Outliers with Opposing Signals Have an Outsized Effect on Neural Network Optimization

Elan Rosenfeld

Andrej Risteski

Deep Learning: Classics and Trends
1/26/2024

Slides template: Jeremy Bernstein

Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

- Grokking

The Slingshot Mechanism: An Empirical Study of Adaptive Optimizers and the *Grokking Phenomenon*

Vimal Thilak vthilak@apple.com
Etai Littwin elittwin@apple.com
Shuangfei Zhai szhai@apple.com
Omid Saremi osaremi@apple.com
Roni Paiss rpaiss@apple.com
Joshua Susskind jsusskind@apple.com

Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit

Boaz Barak Harvard University
Benjamin L. Edelman Harvard University
Surbhi Goel Microsoft Research & University of Pennsylvania
Sham Kakade Harvard University
Eran Malach Hebrew University of Jerusalem
Cyril Zhang Microsoft Research

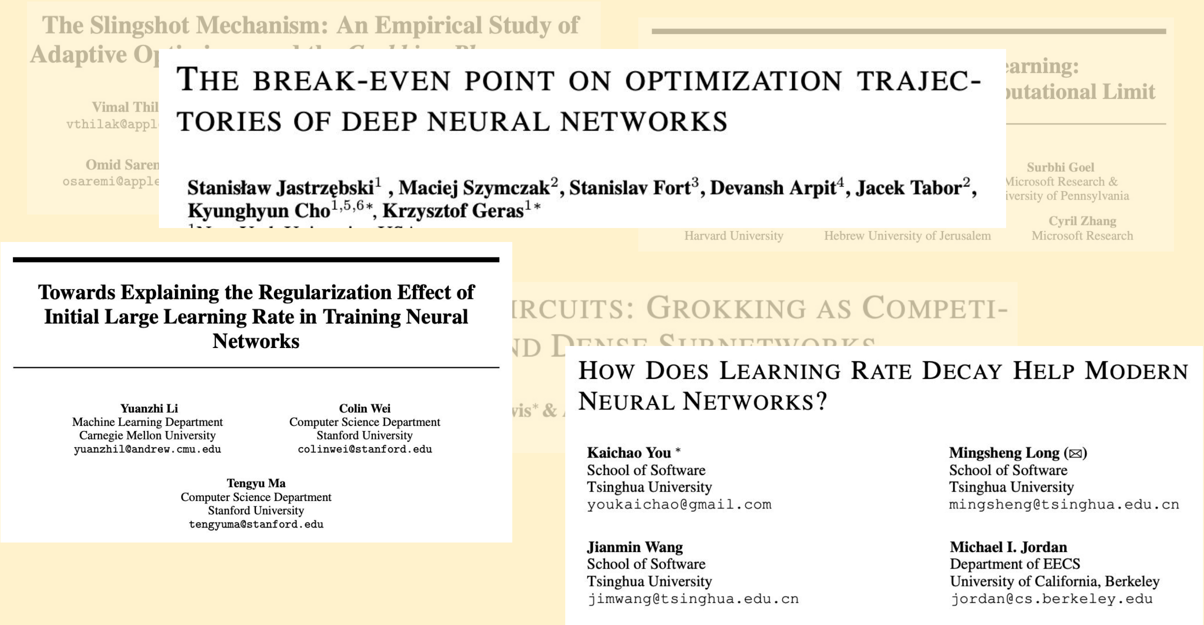
A TALE OF TWO CIRCUITS: GROKKING AS COMPETITION OF SPARSE AND DENSE SUBNETWORKS

William Merrill*, Nikolaos Tsilivis* & Aman Shukla
New York University

Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

- Grokking
- Benefits of Large LR



THE BREAK-EVEN POINT ON OPTIMIZATION TRAJECTORIES OF DEEP NEURAL NETWORKS

Stanisław Jastrzębski[1], Maciej Szymczak[2], Stanislav Fort[3], Devansh Arpit[4], Jacek Tabor[2], Kyunghyun Cho[1,5,6*], Krzysztof Geras[1*]

Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks

Yuanzhi Li, Machine Learning Department, Carnegie Mellon University, yuanzhil@andrew.cmu.edu

Colin Wei, Computer Science Department, Stanford University, colinwei@stanford.edu

Tengyu Ma, Computer Science Department, Stanford University, tengyuma@stanford.edu

HOW DOES LEARNING RATE DECAY HELP MODERN NEURAL NETWORKS?

Kaichao You *, School of Software, Tsinghua University, youkaichao@gmail.com

Mingsheng Long (✉), School of Software, Tsinghua University, mingsheng@tsinghua.edu.cn

Jianmin Wang, School of Software, Tsinghua University, jimwang@tsinghua.edu.cn

Michael I. Jordan, Department of EECS, University of California, Berkeley, jordan@cs.berkeley.edu

Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

- Grokking
- Benefits of Large LR
- Batchnorm

Partially Understood Phenomena Abound in NN Optimization

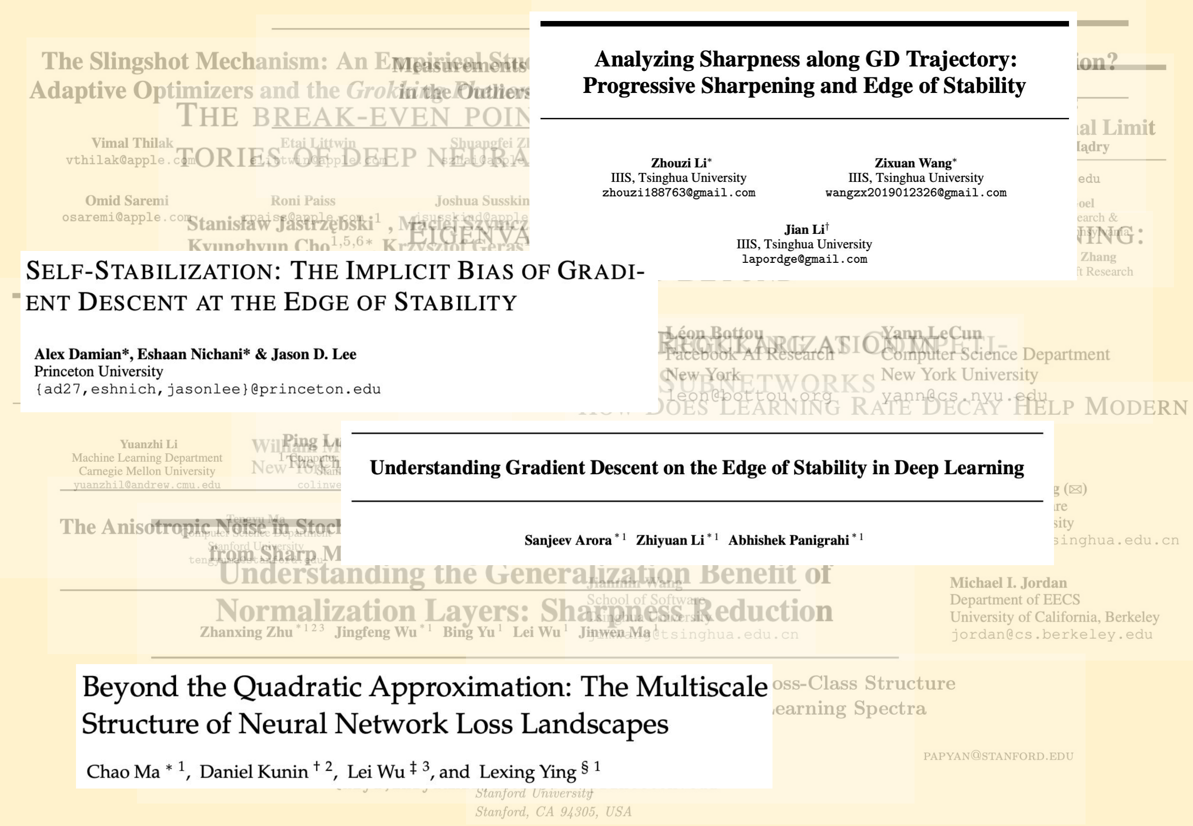
An incomplete list:

- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers

Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

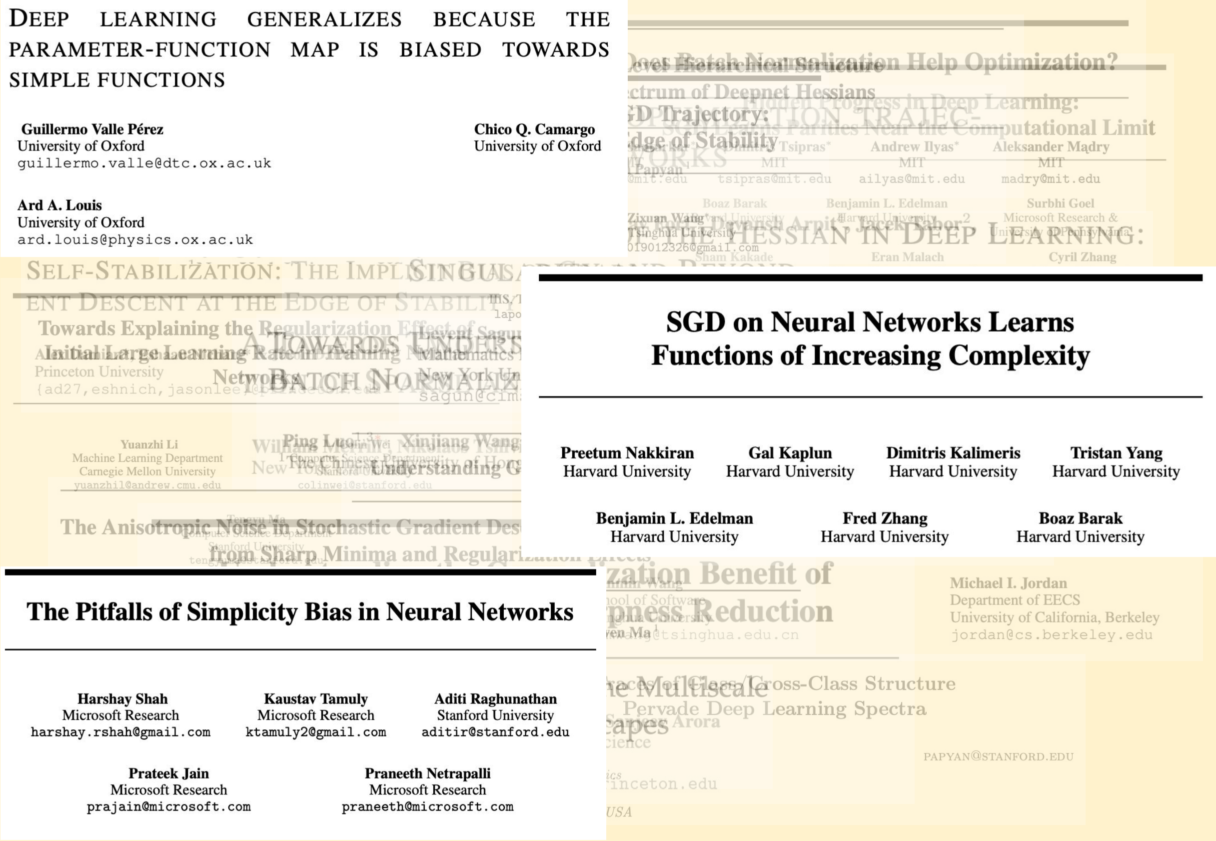
- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers
- Sharpening/EoS



Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

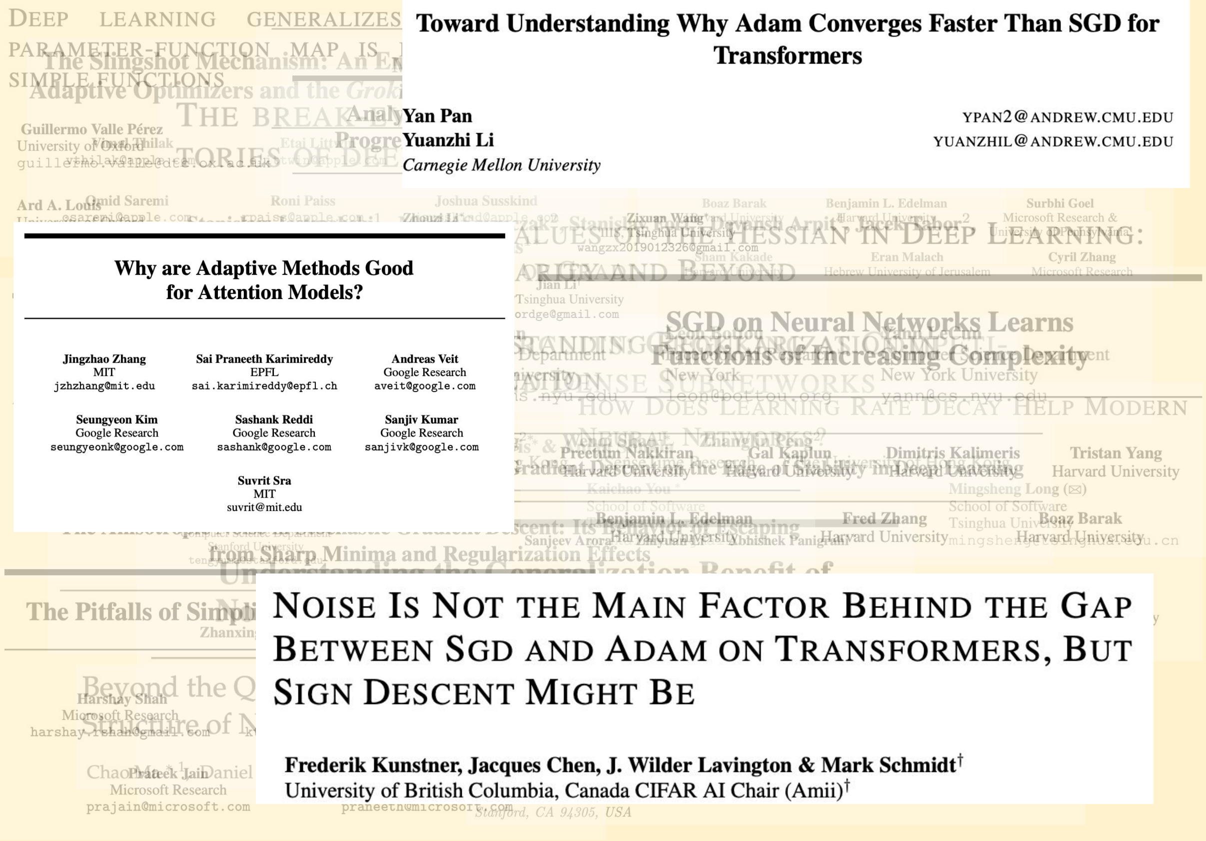
- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers
- Sharpening/EoS
- Simplicity Bias



Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

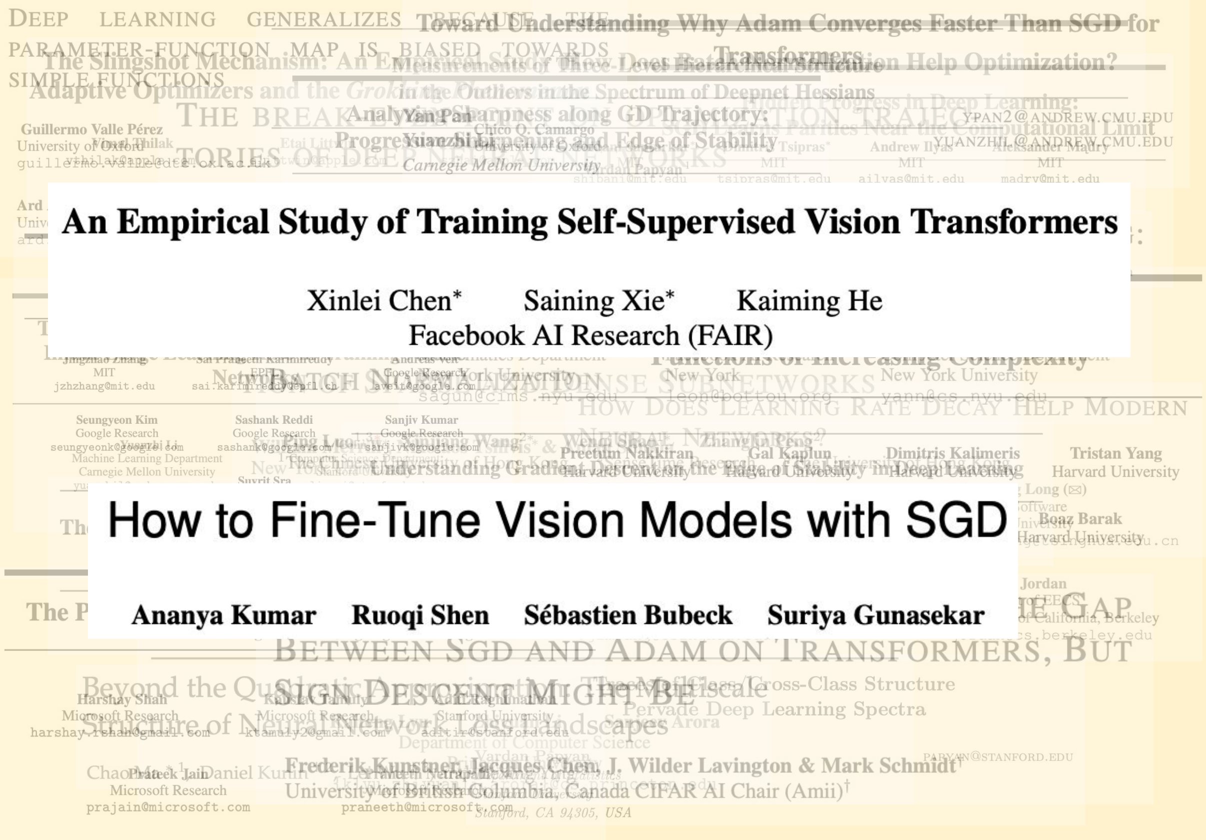
- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers
- Sharpening/EoS
- Simplicity Bias
- Adaptive Methods



Partially Understood Phenomena Abound in NN Optimization

An incomplete list:

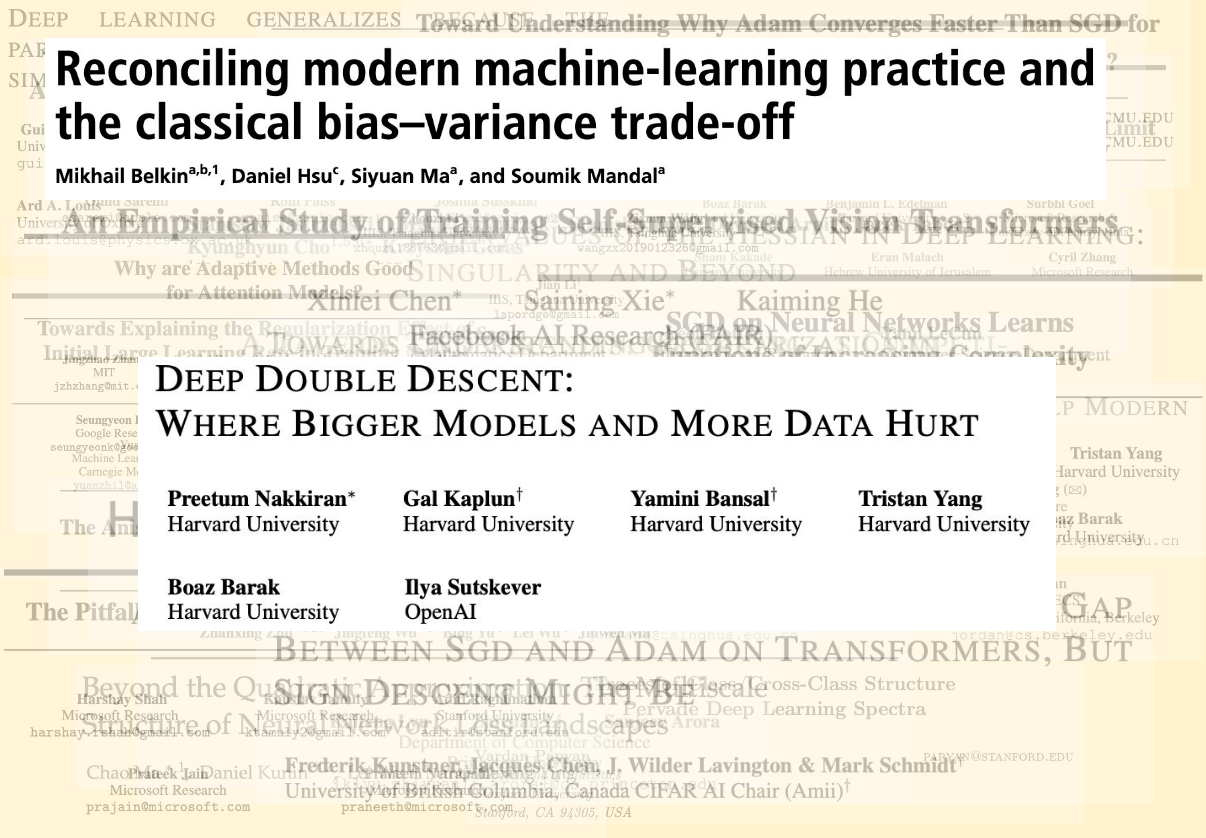
- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers
- Sharpening/EoS
- Simplicity Bias
- Adaptive Methods
- Unstable Training



