# Information-Theoretic Probing with Minimum Description Length

Lena Voita & Ivan Titov
University of Edinburgh

Charles Lovering
charles_lovering@brown.edu

# Resources

mdl paper

---

mdl code

---

mdl blog

---

bayesian layers paper

---

bits-back argument paper

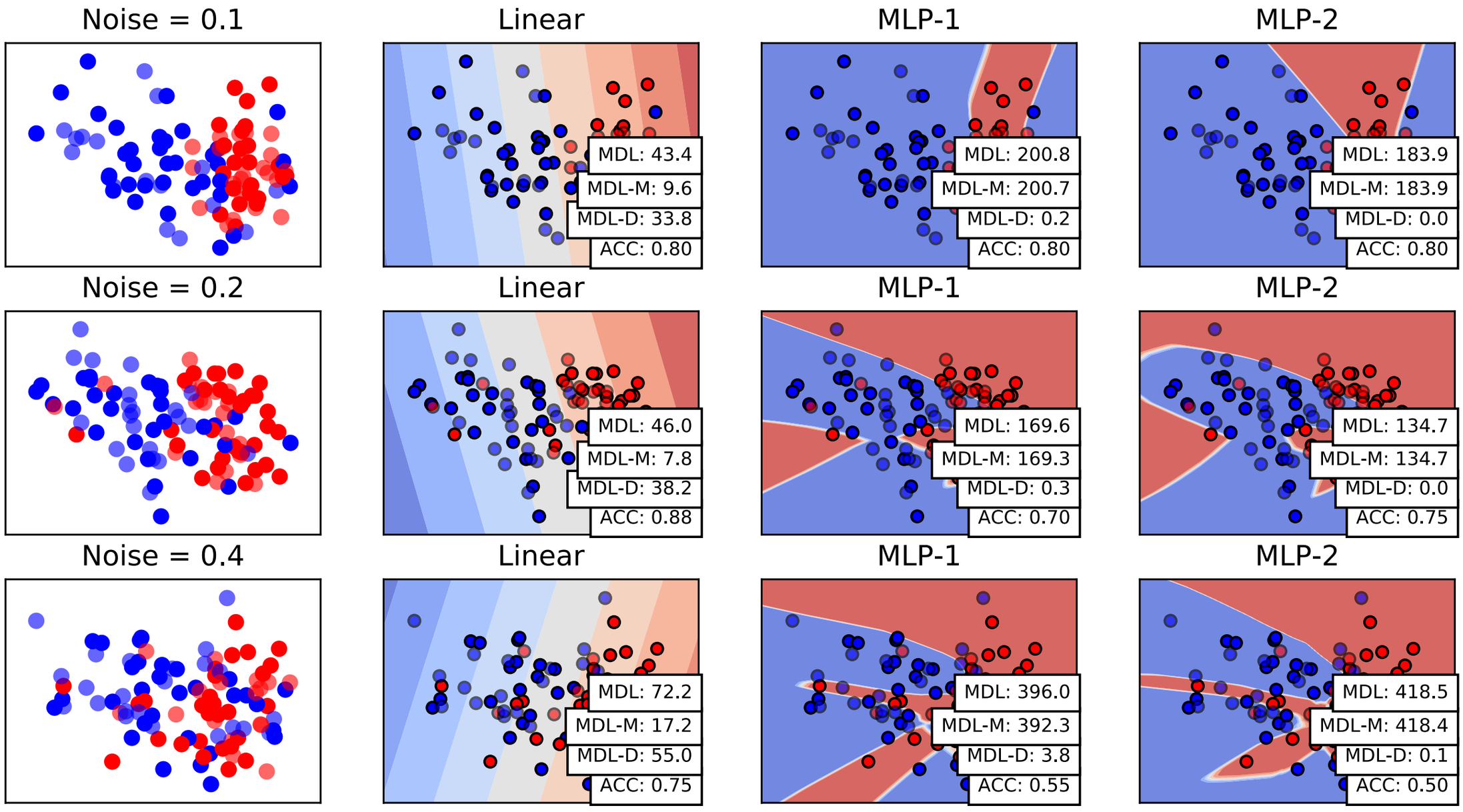
---

probabilistic graphical models course

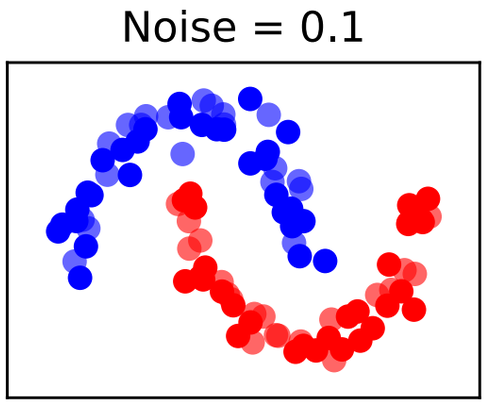
# Small Experiments For Building Intuition

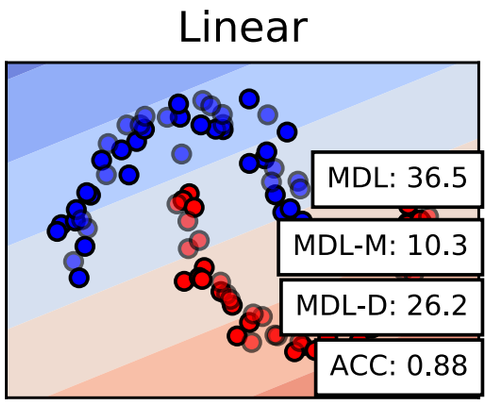
The next four slides are some results I put together: Press on if you don't know what MDL is already. However, my presentations are meant to be presented (not read), so you should probably just go to the paper/blog.

