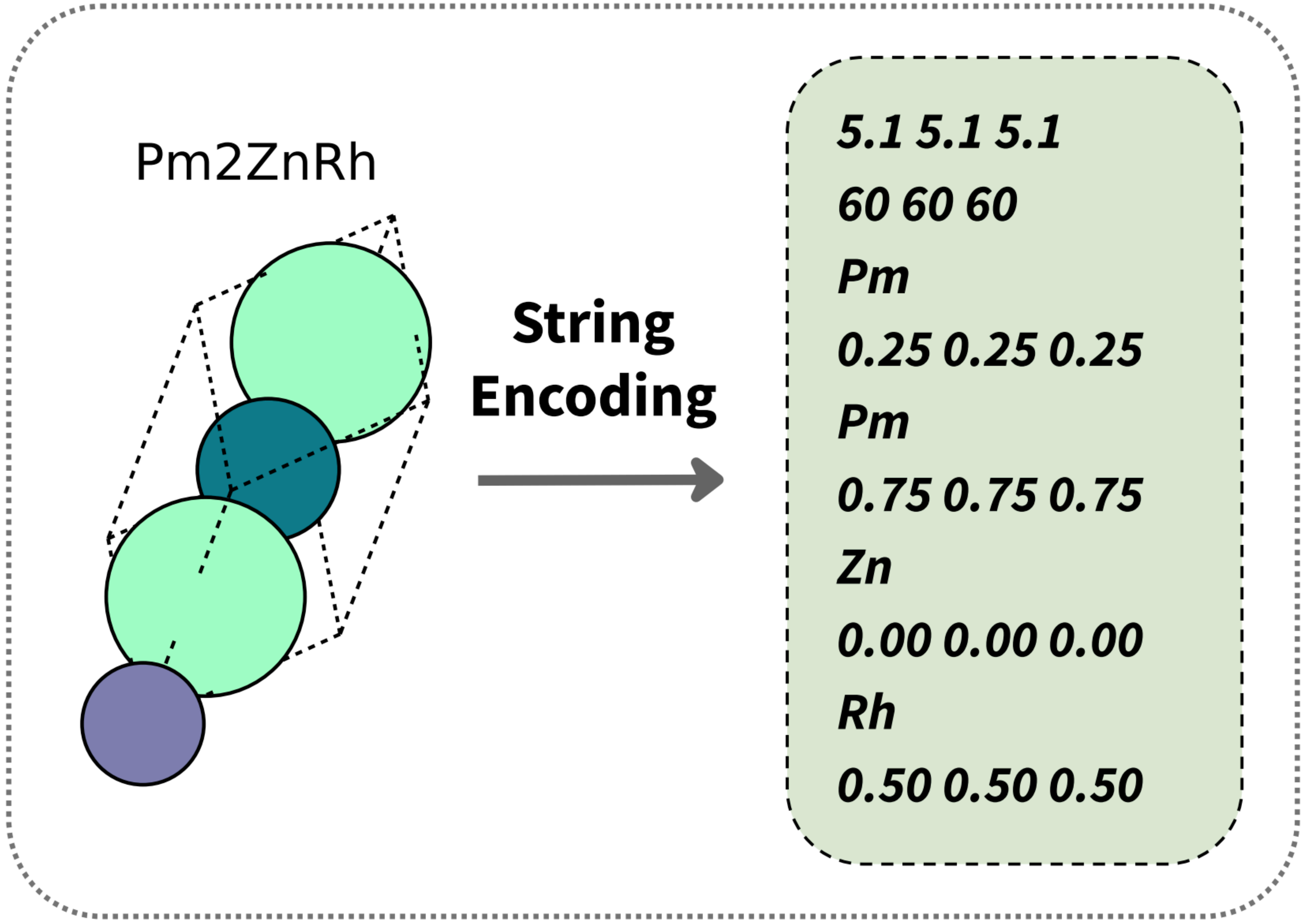
Train-time augmentation



(Accuracy, higher = better)

# Materials datasets used by LLMs?

Gruver et al, "Fine-Tuned LLMs Generate Stable Materials as Text"




Use LM to distinguish between this text vs atom-permuted version

Binary classification accuracy 95%
→ highly non-random orderings!