Partially Understood Phenomena Abound in NN Optimization

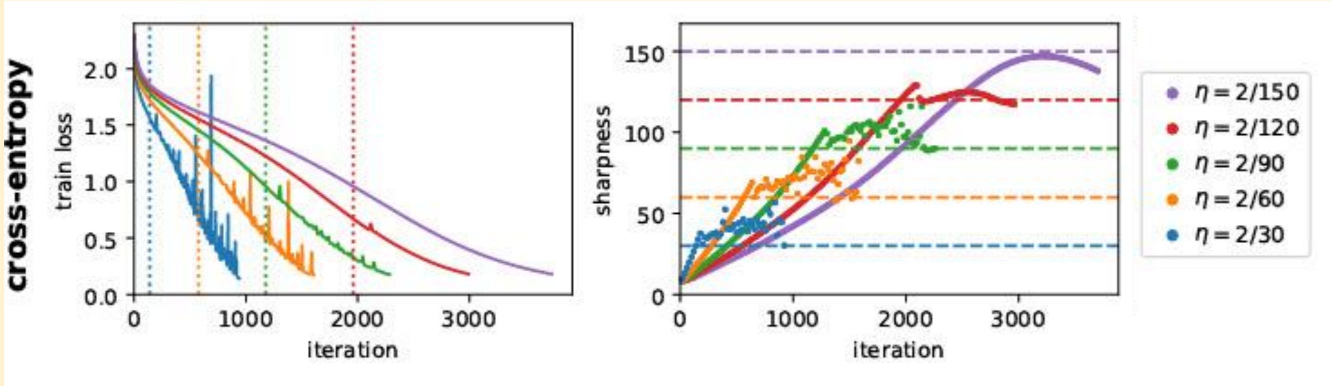
An incomplete list:

- Grokking
- Benefits of Large LR
- Batchnorm
- Hessian Spectrum Outliers
- Sharpening/EoS
- Simplicity Bias
- Adaptive Methods
- Unstable Training
- Double Descent



Surely, *some* of these results are related... but unclear how.

Progressive Sharpening + Edge of Stability

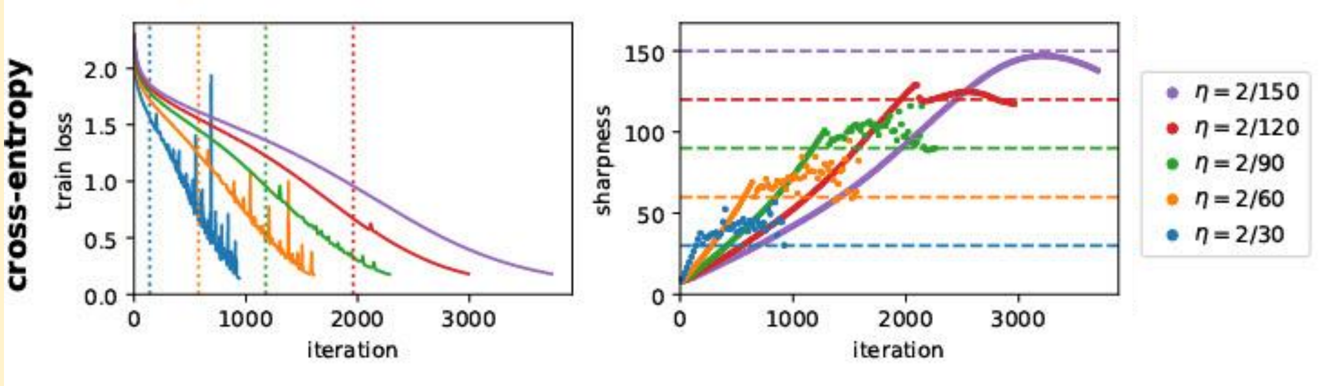


Meanwhile, loss decreases non-monotonically, with frequent "spikes".

"Sharpness" = top eigenvalue of loss Hessian
First rises to $2/\eta$...
Then hovers around that value.

[1] Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. Cohen et al. 2020.

Progressive Sharpening + Edge of Stability



This is just more evidence that **something more is needed** to understand NN training dynamics...

[1] Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. Cohen et al. 2020.

Yet Another Phenomenon

I'm going to present our finding:
another interesting phenomenon in neural network optimization.

But the goal is not just to add to the growing list.

Instead, we hope it can help explain and unify
these observations via a **shared underlying cause**.*

Yet Another Phenomenon

(We also look at SGD)

Let's run the following experiment:

1. Train a neural network with full-batch gradient descent on CIFAR-10.
2. Track losses on each training point *individually*.
3. Fix some iteration T.
4. Calculate changes in loss on each point from step T to step T+1.
5. Visualize the samples with the *most positive* and *most negative* changes.

What should we expect to see?

Yet Another Phenomenon

VGG-11

Largest increase in loss

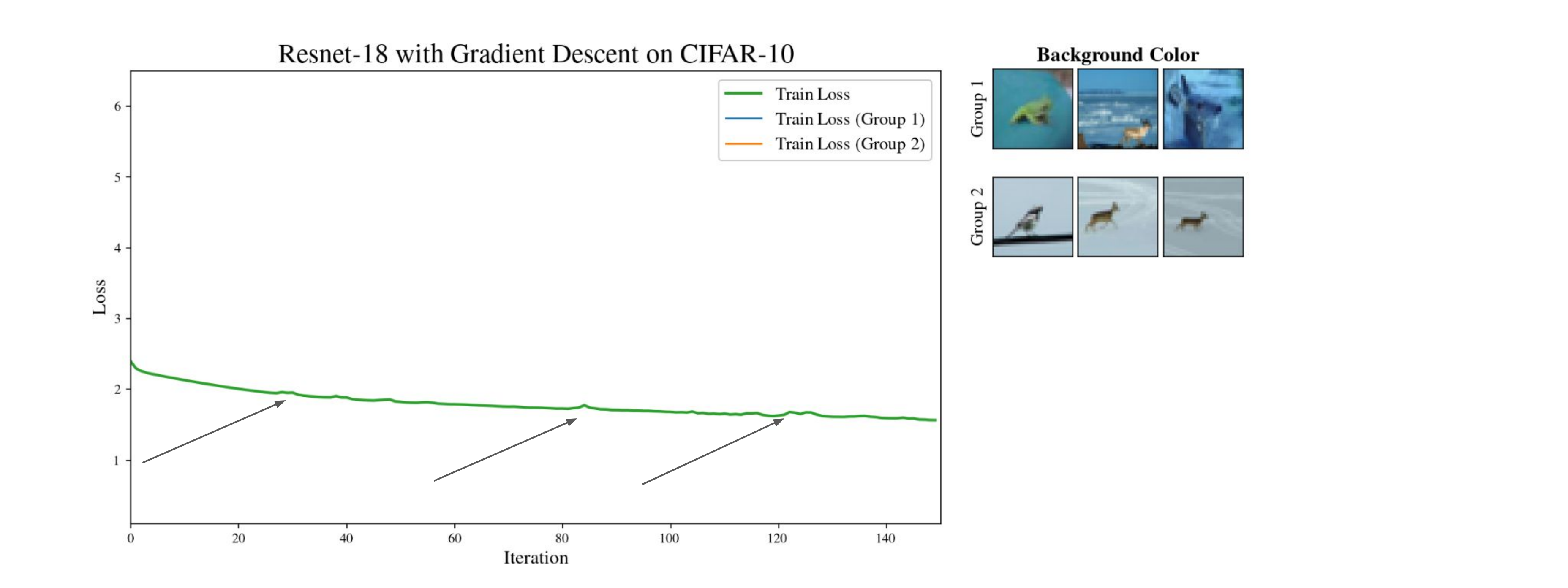
Largest decrease in loss

ResNet-18

The precise patterns change, but this occurs *all throughout training*.

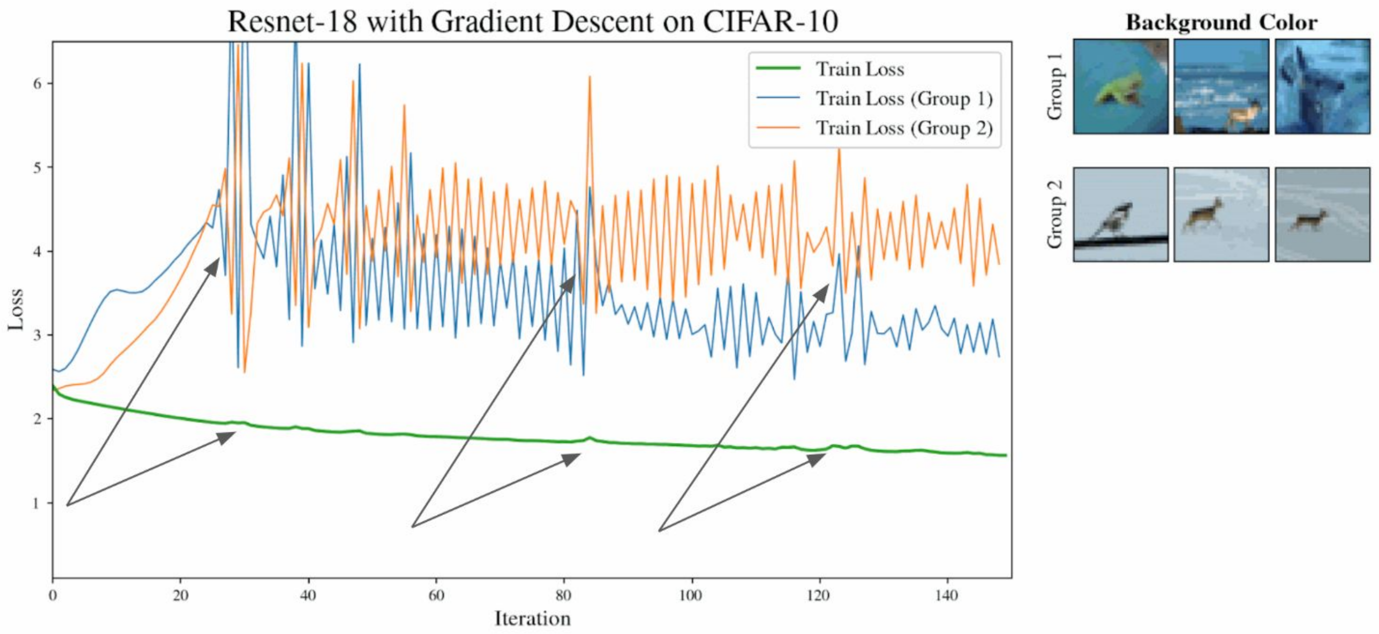
Visualizing the Group Losses

These groups are ~20 samples each.