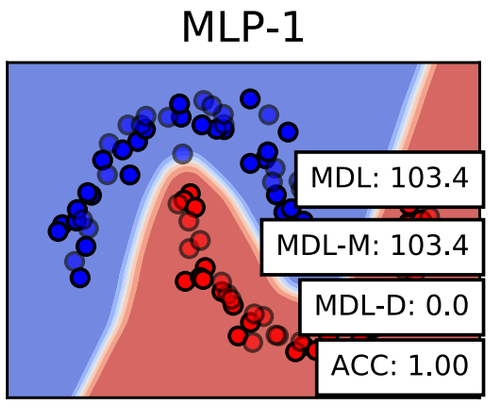
# Linearly Separable

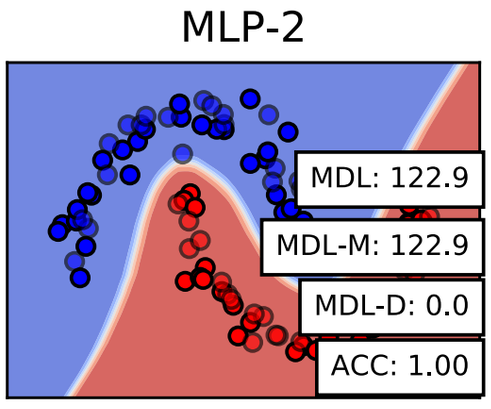


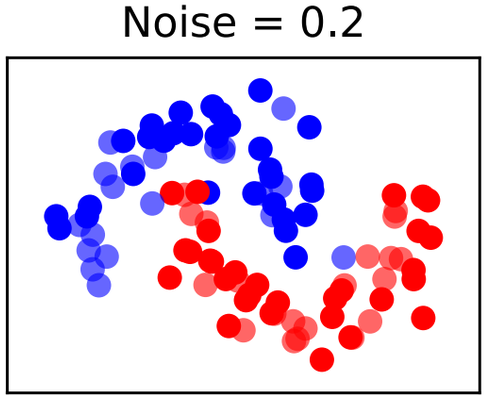
# Moons

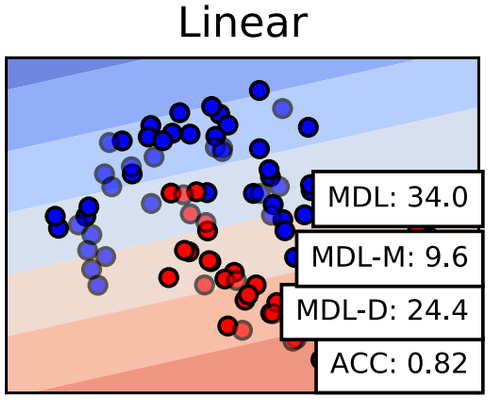

Noise = 0.1

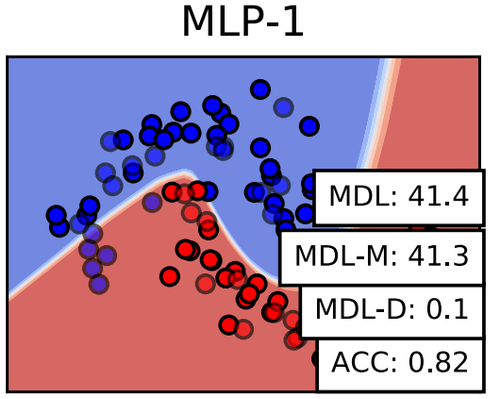

Linear
MDL: 36.5
MDL-M: 10.3
MDL-D: 26.2
ACC: 0.88

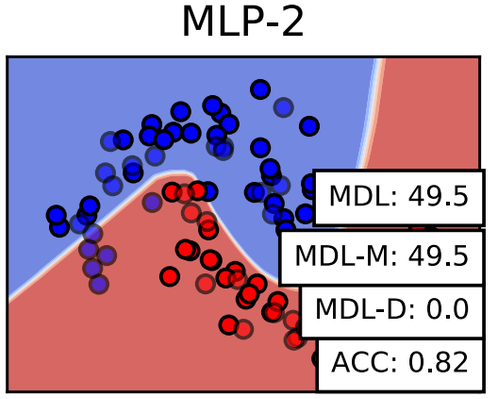

MLP-1
MDL: 103.4
MDL-M: 103.4
MDL-D: 0.0
ACC: 1.00

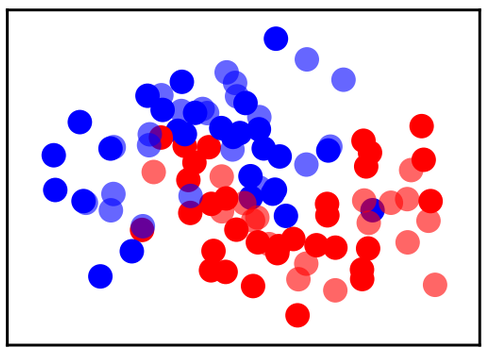

MLP-2
MDL: 122.9
MDL-M: 122.9
MDL-D: 0.0
ACC: 1.00

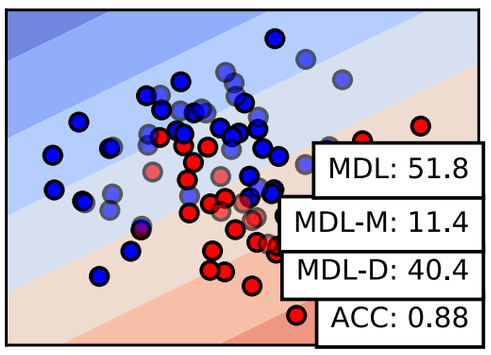

Noise = 0.2

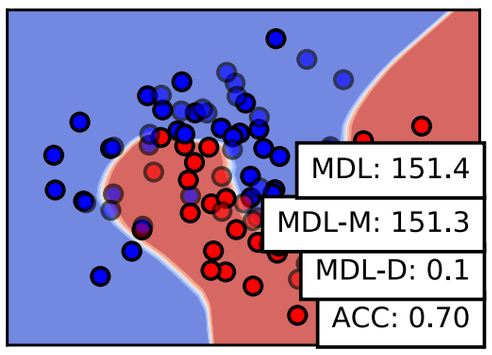

Linear
MDL: 34.0
MDL-M: 9.6
MDL-D: 24.4
ACC: 0.82

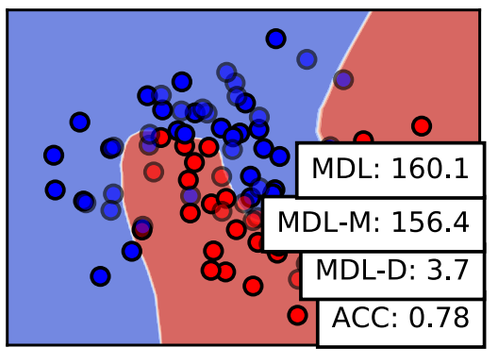

MLP-1
MDL: 41.4
MDL-M: 41.3
MDL-D: 0.1
ACC: 0.82


MLP-2
MDL: 49.5
MDL-M: 49.5
MDL-D: 0.0
ACC: 0.82

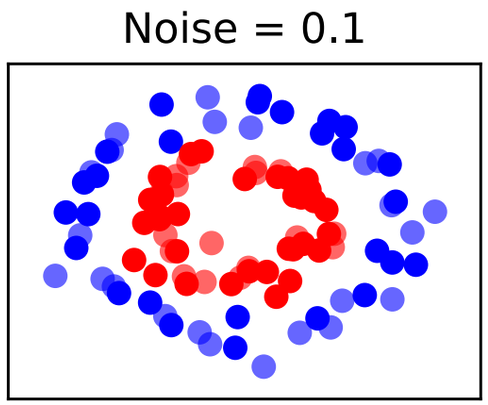

Noise = 0.4

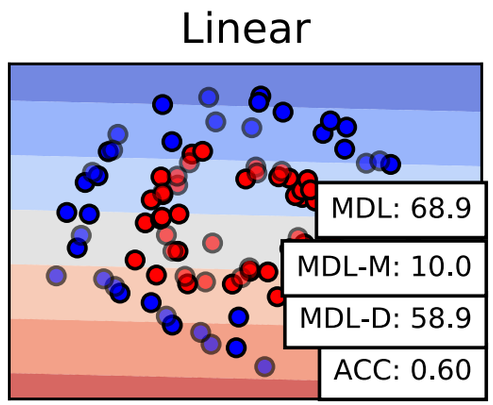

Linear
MDL: 51.8
MDL-M: 11.4
MDL-D: 40.4
ACC: 0.88

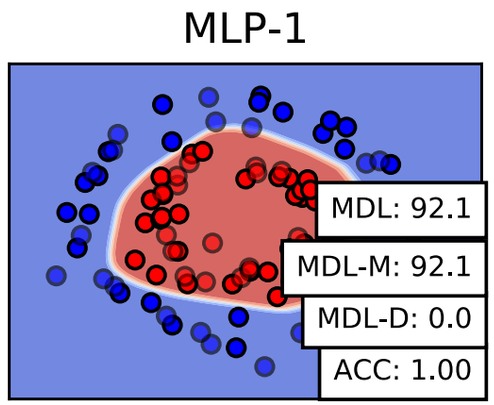

MLP-1
MDL: 151.4
MDL-M: 151.3
MDL-D: 0.1
ACC: 0.70

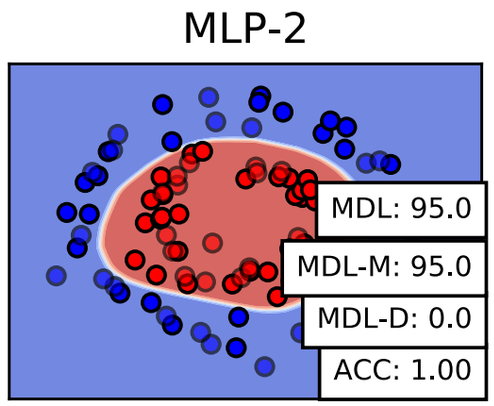

MLP-2
MDL: 160.1
MDL-M: 156.4
MDL-D: 3.7
ACC: 0.78

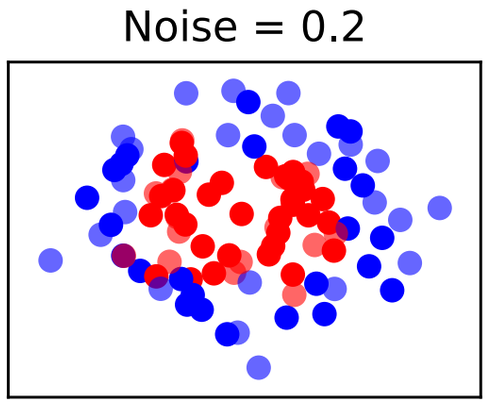
# Circles

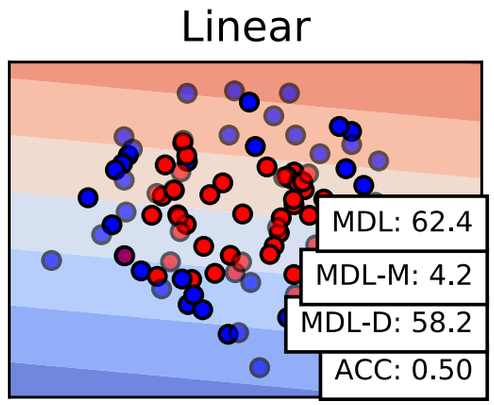

Noise = 0.1

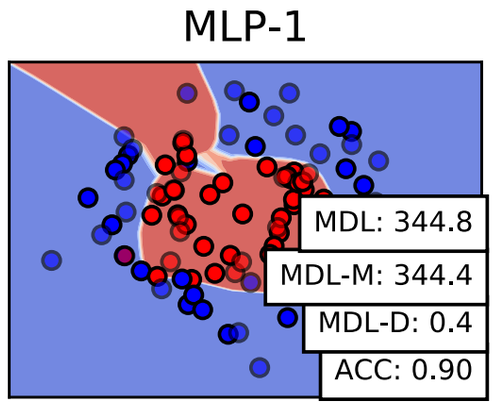

Linear
MDL: 68.9
MDL-M: 10.0
MDL-D: 58.9
ACC: 0.60

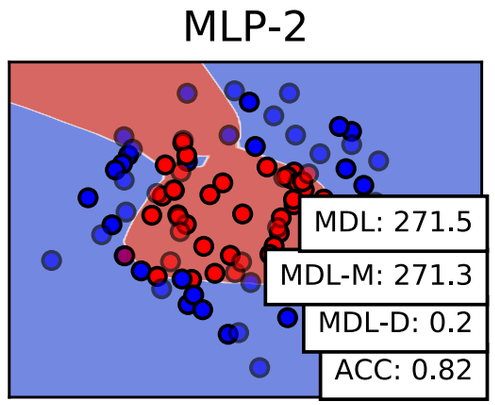

MLP-1
MDL: 92.1
MDL-M: 92.1
MDL-D: 0.0
ACC: 1.00

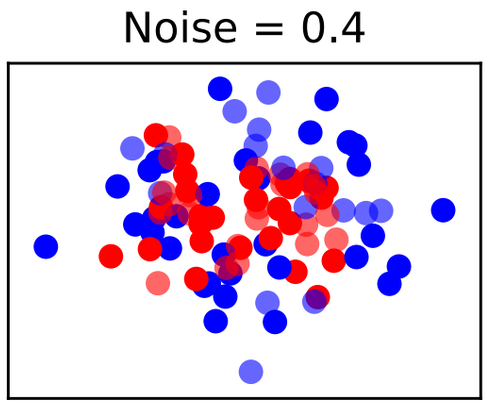

MLP-2
MDL: 95.0
MDL-M: 95.0
MDL-D: 0.0
ACC: 1.00

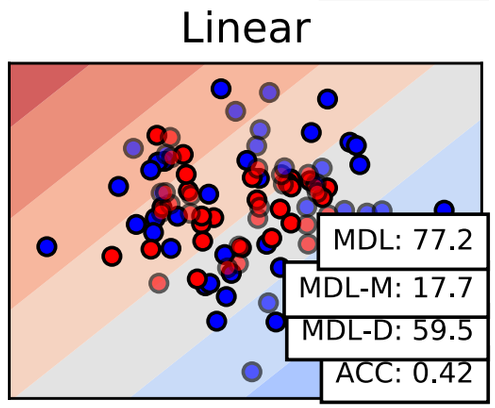

Noise = 0.2

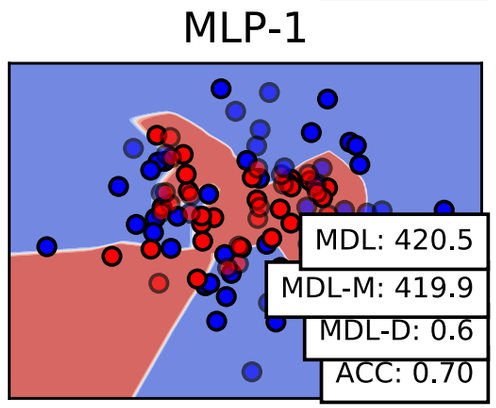

Linear
MDL: 62.4
MDL-M: 4.2
MDL-D: 58.2
ACC: 0.50

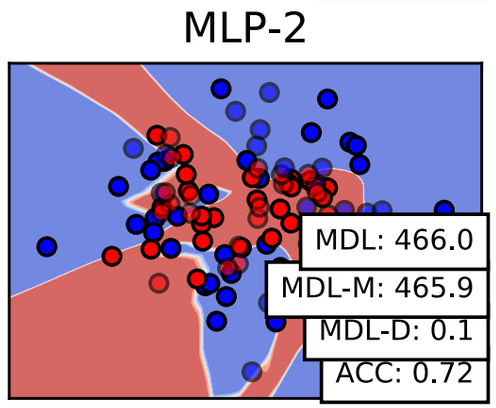

MLP-1
MDL: 344.8
MDL-M: 344.4
MDL-D: 0.4
ACC: 0.90


MLP-2
MDL: 271.5
MDL-M: 271.3
MDL-D: 0.2
ACC: 0.82


Noise = 0.4


MDL: 77.2
MDL-M: 17.7
MDL-D: 59.5
ACC: 0.42

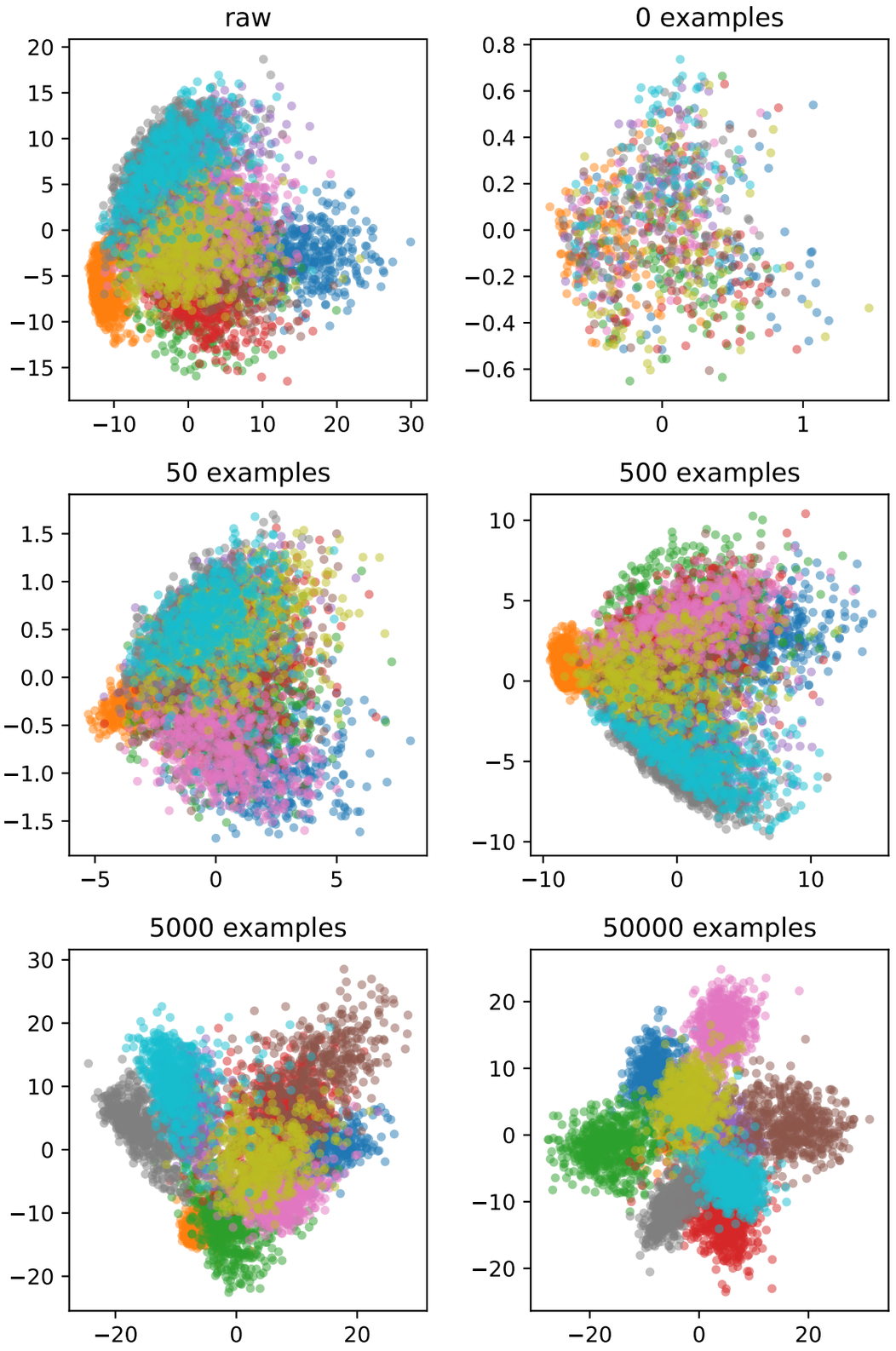

MDL: 420.5
MDL-M: 419.9
MDL-D: 0.6
ACC: 0.70


MDL: 466.0
MDL-M: 465.9
MDL-D: 0.1