*Moreover: paper found that permutation augmentations hurt generative performance*

# What are the implications for learning?

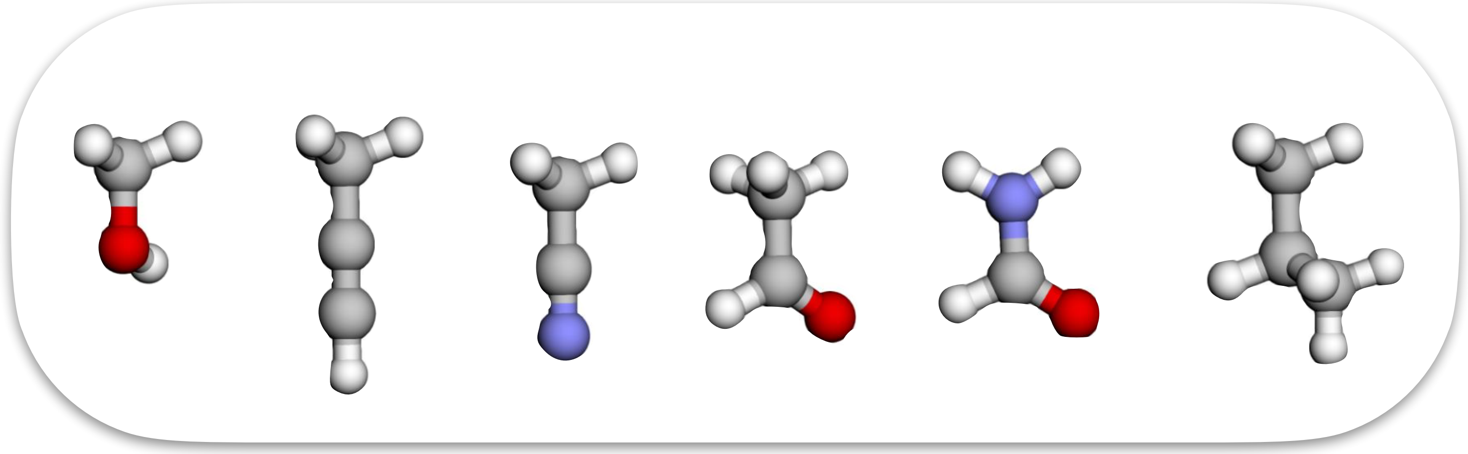
QM9
Dataset

Test-time augmentation

| | ✓ | ✗ |
|---|---|---|
| ✓ | **0.256** | **0.257** |
| ✗ | 0.283 | 0.289 |

Train-time au

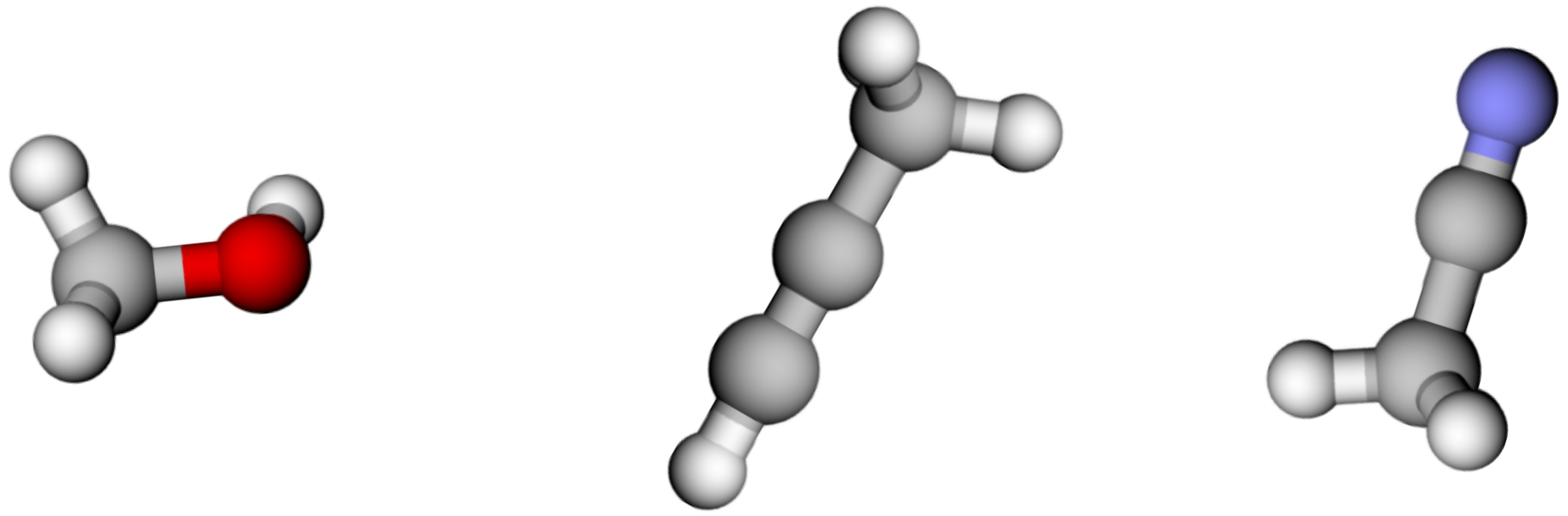
Augmentation helps on unaugmented test set — despite distribution shift!