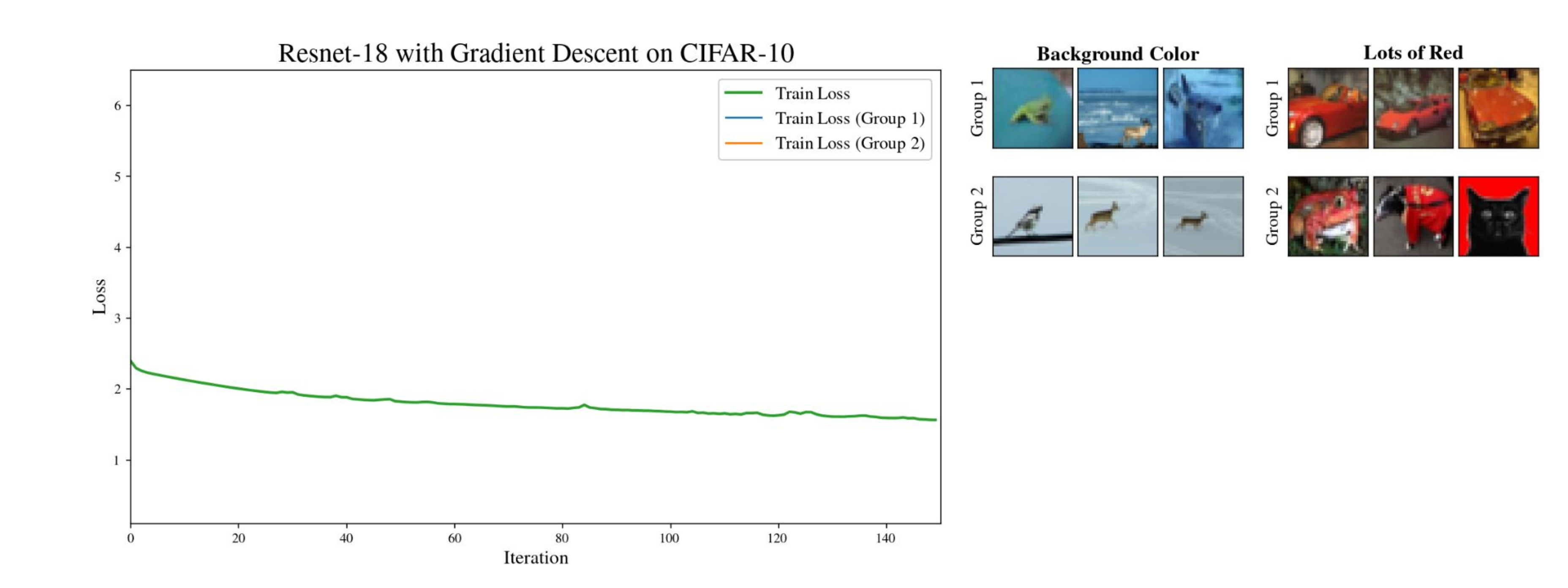
Samples were selected for largest change in loss, so we expect a “spike” somewhere.

Visualizing the Group Losses



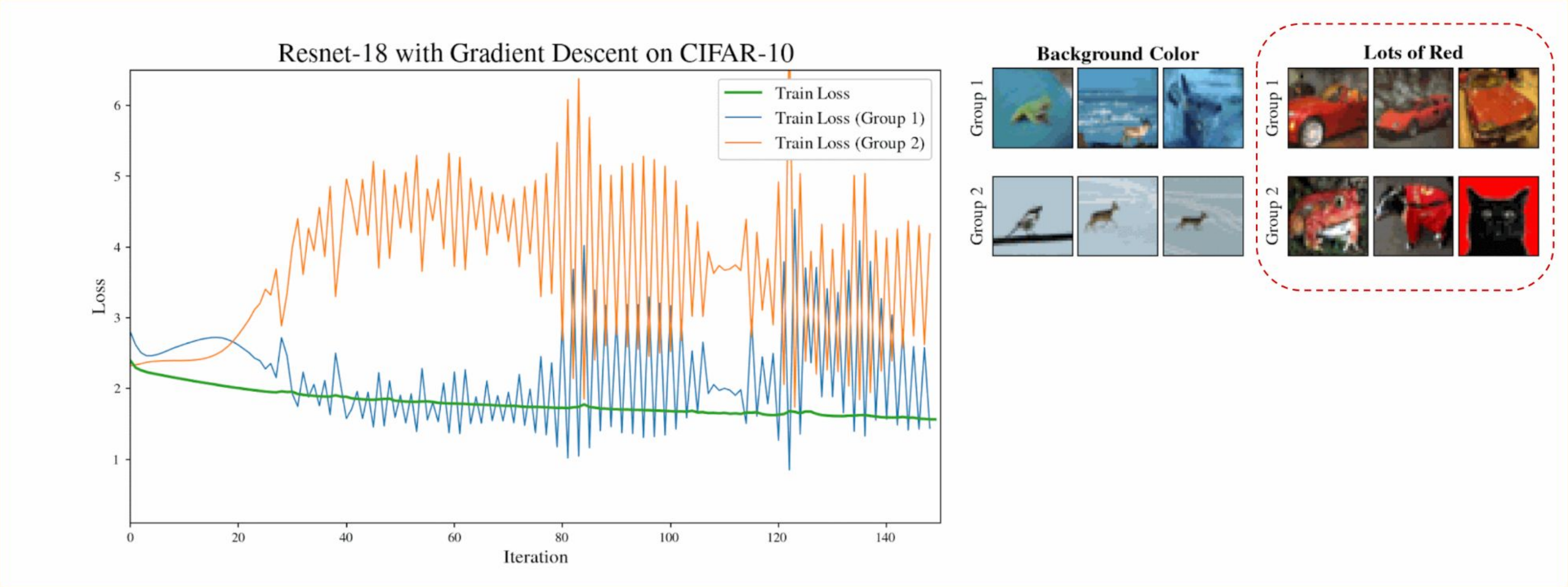
These opposing groups oscillate with large amplitude *continuously*!

Visualizing the Group Losses



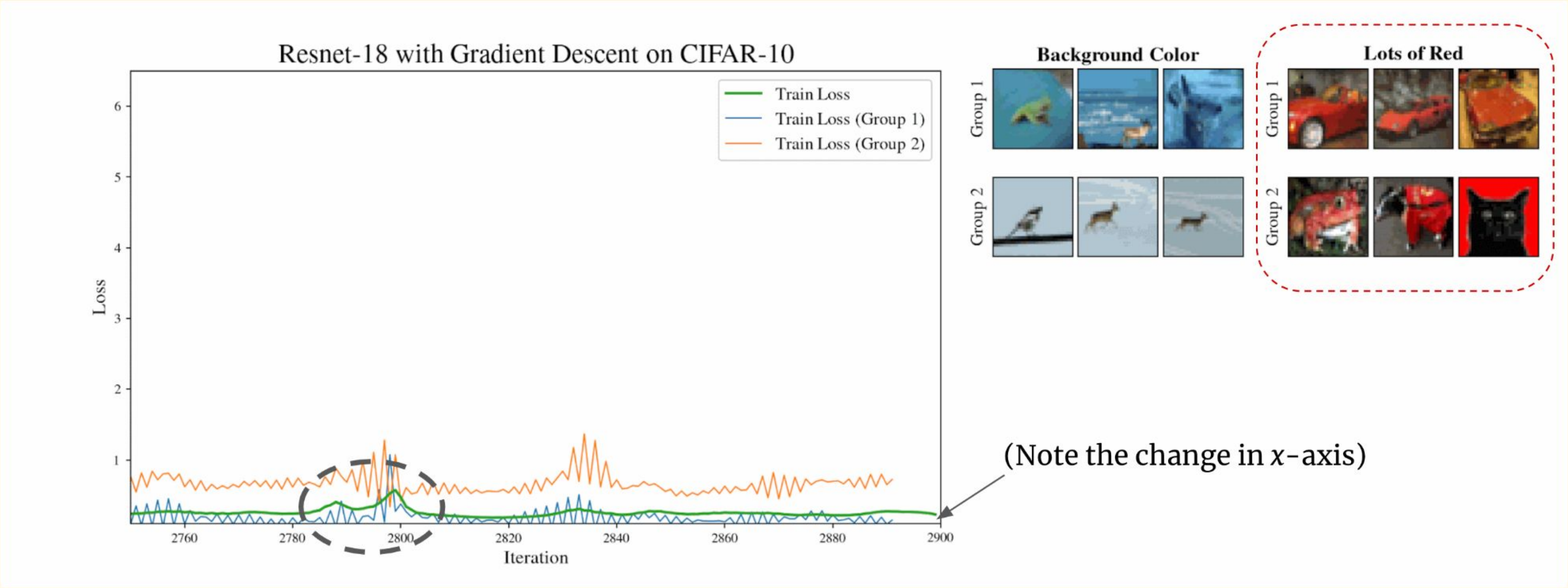
What about another group?

Visualizing the Group Losses



When we're close to interpolating, shouldn't this effect be reduced?