ACC: 0.72

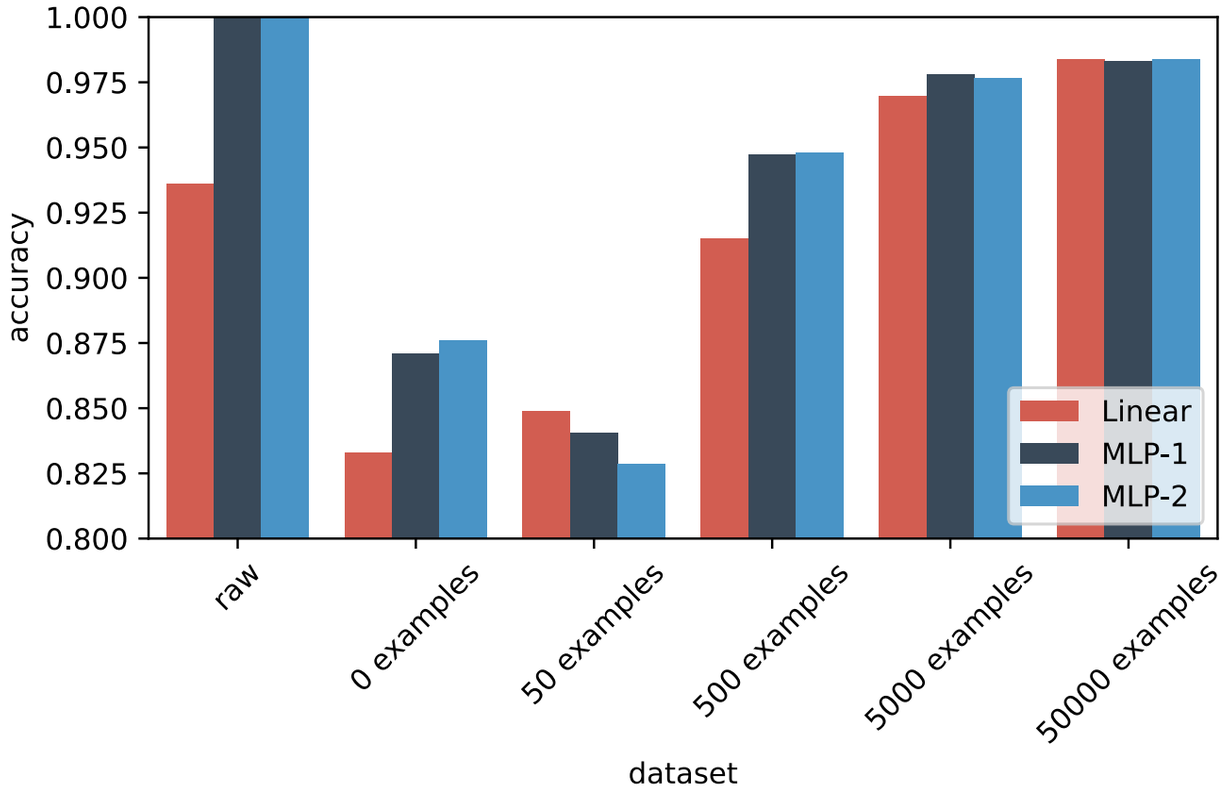
# MNIST

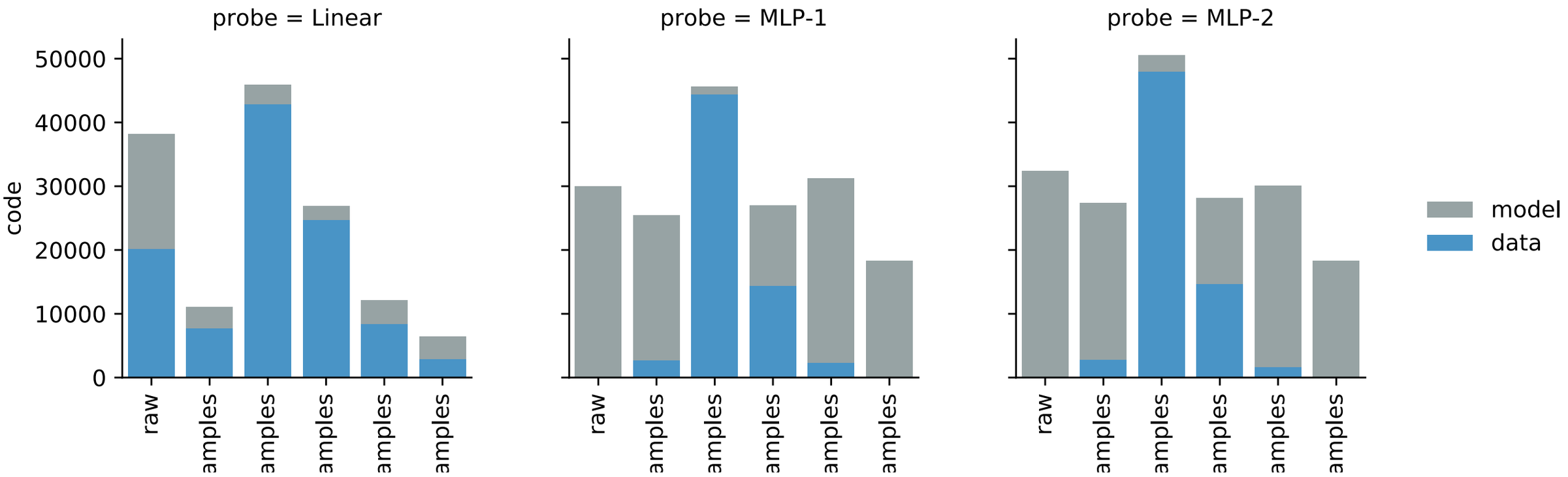
Raw are the raw images flattened. Each other chart show representations processed by a CNN. For example, at 0 examples, the model is untrained. At 50K examples, the model has done slightly less than one epoch (over all 60K examples).

(All charts use PCA with 2 dimensions.)



# MNIST


1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
accuracy
Linear
MLP-1
MLP-2
raw
0 examples
50 examples
500 examples
5000 examples
50000 examples
dataset


probe = Linear
probe = MLP-1
probe = MLP-2
50000
40000
30000
20000
10000
0
code
raw
model
data

# My Take-Aways

1. MDL is a function of the model and the data.
2. A given representation can only be better (have lower MDL) than another with reference to a shared setup, and still it remains hard to compare.
3. It is hard to compare the scale of various MDLs.

# TLDR

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

# TLDR

Choose the model that gives the shortest description.

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

Rissanen, Jorma. "Modeling by shortest data description." *Automatica* 14.5 (1978): 465-471.

# TLDR

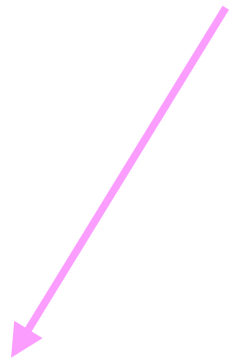
Choose the model that gives the shortest description.

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

Variational Inference & Bayesian Learning

# TLDR

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$



$$\mathcal{L}_{\text{Var}} = \text{KL}(\beta \parallel \alpha)$$
$$- \mathbb{E}_{\theta \sim \beta} \sum_{i=1}^{n} \log_2 p_\theta(y_i | x_i)$$

# TLDR

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

$$\mathcal{L}_{\text{Var}} = \mathrm{KL}(\beta \,\|\, \alpha) - \mathbb{E}_{\theta \sim \beta} \sum_{i=1}^{n} \log_2 p_\theta(y_i | x_i)$$

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

# TLDR

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