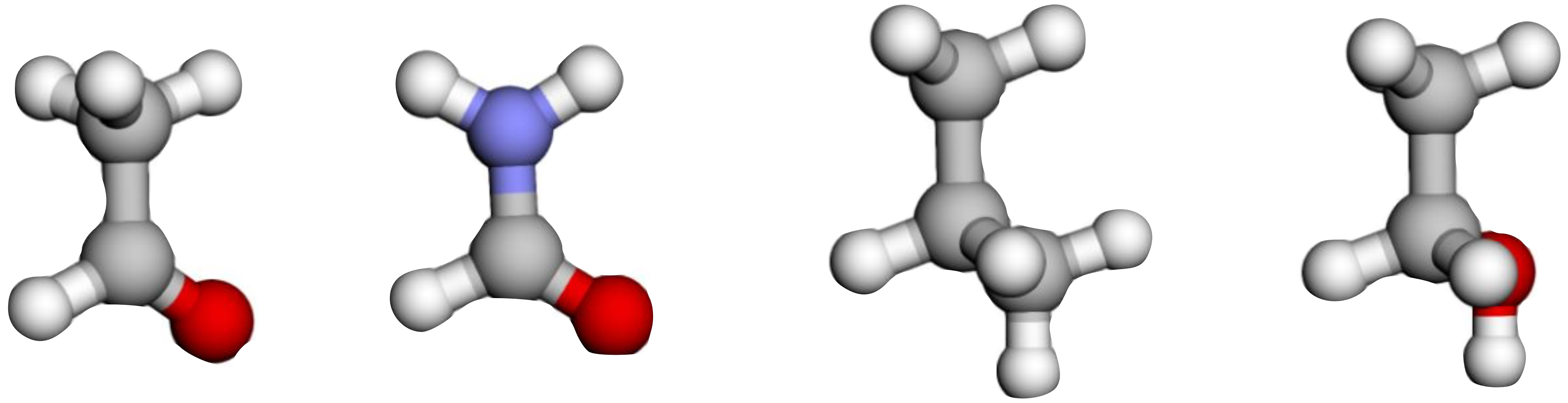
Regression target $\mu$

(MAE, lower = better)

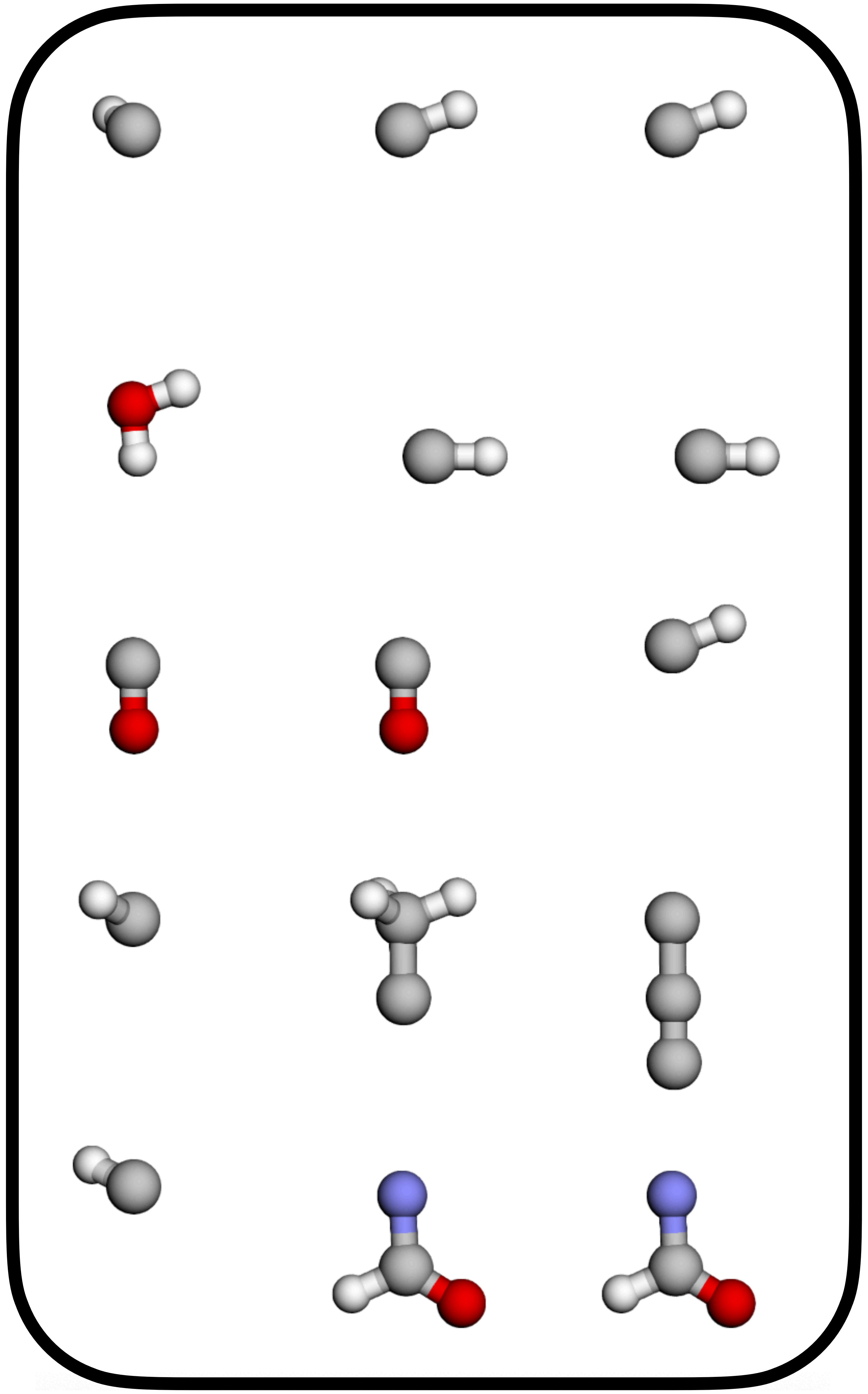
Why does training on this augmented dataset:

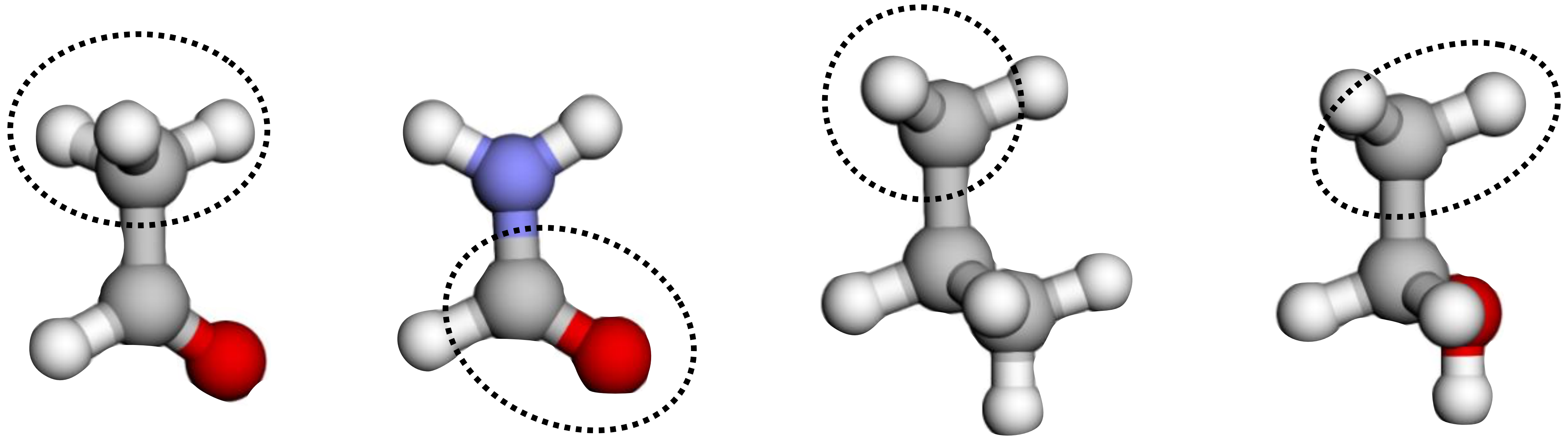


improve performance on this unaugmented test set?