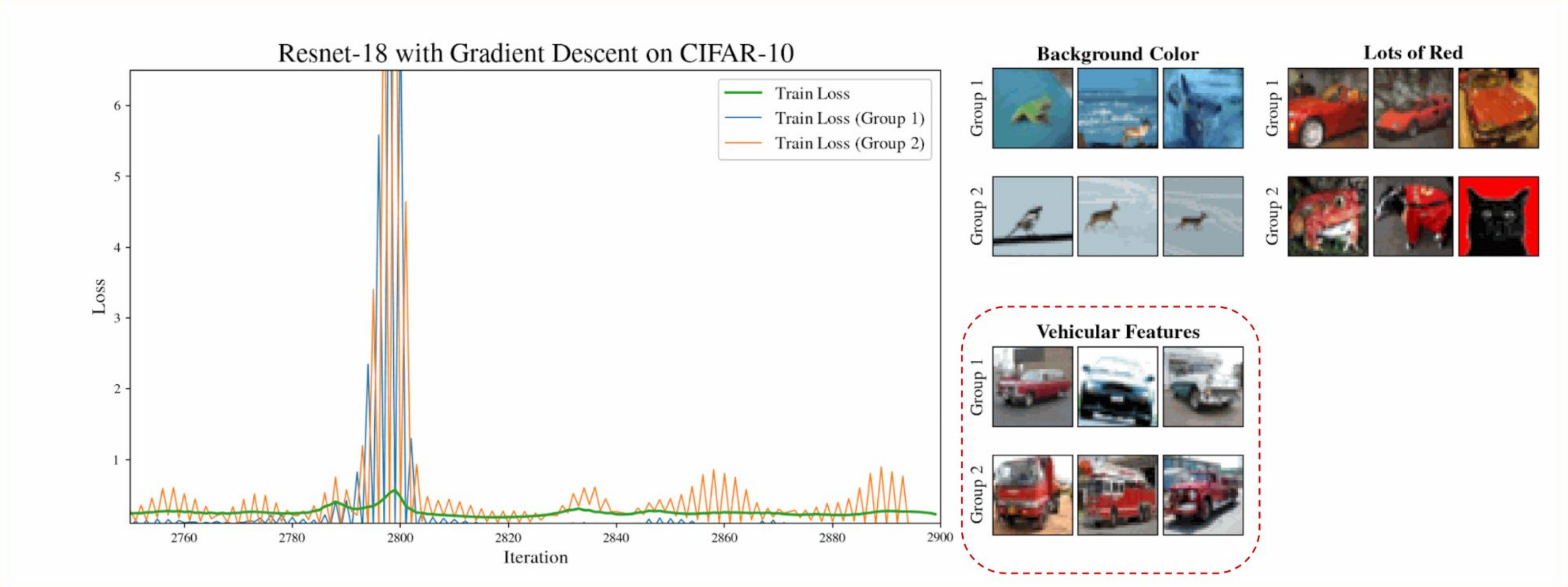
Visualizing the Group Losses



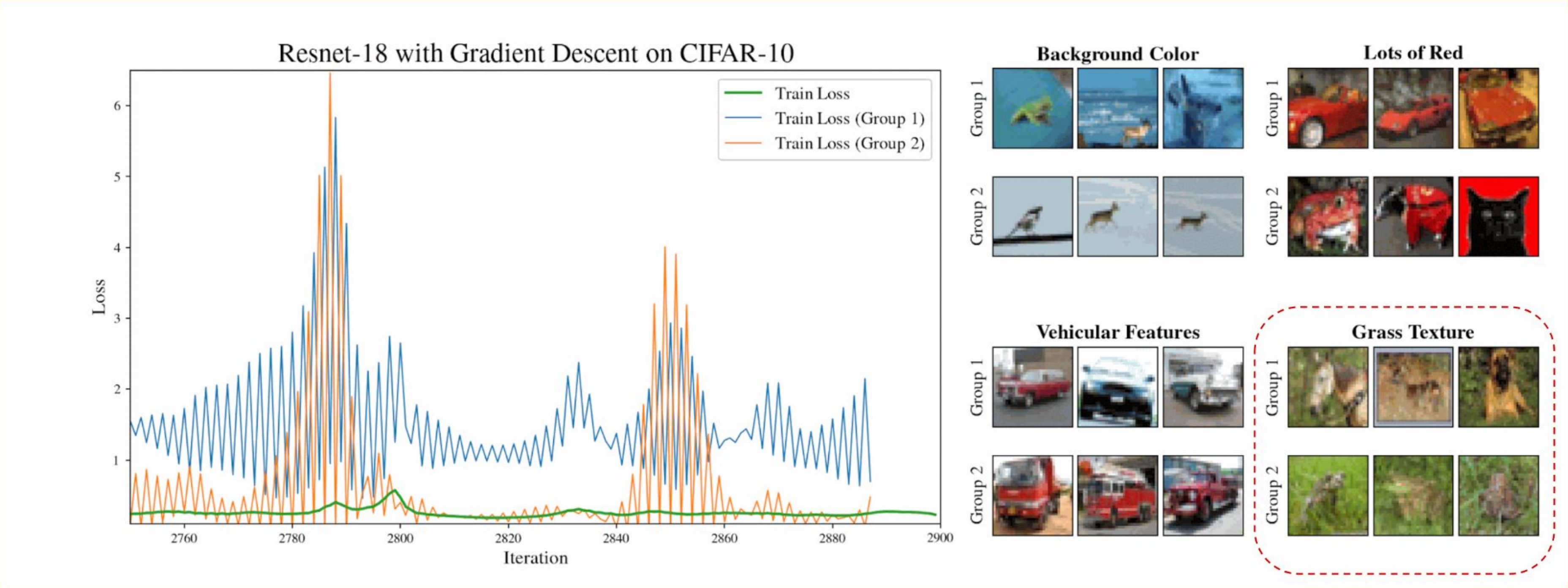
Yes, the amplitude is substantially smaller…

So what's causing these loss increases?

Visualizing the Group Losses



Visualizing the Group Losses



Even at the end stages of training, large loss swings are still occurring.

What's Going On?

- **Prevalent features**, often with distinct colors.
  - Roughly, "prevalent" ≈ "fills a lot of the image"
- **Begin simple**, become progressively more complex.
  - "Simple" ≈ "available at random initialization"
- Large gradients pointing in **opposite directions**.
  - Learning "red = car" decreases loss on red cars, increases loss on red *non*-cars

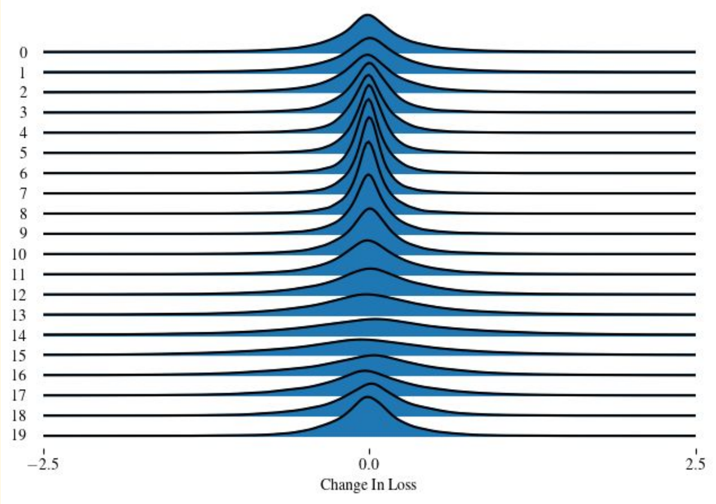
We call these features—or the gradients they induce—***Opposing Signals***.

ChatGPT's best alternative suggestion: ***Backprop Battle***

What's Going On?

Does this occur for *every* training sample?

Distribution of changes in loss:

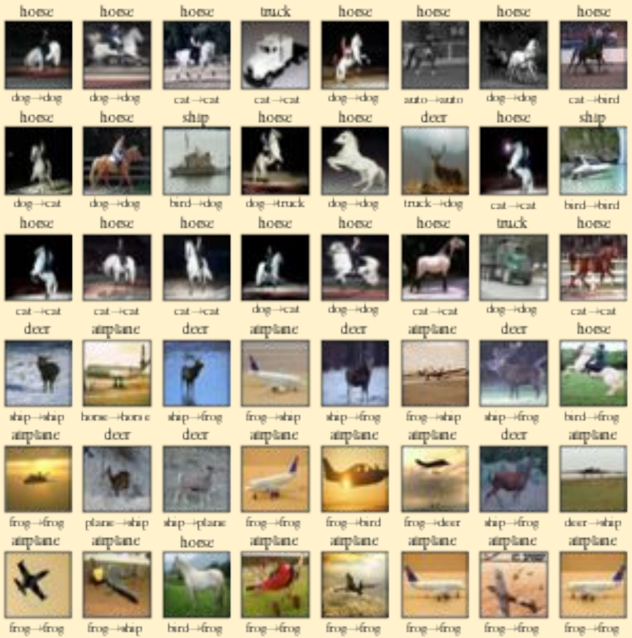


These samples are *significant* outliers.

What Causes Opposing Signals?

Is this a property of architecture (ConvNet)?

No. Same occurs in a Vision Transformer.



What Causes Opposing Signals?

Maybe it’s a property of the data modality (images)?

Also no.

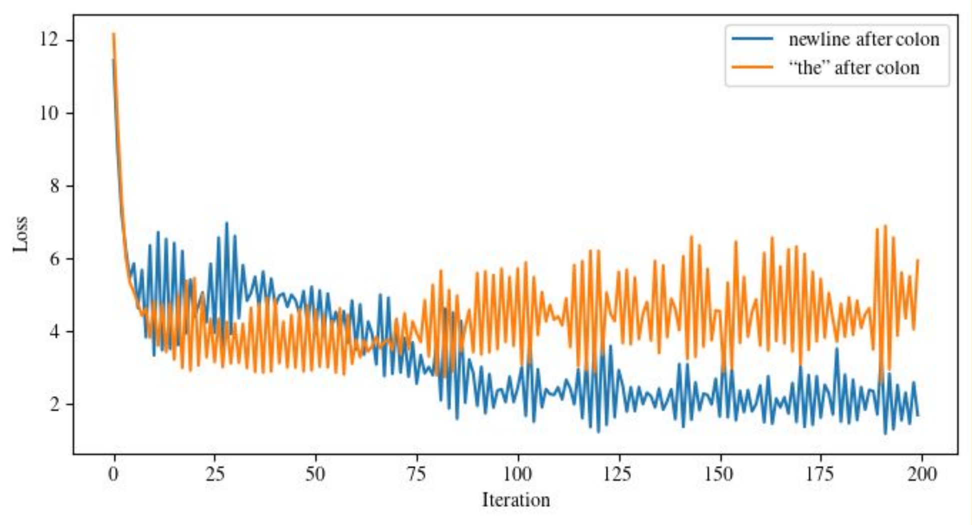
Group 1

```
Salcedo said of the work:[\n]
Enter your email address:[\n]
According to the CBO update:[\n]
Here’s how the Giants can still make the
playoffs:[\n]
in early 2018.\n\nAccording to the CBO update:[\n]
other than me being myself.”\n\nWATCH:[\n]
```

Group 2

```
MPs in Westminster. But to me it is obvious: [the]
The wheelset is the same as that on the model above:
[the]
all other acts of love, both divine and human: [the]
from the Kurds’ two main political parties: [the]
title of precisely what makes it so wonderful: [the]
you no doubt noticed something was missing: [the]
```

(bracket is next token)



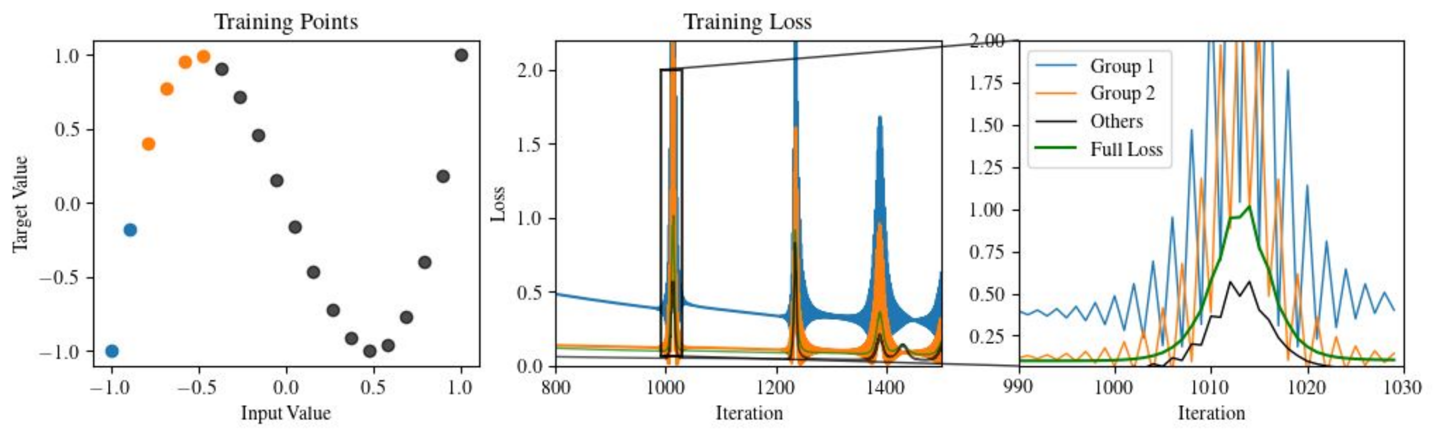
GPT-2 on OpenWebText

What Causes Opposing Signals?