$$\mathcal{L}_{\text{Var}} = \mathrm{KL}(\beta \,\|\, \alpha) - \mathbb{E}_{\theta \sim \beta} \sum_{i=1}^{n} \log_2 p_\theta(y_i | x_i)$$

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

# TLDR

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

# Models are doing pretty well

# GLUE

| | |
|---|---|
| NLI | MNLI, QNLI, RTE, Winograd NLI |
| Other Semantic Tasks | Semantic Textual Similarity, Stanford Sentiment Treebank, MR Paraphrase |
| Acceptability | Cola |

# GLUE

| T5 | 90.3 |
|---|---|
| Human Baseline | 87.1 |
| BERT++ | 80.5 |
| BERT | 78.1 |

# SuperGLUE

Multi-Sentence Understanding

Common Sense

Non-trivial QA

# SuperGLUE

| | |
|---|---|
| Human Baseline | 89.8 |
| **T5** | **89.3** |
| BERT++ | 71.5 |
| BERT | 69.0 |

# Why do the models behave the way they do?

# What do models learn?

# Diagnostic classifiers; “probes”.

# Diagnostic classifiers; “probes”.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In International Conference on Learning Representations.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick. "Bert rediscovers the classical nlp pipeline." arXiv preprint arXiv:1905.05950 (2019).

Zhang, Kelly W., and Samuel R. Bowman. "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis." arXiv preprint arXiv:1809.10040 (2018).

Kim, Najoung, et al. "Probing what different NLP tasks teach machines about function word comprehension." arXiv preprint arXiv:1904.11544 (2019).

Warstadt, Alex, et al. "Investigating BERT's Knowledge of Language: Five Analysis Methods with NPIs." arXiv preprint arXiv:1909.02597 (2019).

Clark, Kevin, et al. "What Does BERT Look At? An Analysis of BERT's Attention." arXiv preprint arXiv:1906.04341 (2019).

Goldberg, Yoav. "Assessing BERT's Syntactic Abilities." arXiv preprint arXiv:1901.05287 (2019).

Conneau, Alexis, et al. "What you can cram into a single vector: Probing sentence embeddings for linguistic properties." arXiv preprint arXiv:1805.01070 (2018).

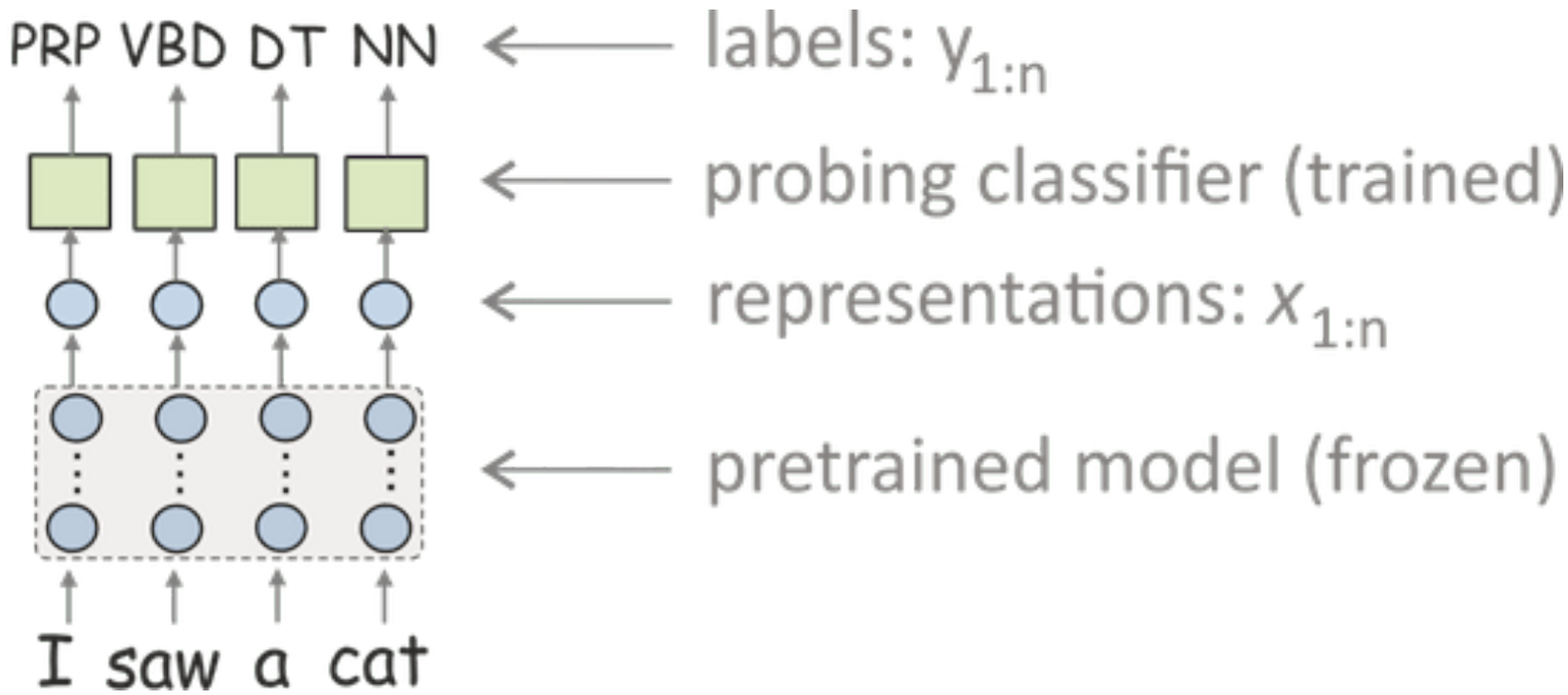
…

Goal:

predict labels

Measure:

probe accuracy

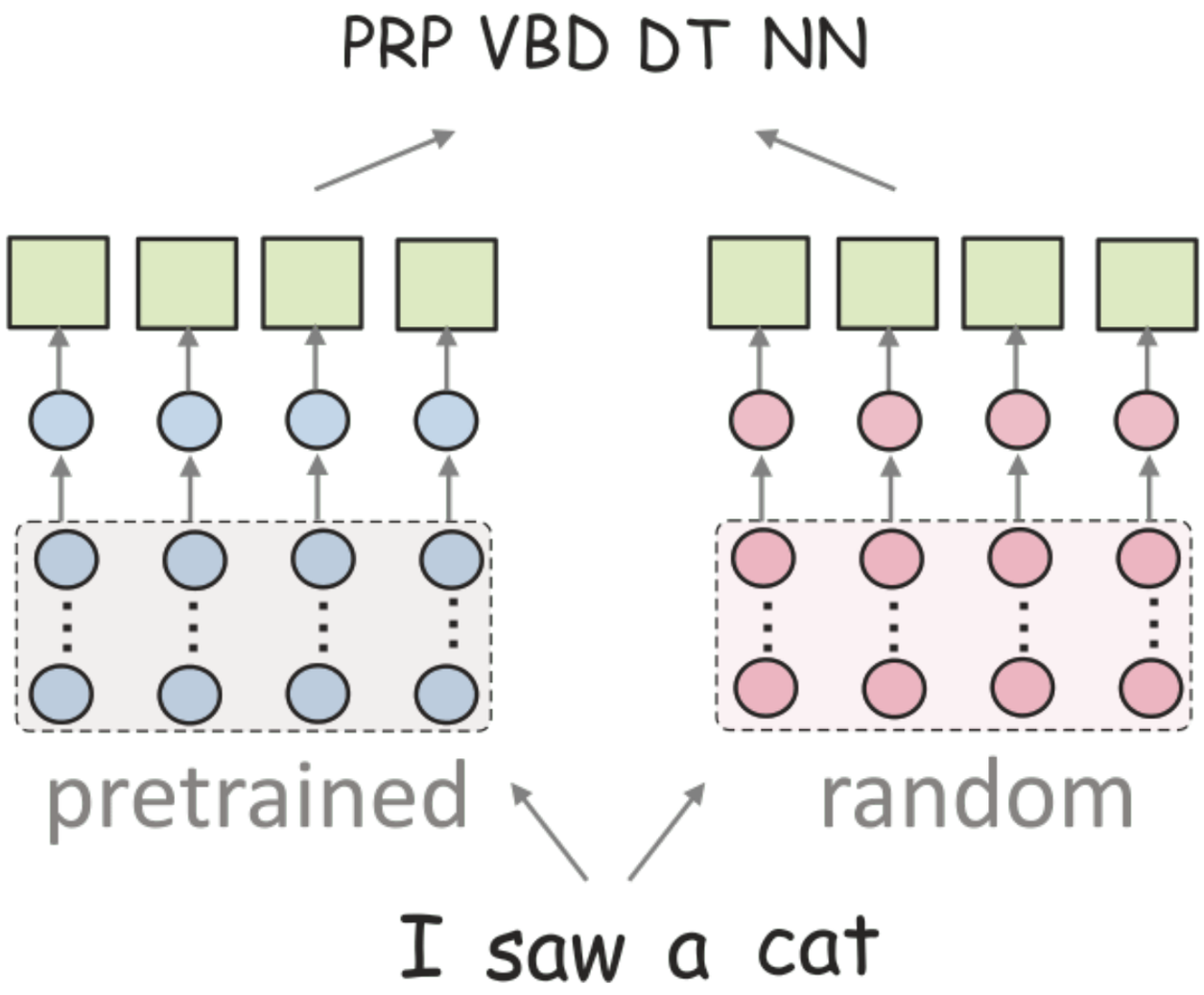



https://lena-voita.github.io/posts/mdl_probes.html

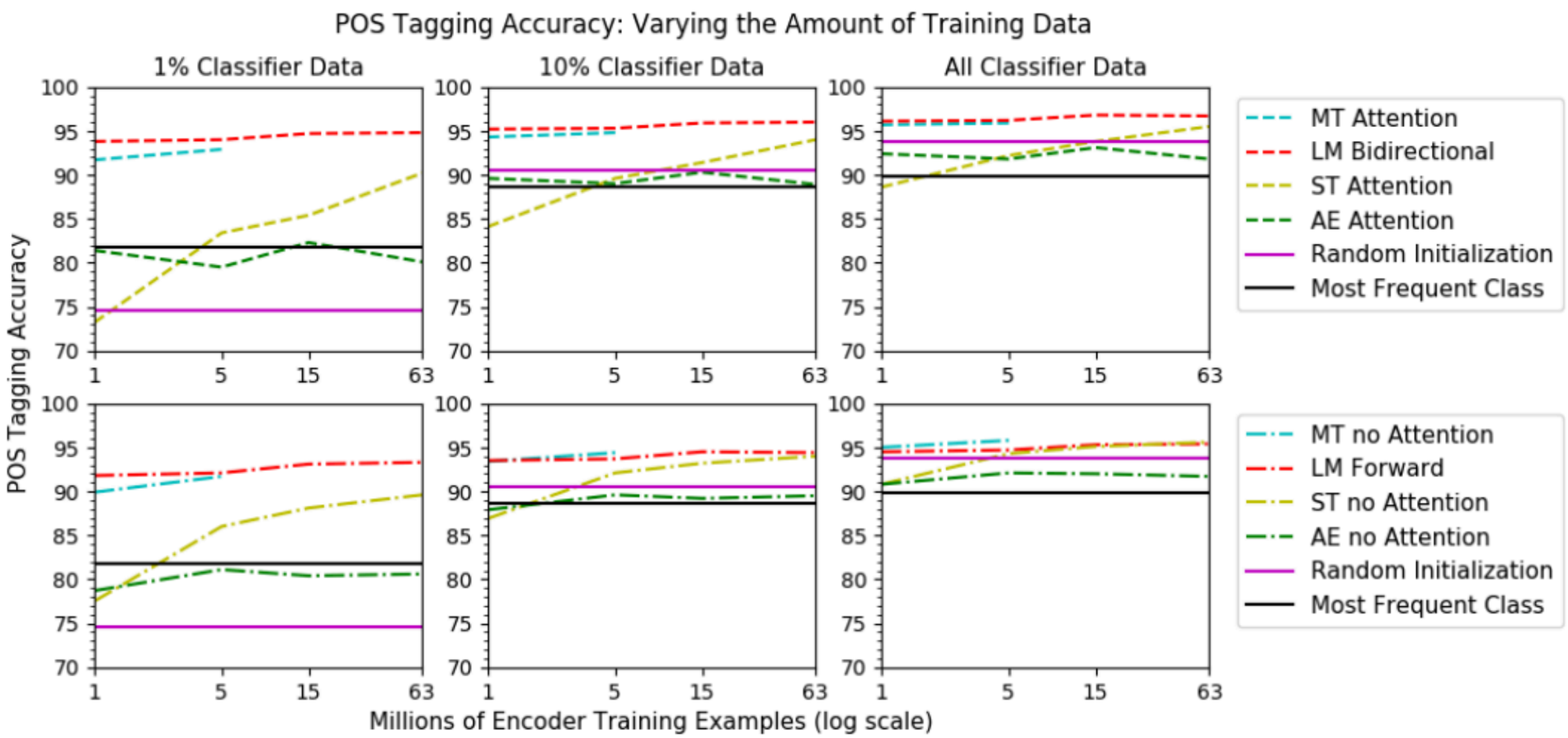
# 1. Models with random weights report similar accuracies to models with trained weights.

Zhang, Kelly W., and Samuel R. Bowman. "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis." *arXiv preprint arXiv:1809.10040* (2018).

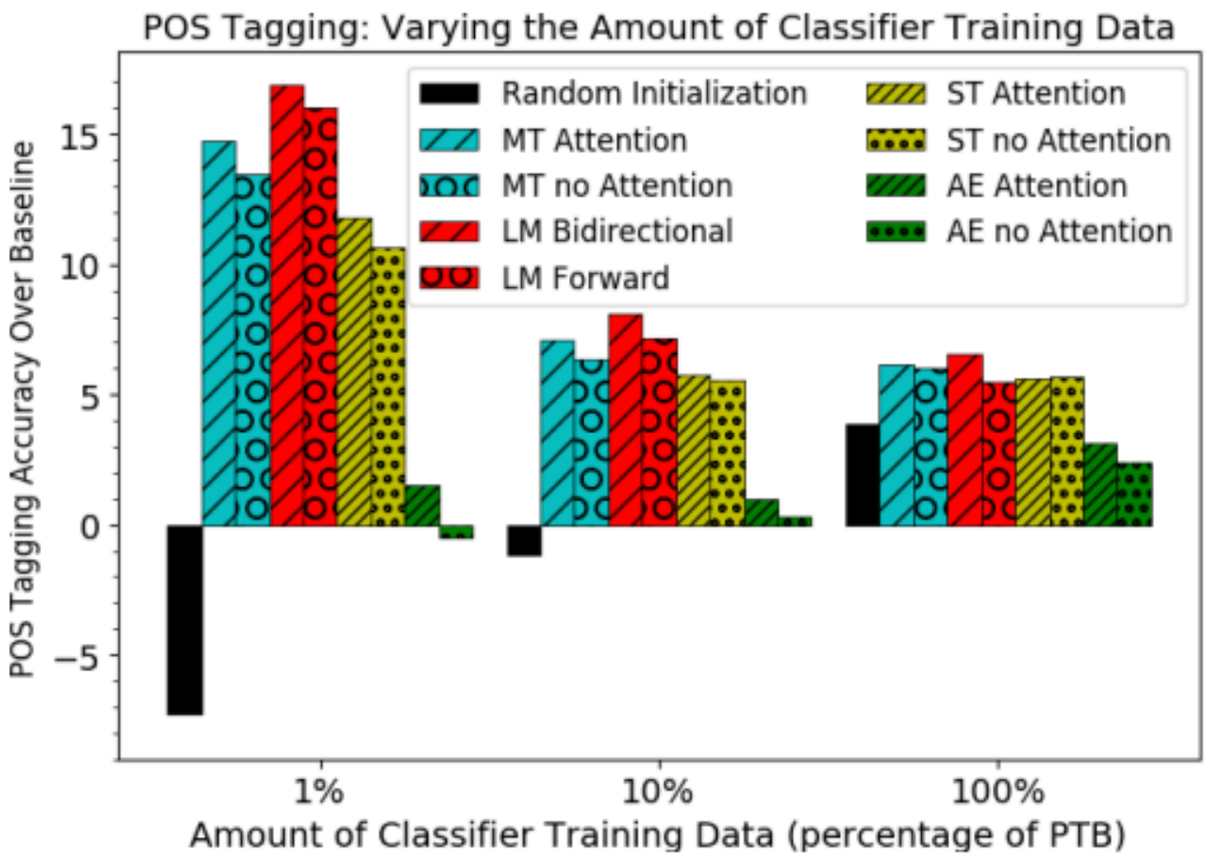
# Model: trained vs random




https://lena-voita.github.io/posts/mdl_probes.html




Zhang, Kelly W., and Samuel R. Bowman. "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis." *arXiv preprint arXiv:1809.10040* (2018).