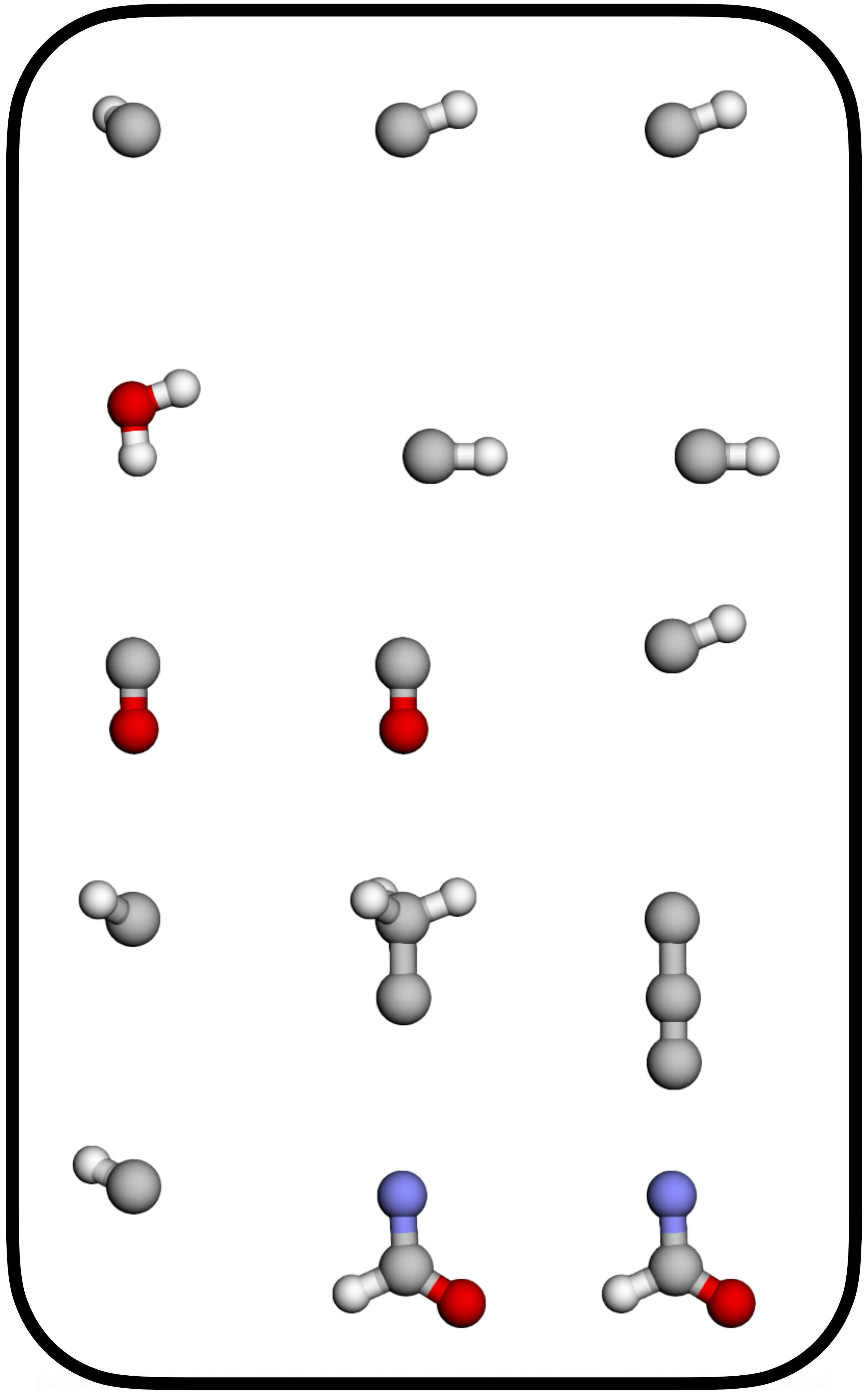


# Is locality the explanation?





# Is locality the explanation?



→ Binary classifier can only distinguish with **67.6%** test accuracy, compared to 99% on full dataset

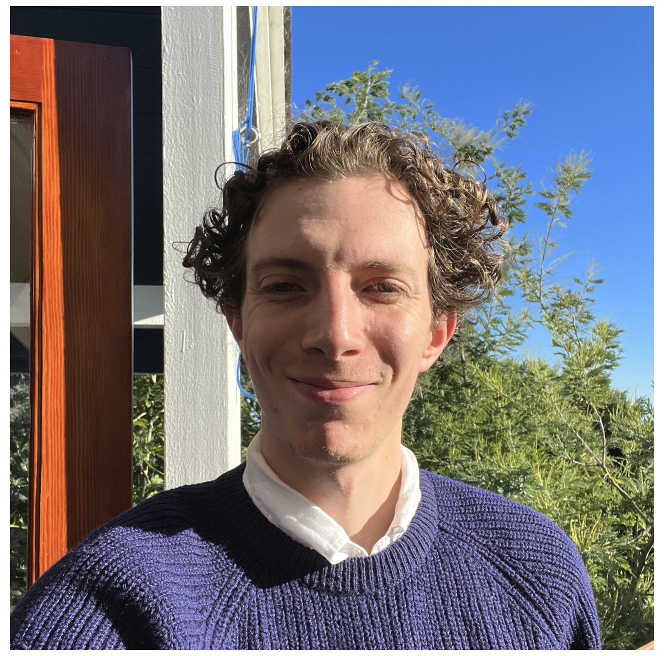
→ **local motifs in QM9 are more evenly distributed over poses, compared to full molecules**

Perhaps local equivariance is key to equivariant methods' success?

# In summary…

- Benchmark point cloud datasets, and at least one “LLM for science” dataset, are highly, quantifiably non-uniform over orientations
    - This has implications for whether you should use equivariant methods!
- In some cases (QM9), augmentation might be doing more than we thought
- **Not covered**: we have theory to back things up in the linear case!
- **Interesting/ongoing directions**: quantifying *task-useful* canonicalization (although: spurious vs. useful correlation will always require an expert, or a test set!)