Maybe it's a property of the data modality (images)?

Also no.

What about the loss (cross-entropy)?



What Causes Opposing Signals?

Remainder of this talk gives our current best understanding, with experiments.

We believe it a consequence of *depth* and *steepest descent*.

We don't fully understand the mechanism here.

- If there are parts you think aren't fully explained, you're right.
- If there are parts you think are *flat out wrong*, you could be right.

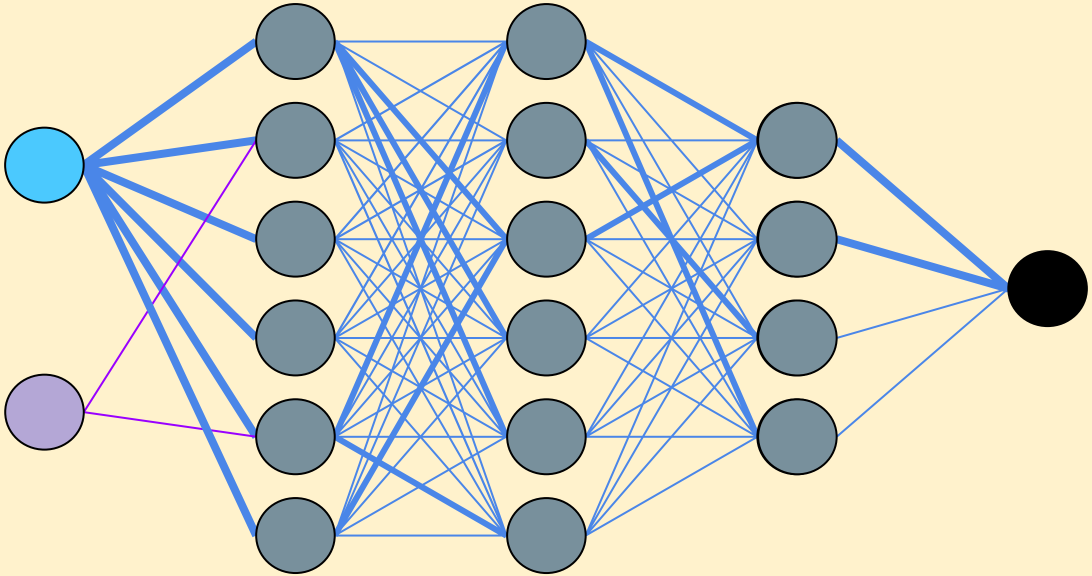
However:

- We have a reasonably descriptive high-level story…
  - and we prove this behavior for a simple model on a 2-layer linear net.*
- It enables *specific* qualitative predictions which we then verify…
  - and it naturally fits into several existing narratives of other phenomena.

A Simplified Story of Gradient Descent on Deep Neural Networks

Consider a randomly initialized MLP with two input features:

1. "Sky": large magnitude + pervasive (propagated to all neurons).
   - Only sufficient for predicting $p($ class | "sky" $)$.
2. "Shape": small magnitude, needs to be learned.
   - But much more useful for loss reduction.

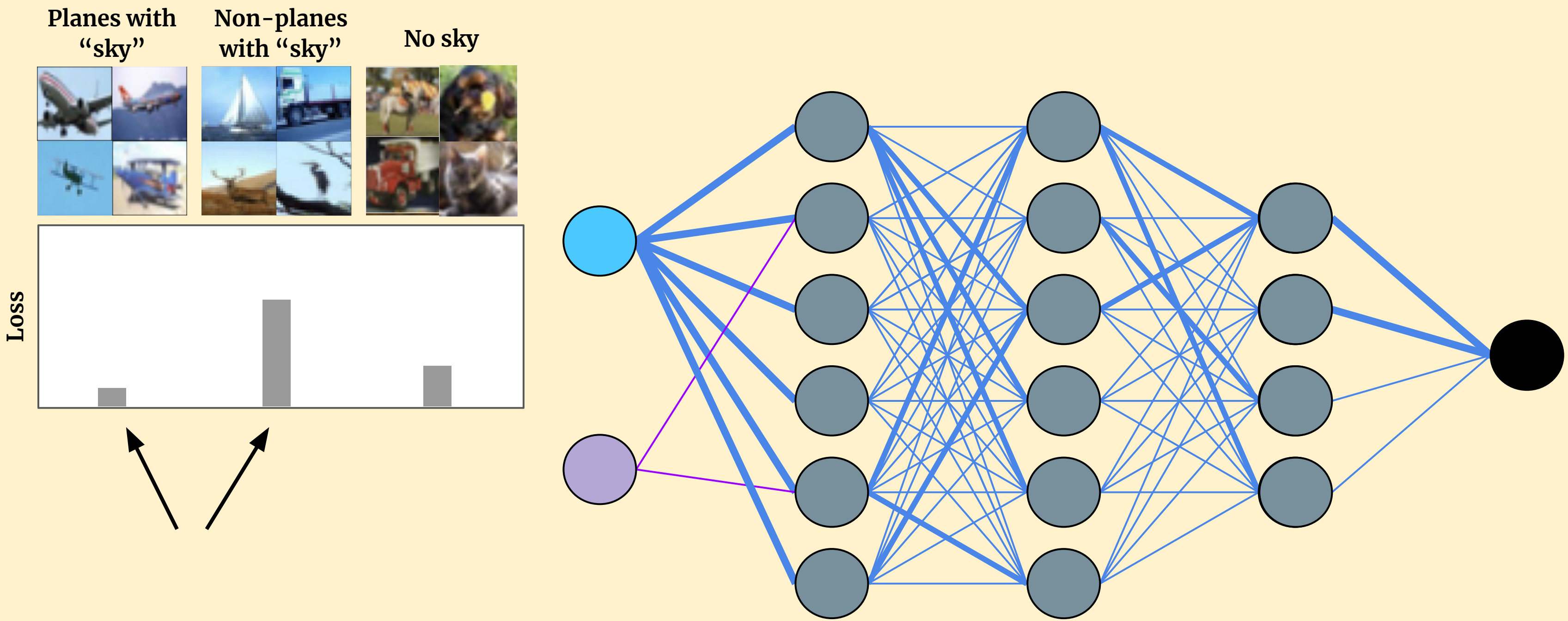


At initialization, network activations are dominated by "sky" on outliers.

- (Suppose network happens to predict "sky = plane")

High loss → large gradients → rebalance towards predicting p(class | "sky").

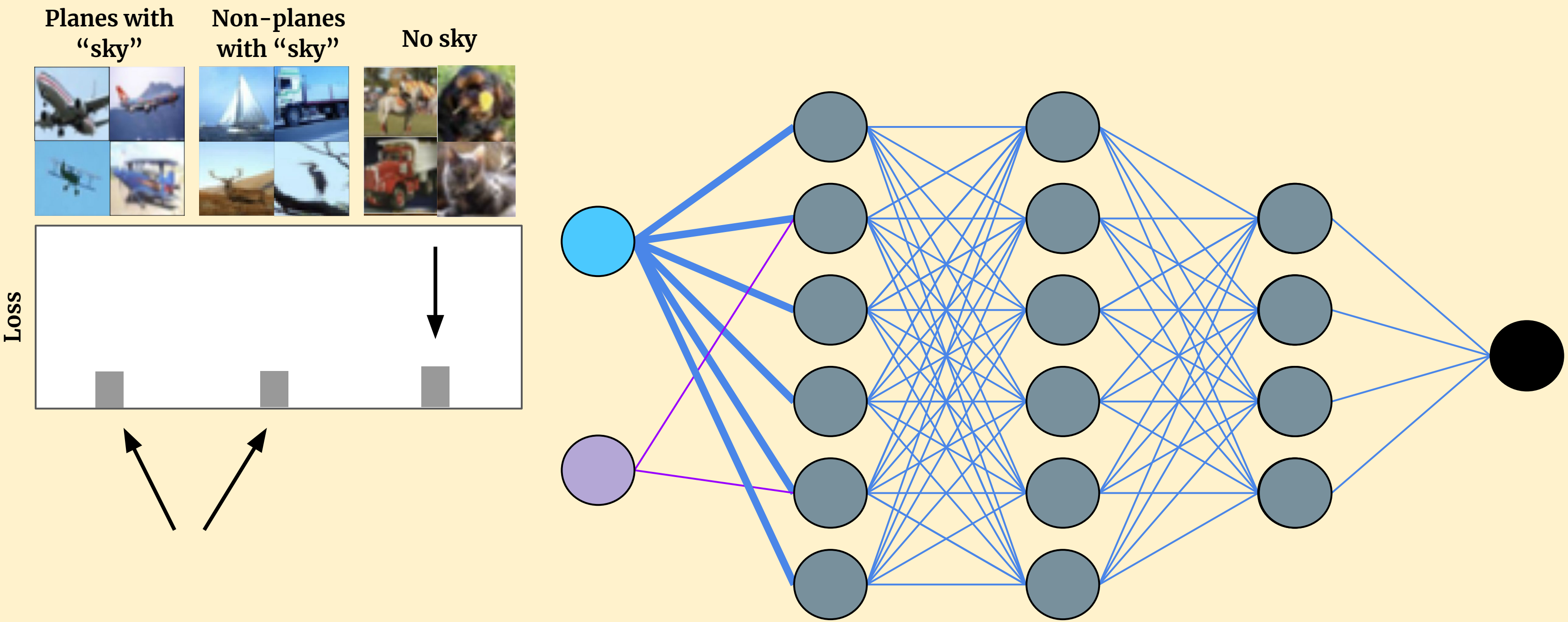
- (This "linear first" behavior has been previously observed[1, 2])



[1] SGD on Neural Networks Learns Functions of Increasing Complexity. Nakkiran et al. 2019.
[2] Do deep neural networks learn shallow learnable examples first? Mangalam and Prabhu 2019.