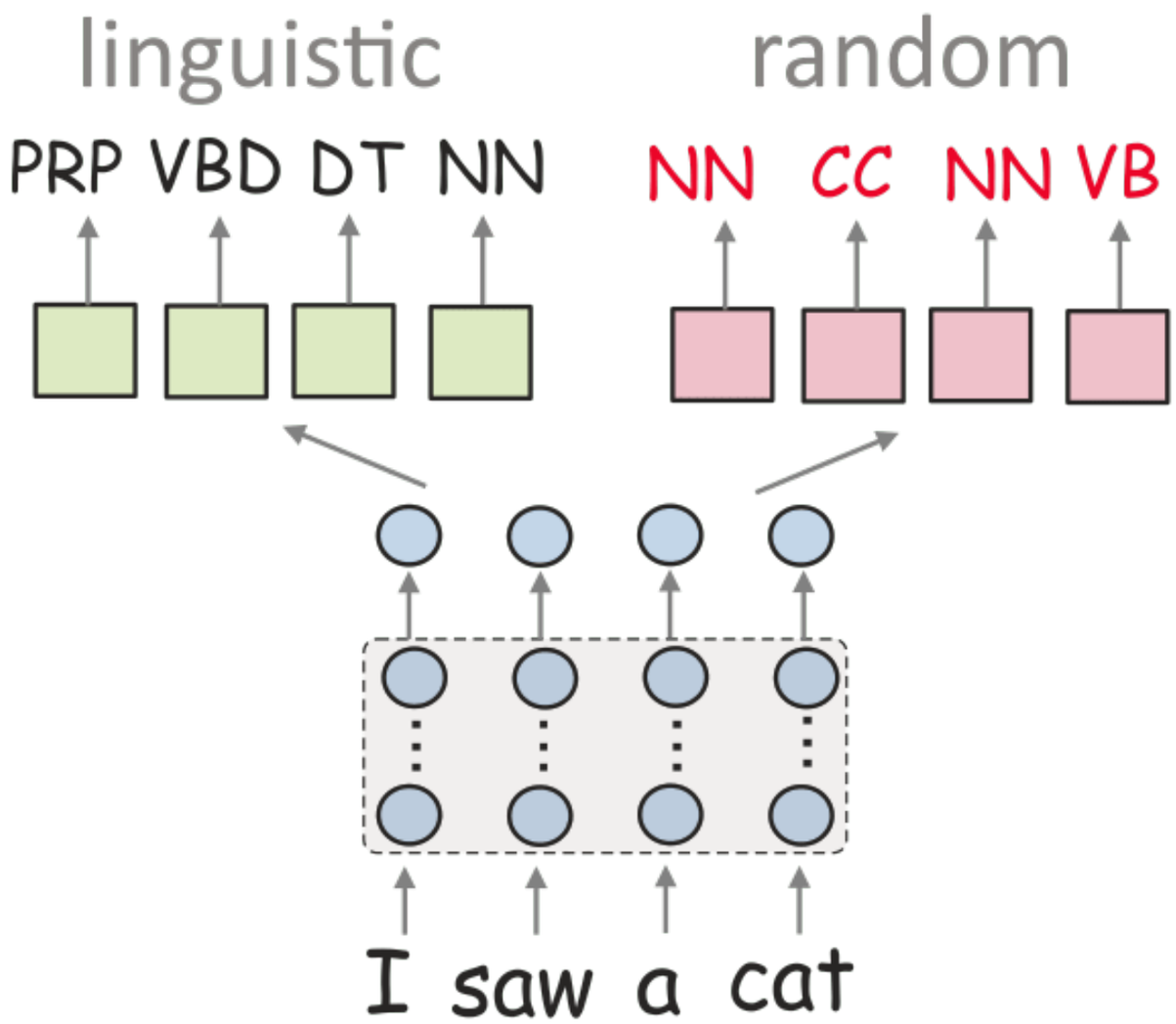

(a) WC-MFC baselines for different amounts of PTB training data: 1% PTB: 81.8%; 10% PTB: 88.6%; 100% PTB: 89.9%.

Zhang, Kelly W., and Samuel R. Bowman. "Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis." *arXiv preprint arXiv:1809.10040* (2018).

# 2. Models often perform equally well on tasks with randomly assigned variables.

Hewitt, John, and Percy Liang. "Designing and Interpreting Probes with Control Tasks." arXiv preprint arXiv:1909.03368 (2019).

# Labels: linguistic vs random

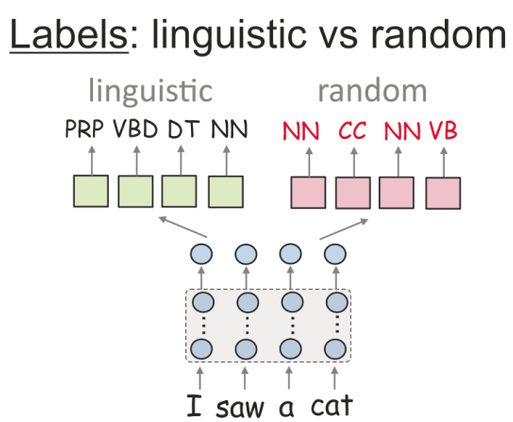



https://lena-voita.github.io/posts/mdl_probes.html

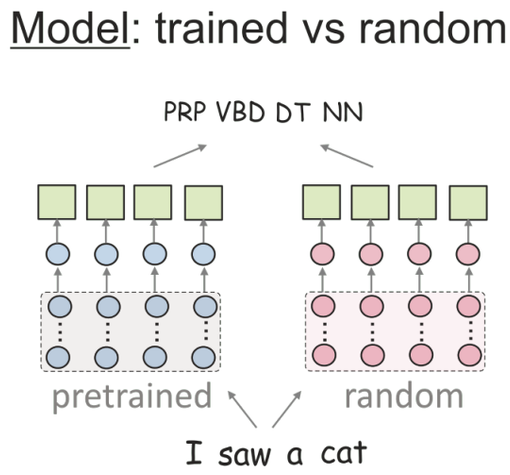
| Probe | PoS | Ctl | Select. | Dep | Ctl | Select. |
|---|---|---|---|---|---|---|
| Probes with Default Hyperparameters | | | | | | |
| Linear | 97.2 | 71.2 | 26.0 | - | - | - |
| Bilinear | - | - | - | 89.0 | 82.4 | 6.6 |
| MLP-1 | 97.3 | 92.8 | 4.5 | 92.3 | 93.0 | -0.7 |
| MLP-2 | 97.3 | 93.2 | 4.2 | 93.9 | 92.0 | 1.9 |
| Probes with 0.4 Dropout | | | | | | |
| Linear | 97.1 | 67.3 | 29.8 | - | - | - |
| Bilinear | - | - | - | 90.4 | 73.7 | 16.7 |
| MLP-1 | 97.5 | 93.4 | 4.1 | 93.8 | 93.1 | 0.7 |
| MLP-2 | 97.4 | 94.1 | 3.4 | 94.7 | 93.5 | 1.3 |
| Probes Designed with Control Tasks | | | | | | |
| Linear | 97.0 | 64.0 | 33.0 | - | - | - |
| Bilinear | - | - | - | 91.0 | 83.1 | 7.9 |
| MLP-1 | 97.2 | 80.6 | 16.6 | 90.5 | 84.3 | 6.2 |
| MLP-2 | 97.2 | 81.7 | 15.4 | 92.8 | 89.8 | 3.0 |

Hewitt, John, and Percy Liang. "Designing and Interpreting Probes with Control Tasks." arXiv preprint arXiv:1909.03368 (2019).

If a random model can solve the linguistic task equally well, what can we say about the model's quality?



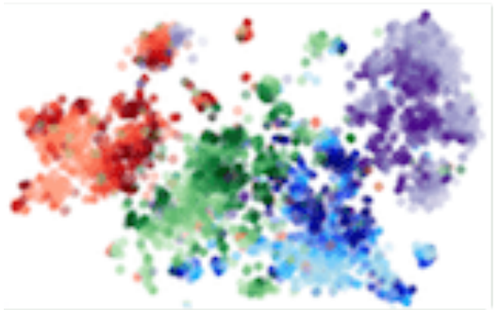
If models solve the random task equally well, is there anything we can say about how well the model represents linguistic features?

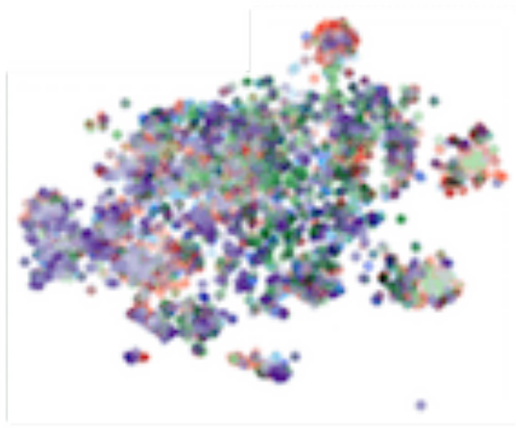


How can we capture how **hard** it is for the probe to learn the task?