# Thanks!



**Elyssa Hofgard**
MIT

**Vasco Portilheiro**
Gatsby, UCL

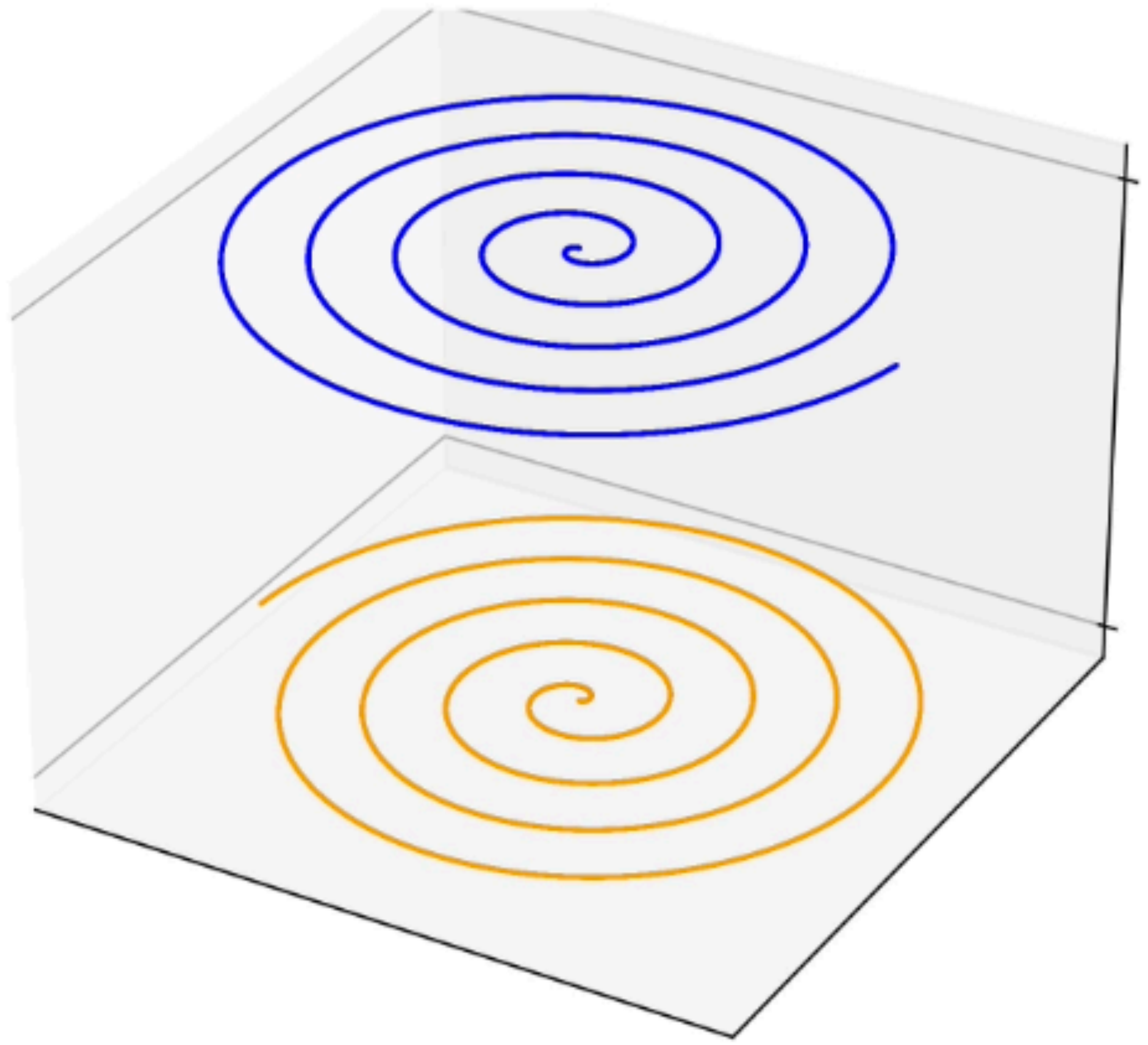
**Yuxuan Chen**
NEU

**Tess Smidt**
MIT

**Robin Walters**
NEU

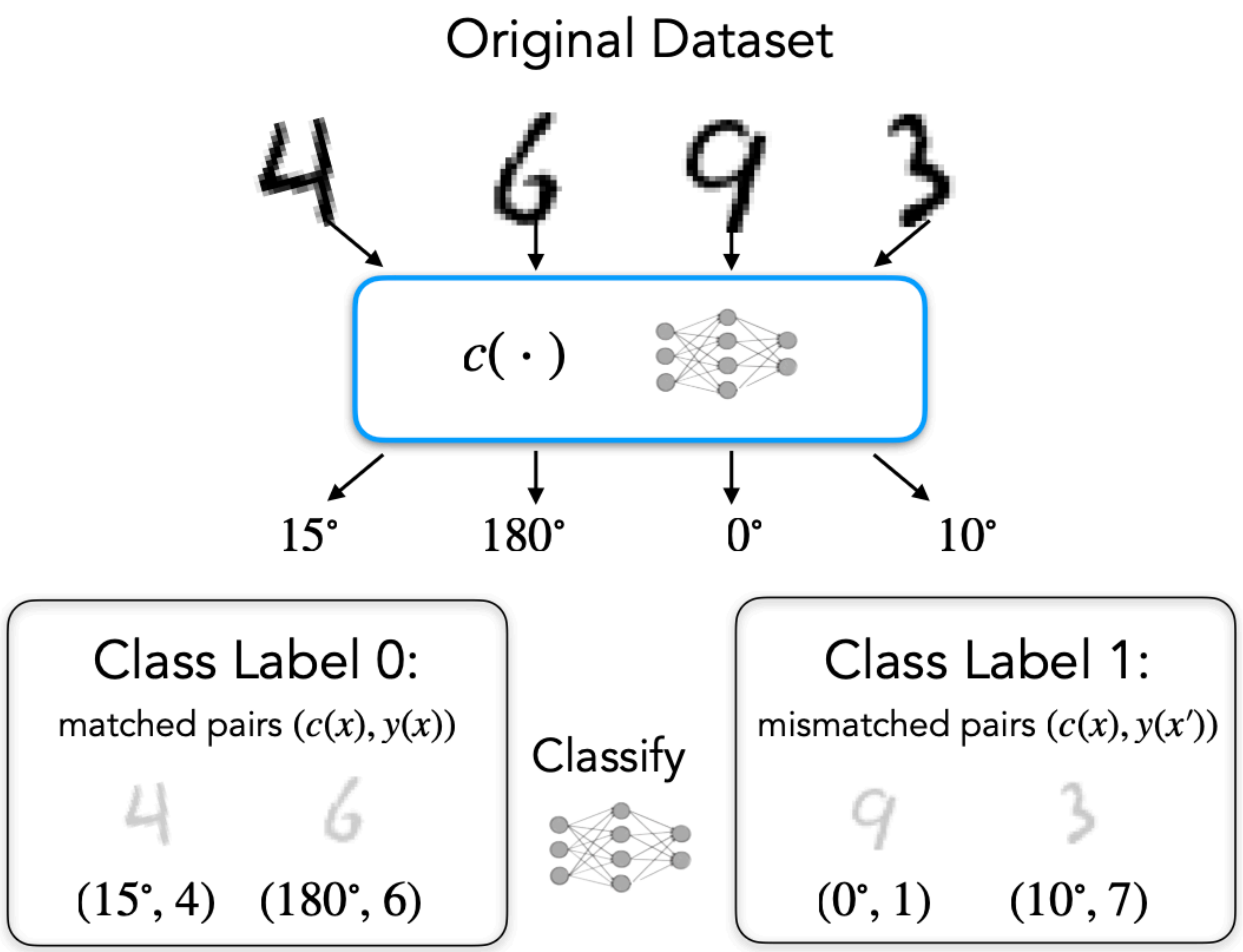
# Backup slides

# Spiral dataset

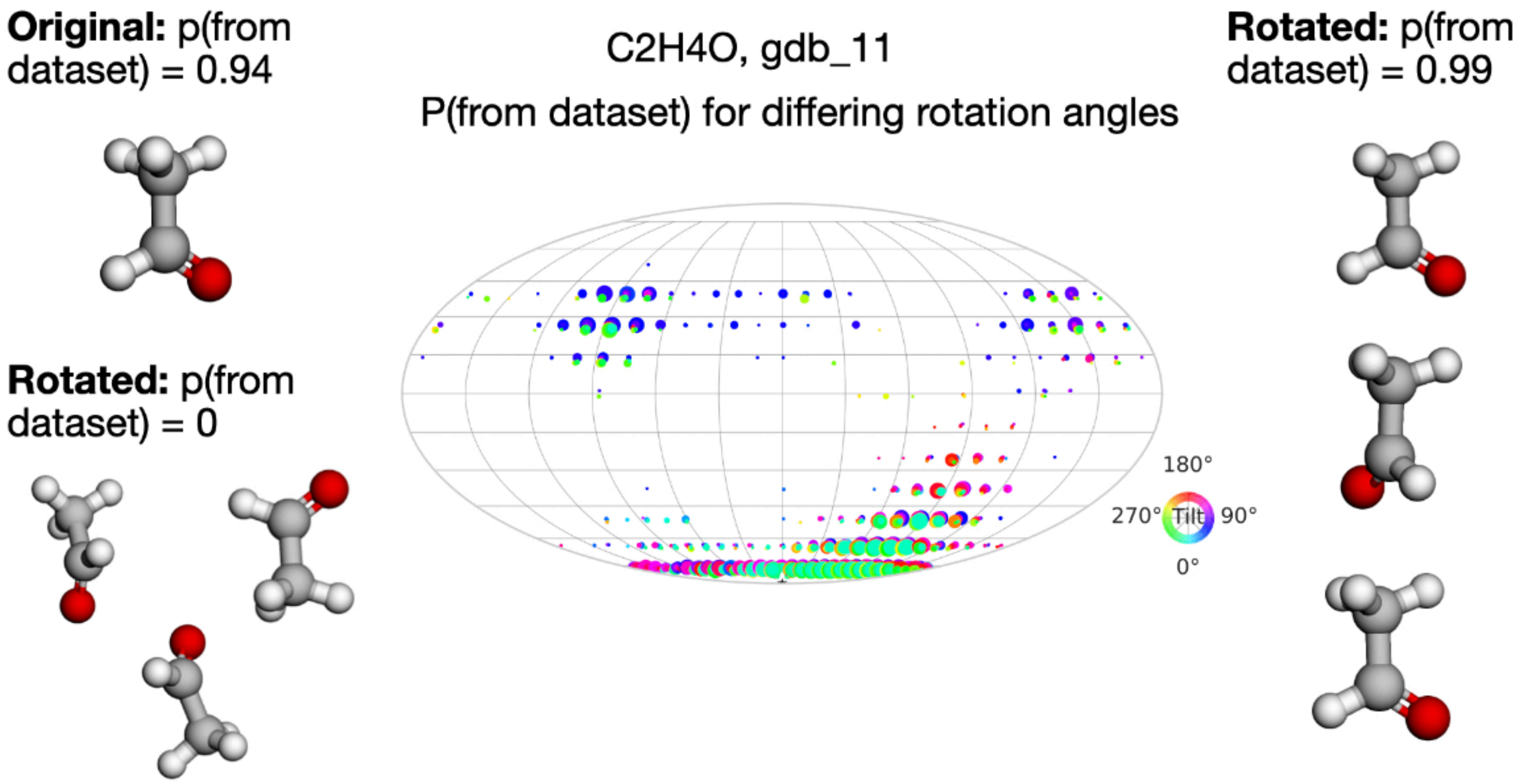


# Task-dependent metric

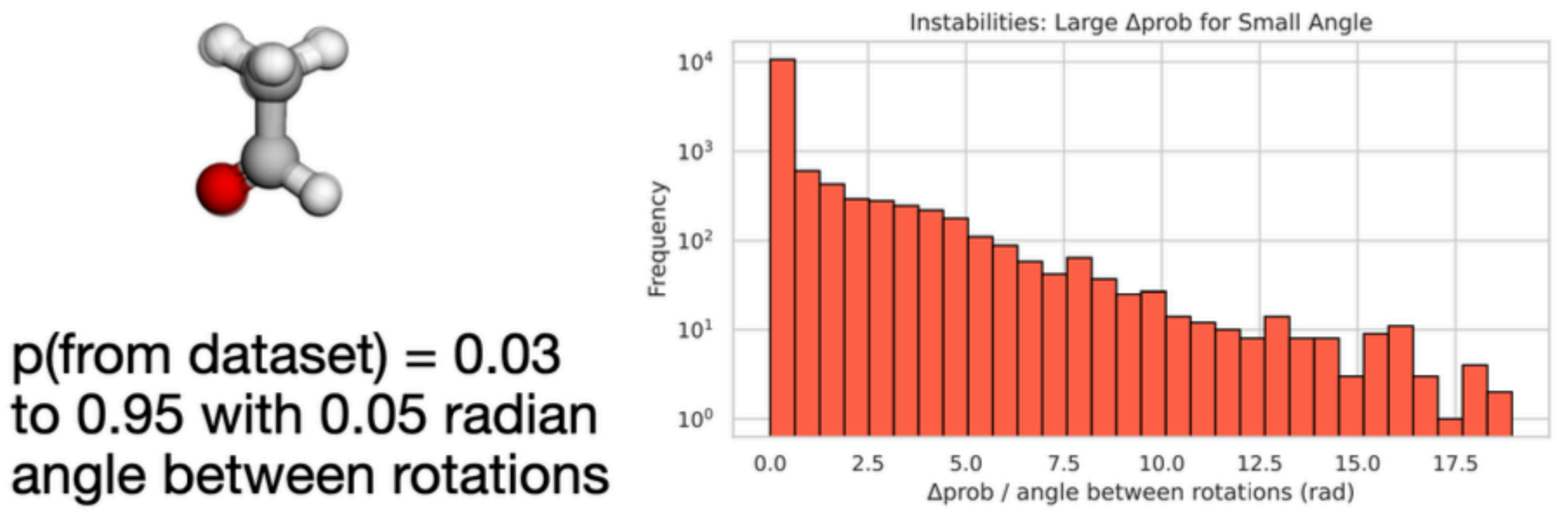
- Inspired by mutual information between pose and label



# Visualizing what the classifier learned




Probabilities per rotation angle for a sample that is classified correctly. The colors correspond to rotations orthogonal to the sphere and the size of the dots corresponds to the probability value. The original molecule in the dataset is shown on the upper left. The lower left shows rotations that cause `prob(original)` to be zero. The right shows rotations that cause `prob(original)` to be high.




Instability in the decision boundary (left): two nearby rotations cause a large change in predicted probability. Histogram (right): certain examples exhibit such instabilities more frequently.

Figure 25: Visualizations of classifier outputs for an example classified incorrectly.