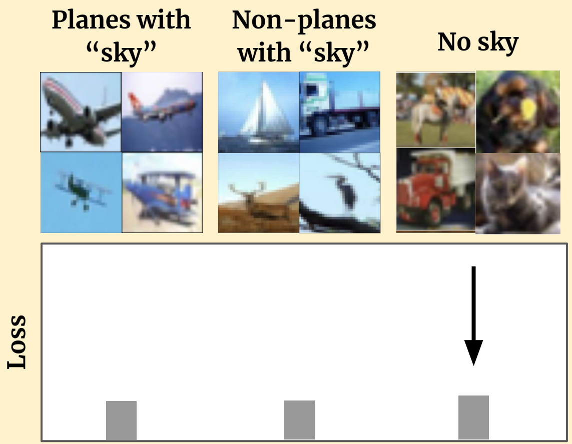
Once this happens, the network can now upweight the more useful "shape" feature.

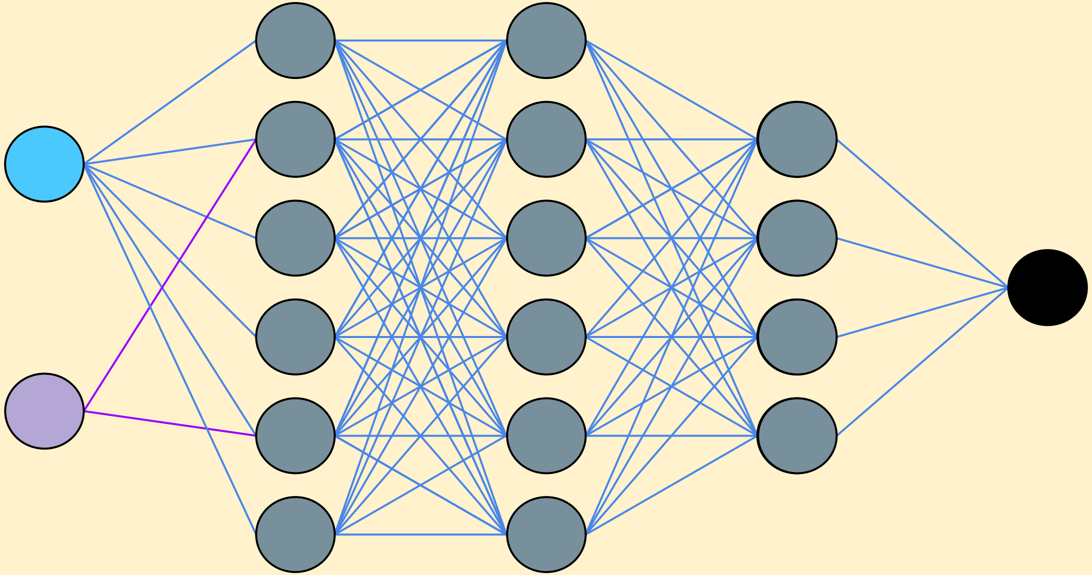
Since the outliers' loss no longer dominates the gradient, let's visualize a non-outlier.



Once this happens, the network can now upweight the more useful “shape” feature.

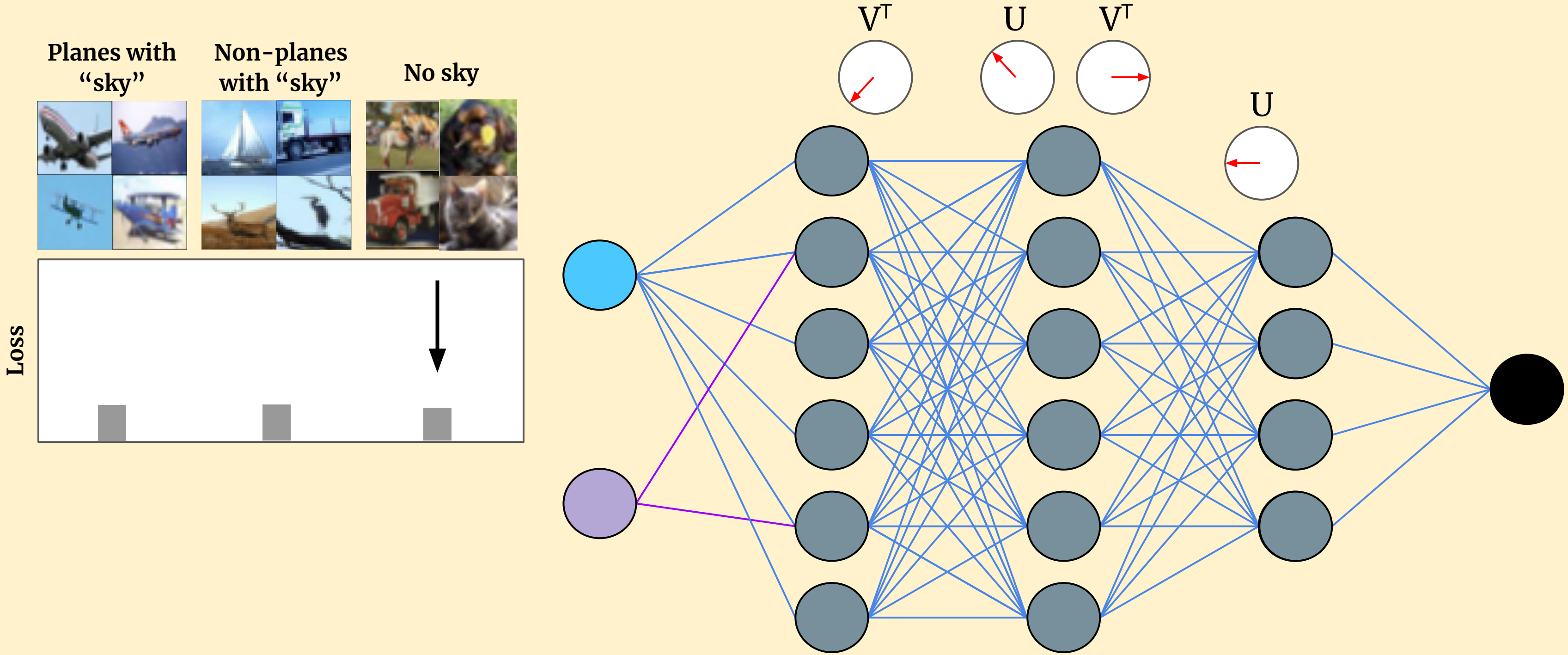
Since the outliers’ loss no longer dominates the gradient, let’s visualize a non-outlier.





As training progresses, the top singular vectors of adjacent layers align to amplify meaningful subspaces. [3, 4]

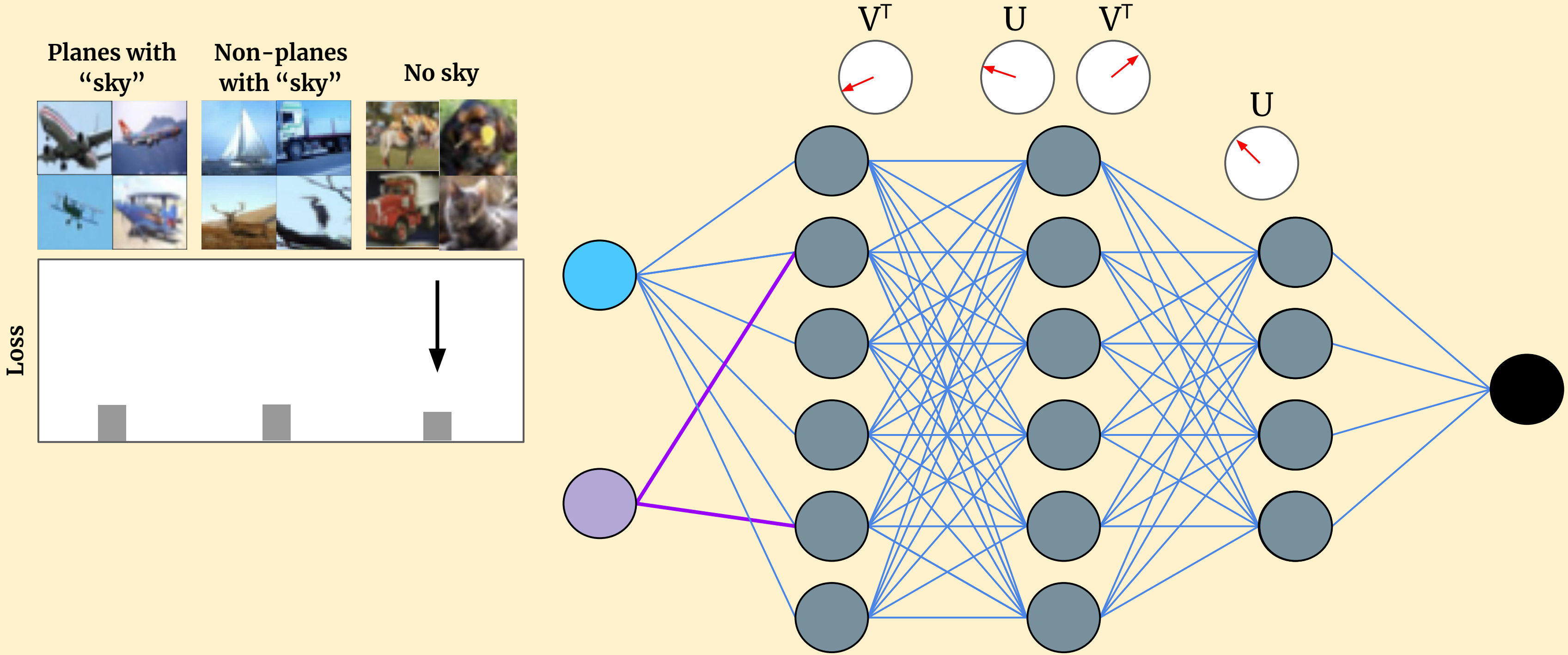
This is how the “shape” feature gets *upweighted.*



[3] Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. Saxe et al. 2013
[4] Unique properties of flat minima in deep networks. Muyaloff and Michaeli, 2020.