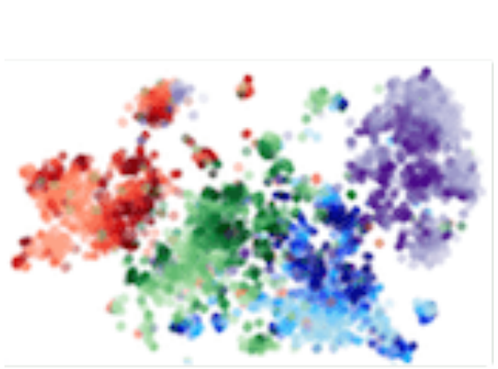
$$\mathcal{D} = \{((x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$$

Representations $x_{1:n}$ and labels $y_{1:n}$.

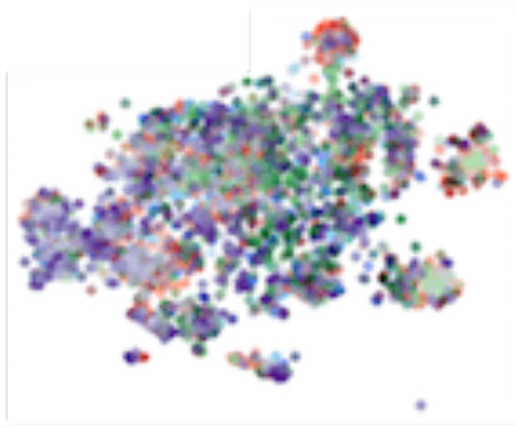




## Strong Regularity

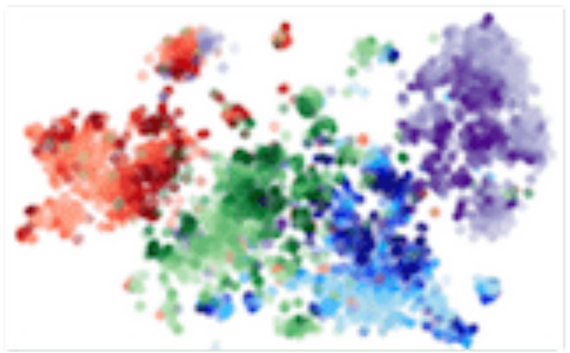


## Weak Regularity



https://lena-voita.github.io/posts/mdl_probes.html

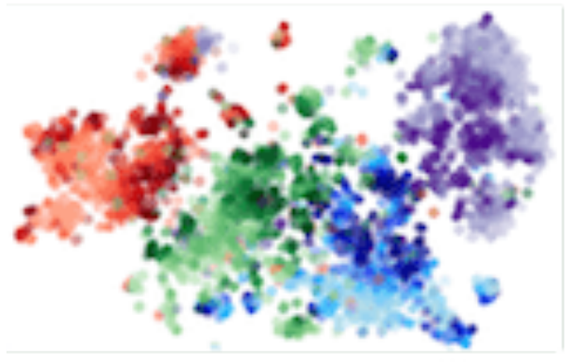
strong regularity



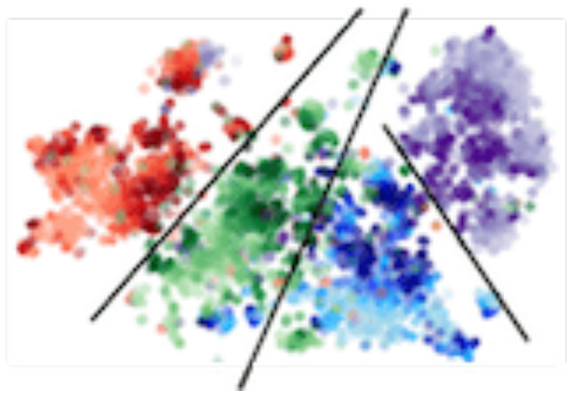
can be revealed
with a few examples

https://lena-voita.github.io/posts/mdl_probes.html

strong regularity



can be explained
with a simple “rule”



https://lena-voita.github.io/posts/mdl_probes.html

$$\mathcal{L}_{\text{Minimum Description Length}} = \mathcal{L}_{\text{Model}} + \mathcal{L}_{\text{Data}}$$

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}\left(y_{t_i+1:t_{i+1}} \middle| x_{t_i+1:t_{i+1}}\right)$$

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

1. Send first block with a uniform code.

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

1. Send first block with a uniform code.
2. Learn model on the sent block(s).

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

1. Send first block with a uniform code.
2. Learn model on the sent block(s).
3. Use model to communicate next block (in a shorter way compared to uniform code.)

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

1. Send first block with a uniform code.
2. Learn model on the sent block(s).
3. Use model to communicate next block (in a shorter way compared to uniform code.)
4. Repeat until all data is sent.

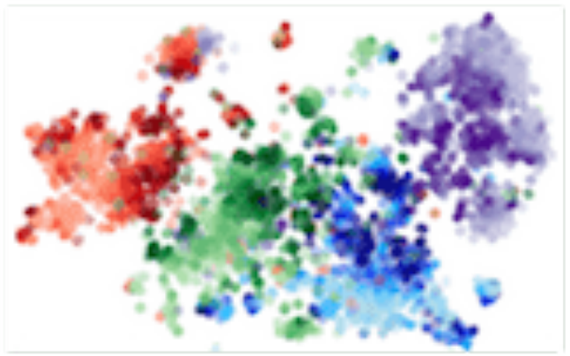
Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

Rissanen, Jorma. "Universal coding, information, prediction, and estimation." *IEEE Transactions on Information theory* 30.4 (1984): 629-636.

$$\mathcal{L}_{\text{Online}} = t_1 \log_2 K - \sum_{i=1}^{S-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}} | x_{t_i+1:t_{i+1}})$$

strong regularity

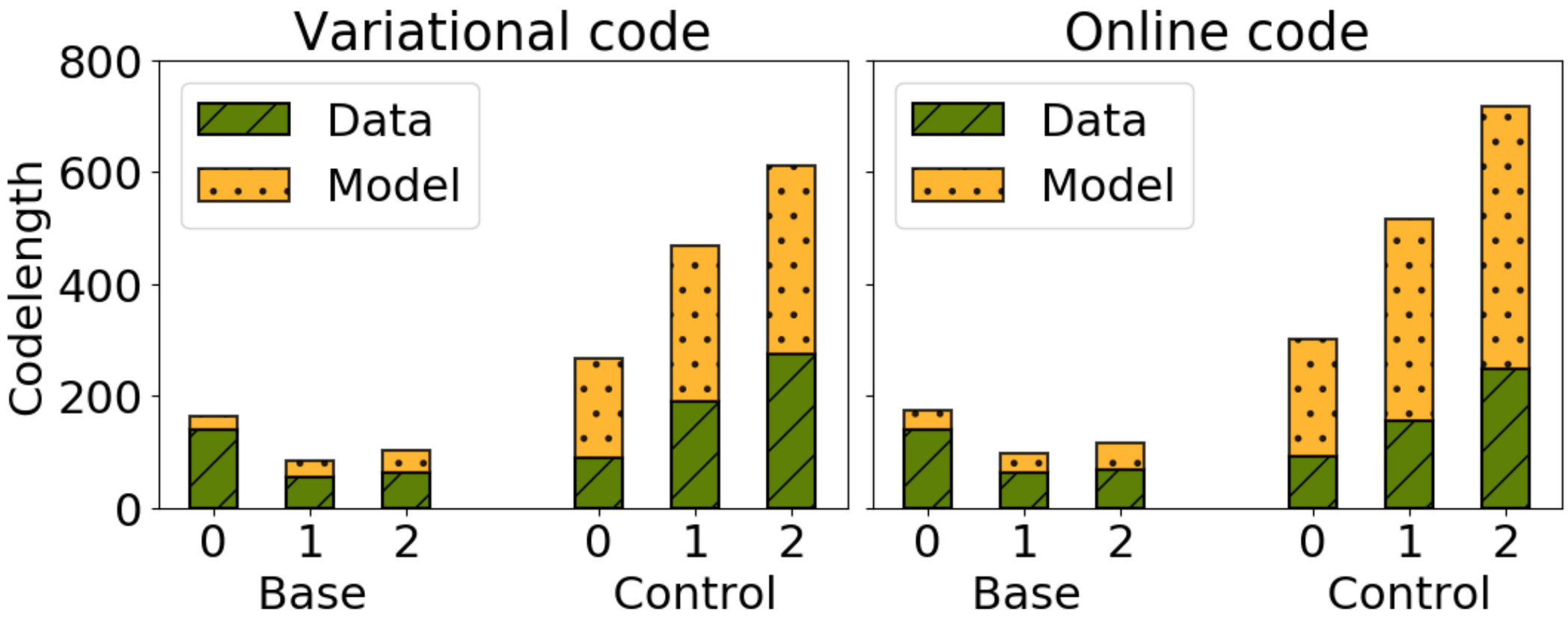


can be revealed with a few examples

https://lena-voita.github.io/posts/mdl_probes.html

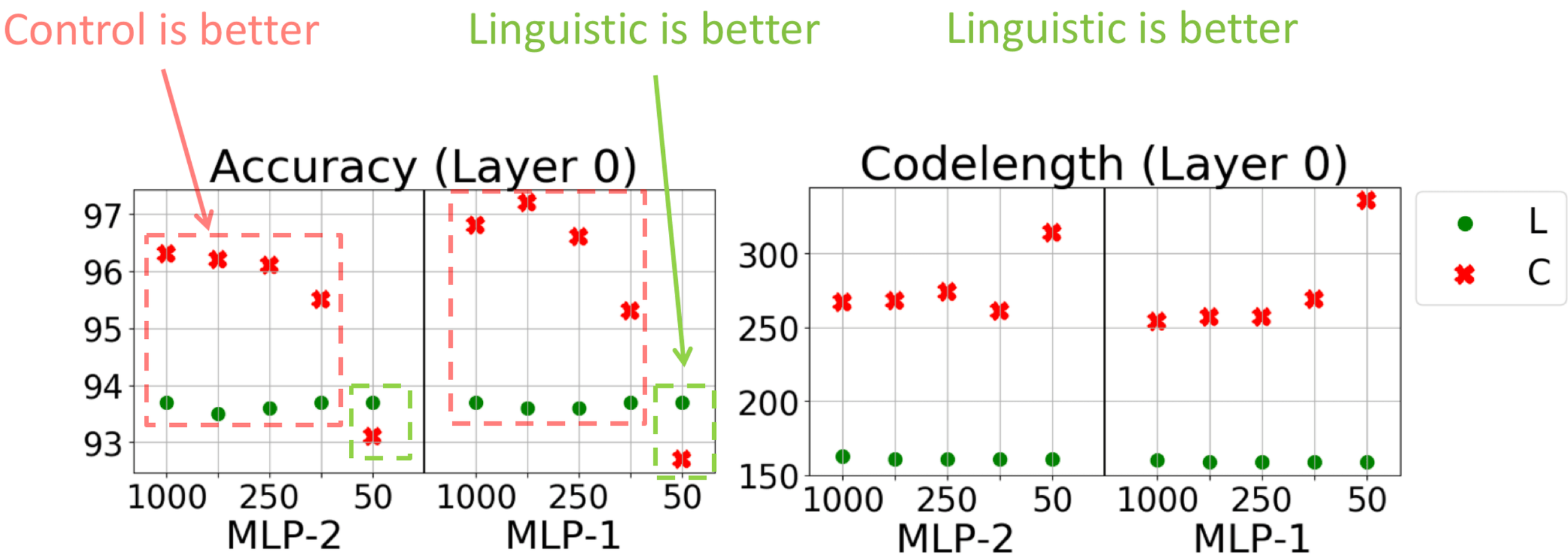
# Results

# Variational & Online Codes Report Similar Behavior




https://lena-voita.github.io/posts/mdl_probes.html

# Across HPs, Accuracy Unclear; MDL Clear

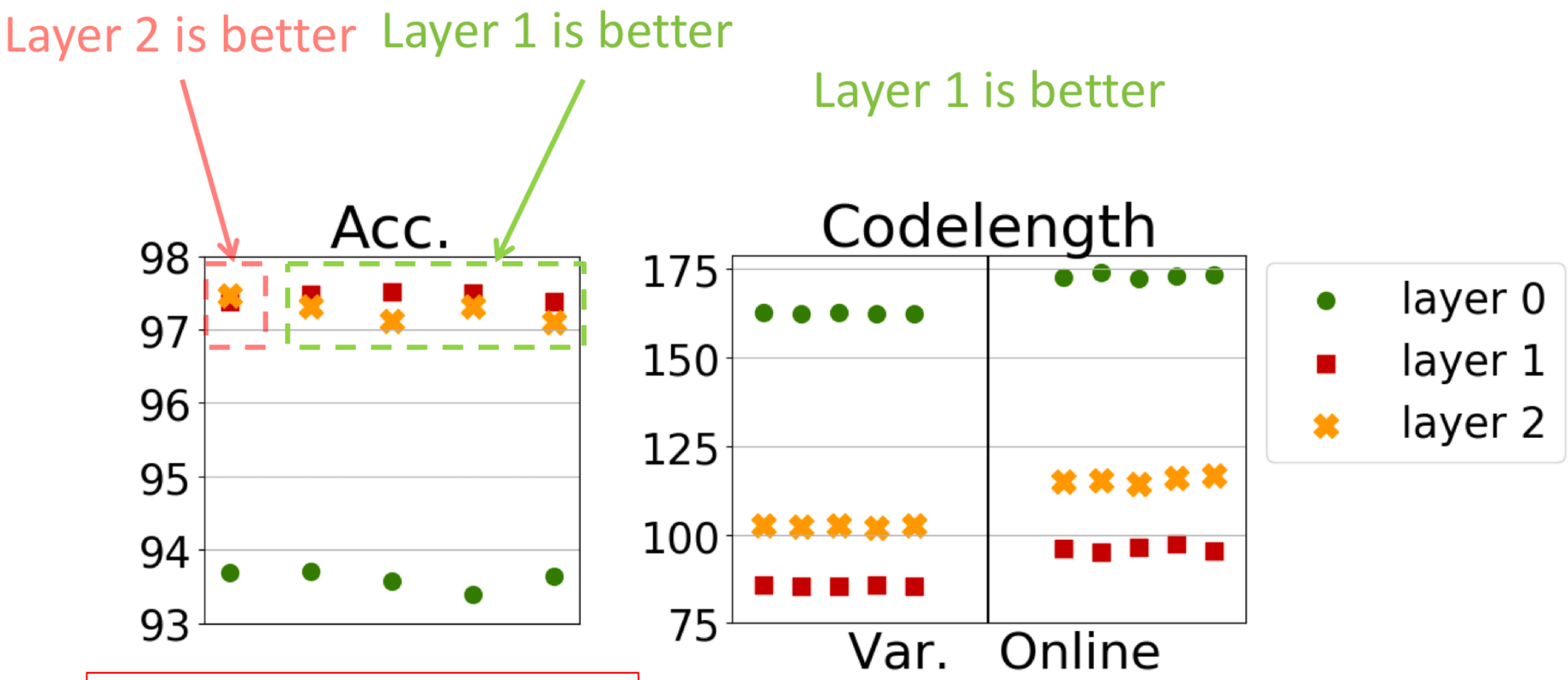



Accuracy is wrong for 8 out of 10 settings, MDL is always correct

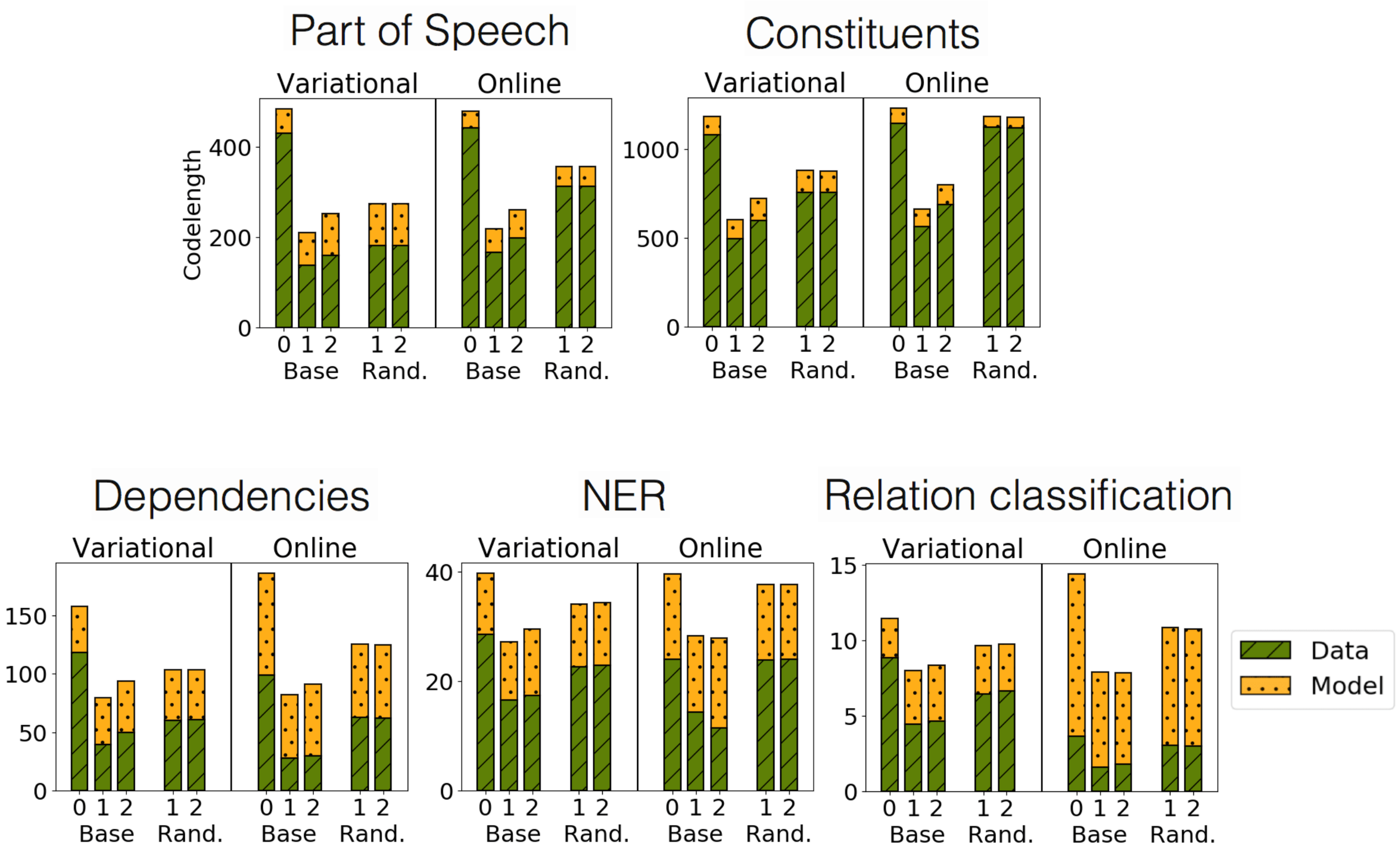
(for accuracy higher is better, for codelength – lower)

https://lena-voita.github.io/posts/mdl_probes.html

# Across Random Seeds, Acc Unstable; MDL Stable




https://lena-voita.github.io/posts/mdl_probes.html


Part of Speech
Constituents
Dependencies
NER
Relation classification
Variational
Online
Codelength
0
200
400
500
1000
50
100
150
20
40
5
10
15
0 1 2
1 2
Base
Rand.
Data
Model

# Author Take-Aways

1. Layer 0 vs. Contextual: Even random contextual is better.
2. Code-lengths for randomly initialized models are higher.
3. Randomly initialized layers don't evolve.

# My Take-Aways

1. Seems better than accuracy.
2. Seems not that much better and requires accuracy for context.
    1. The HPs that "failed" for accuracy were very small.
    2. The separation across random seeds for MDL shows a difference, but we can't readily interpret that difference.