As training progresses, the top singular vectors of adjacent layers align to amplify meaningful subspaces. [3, 4]

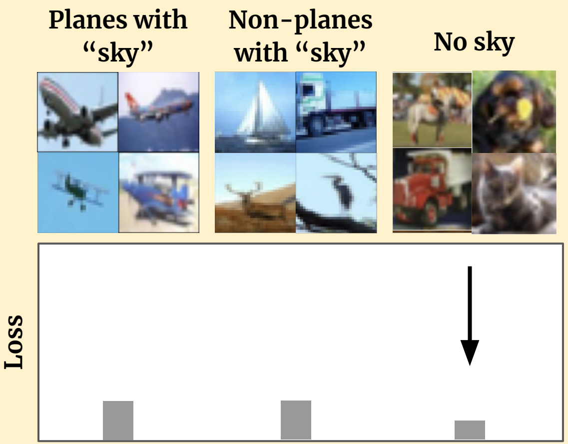
This is how the “shape” feature gets *upweighted.*

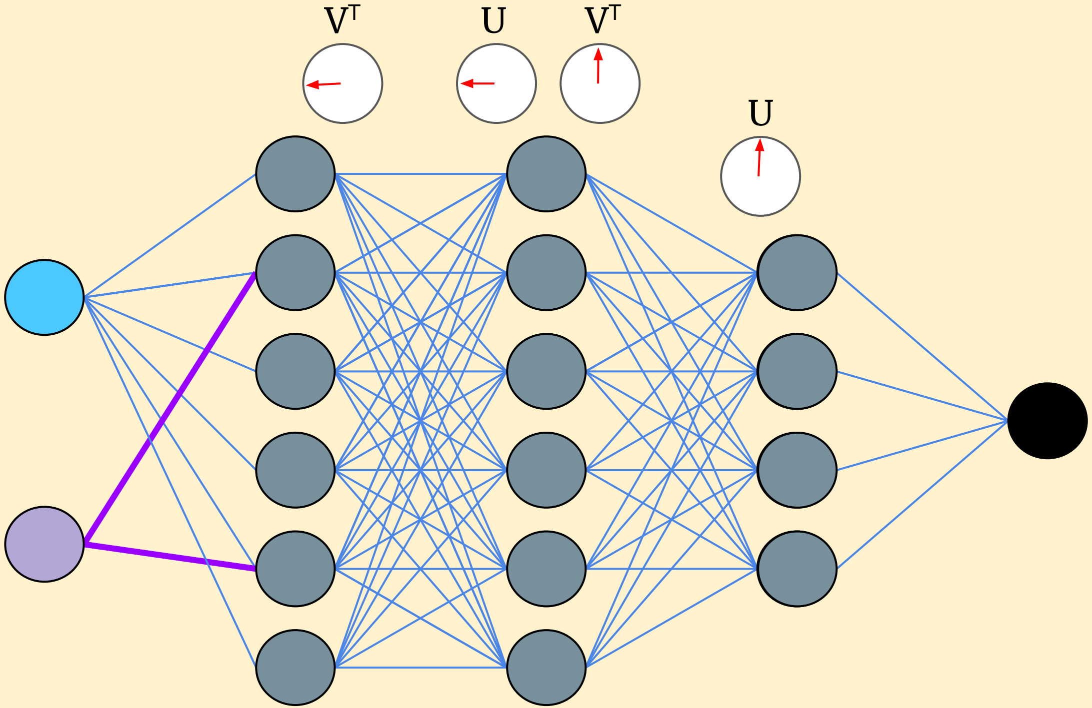


[3] Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. Saxe et al. 2013

[4] Unique properties of flat minima in deep networks. Muyaloff and Michaeli, 2020.

This alignment has been continuously upweighting the more useful signal.





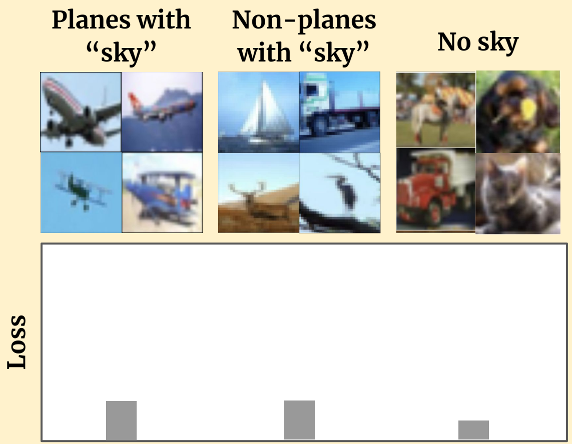
[3] Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. Saxe et al. 2013
[4] Unique properties of flat minima in deep networks. Muyaloff and Michaeli, 2020.

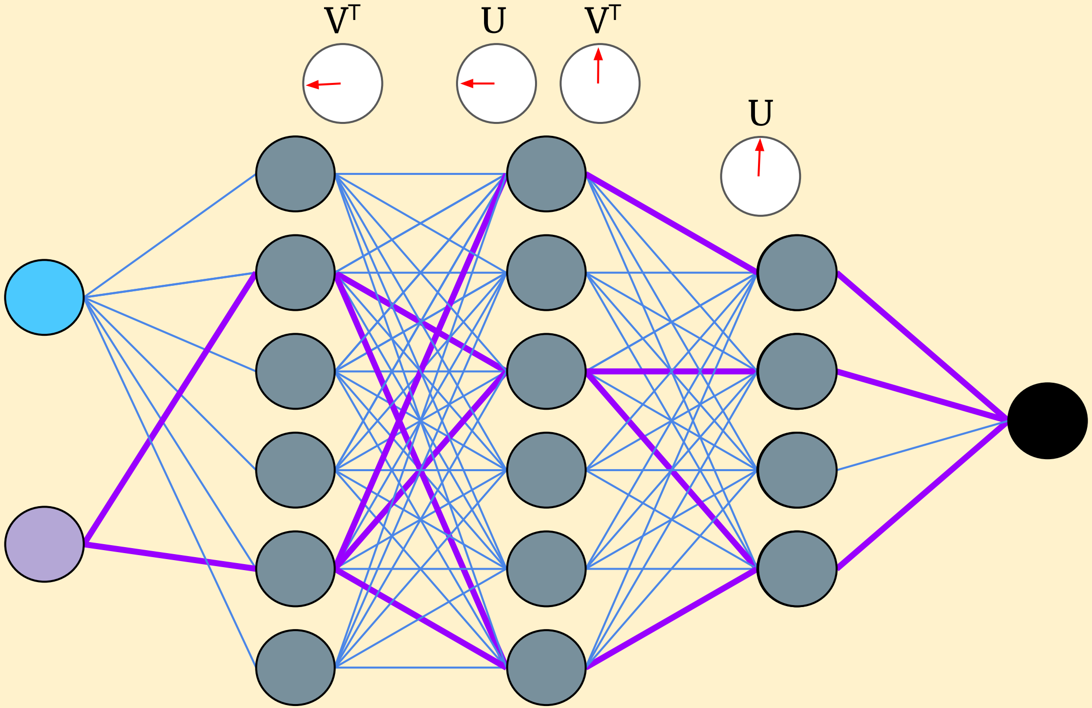
I’ve left one important part out of this visualization:

When “shape” is amplified, **“sky” is amplified too**.

This is the activation pattern for a *non-outlier*.

What would it look like for an outlier with a sky background?

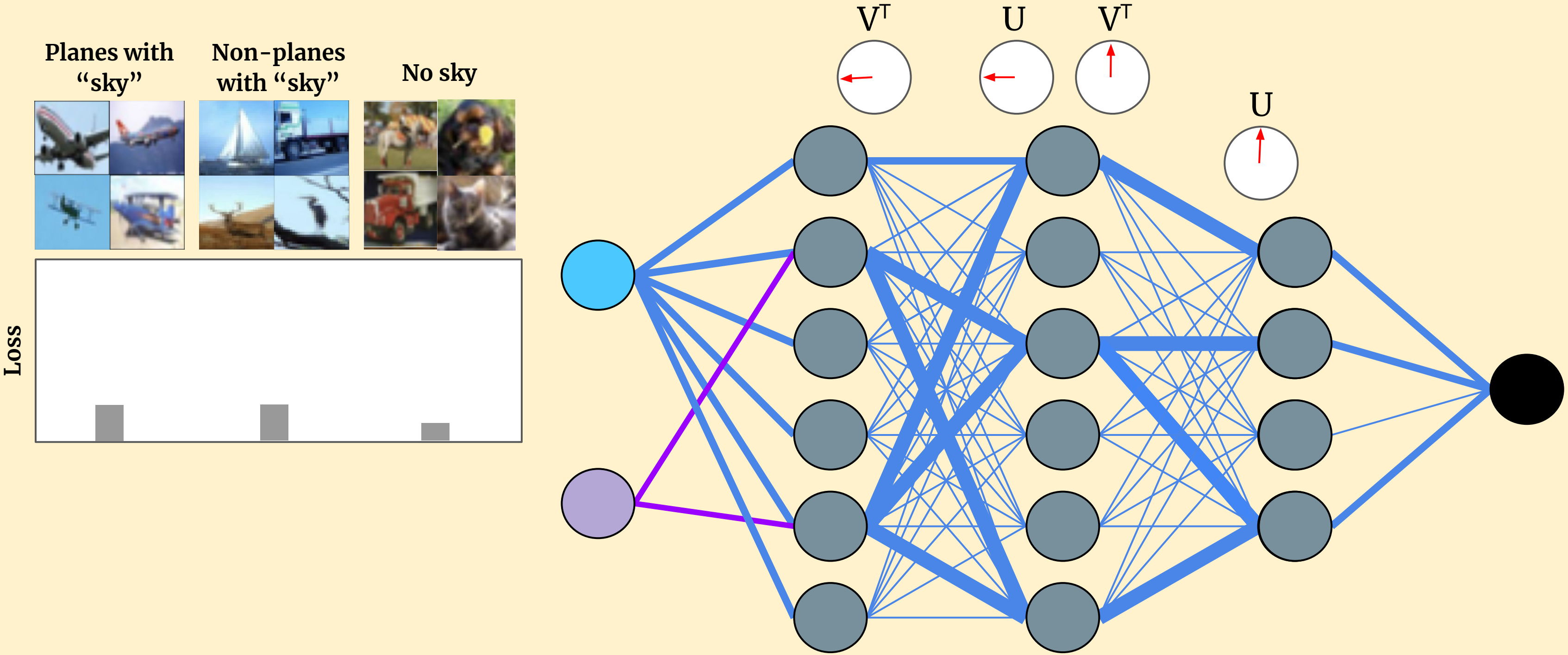




I've left one important part out of this visualization:

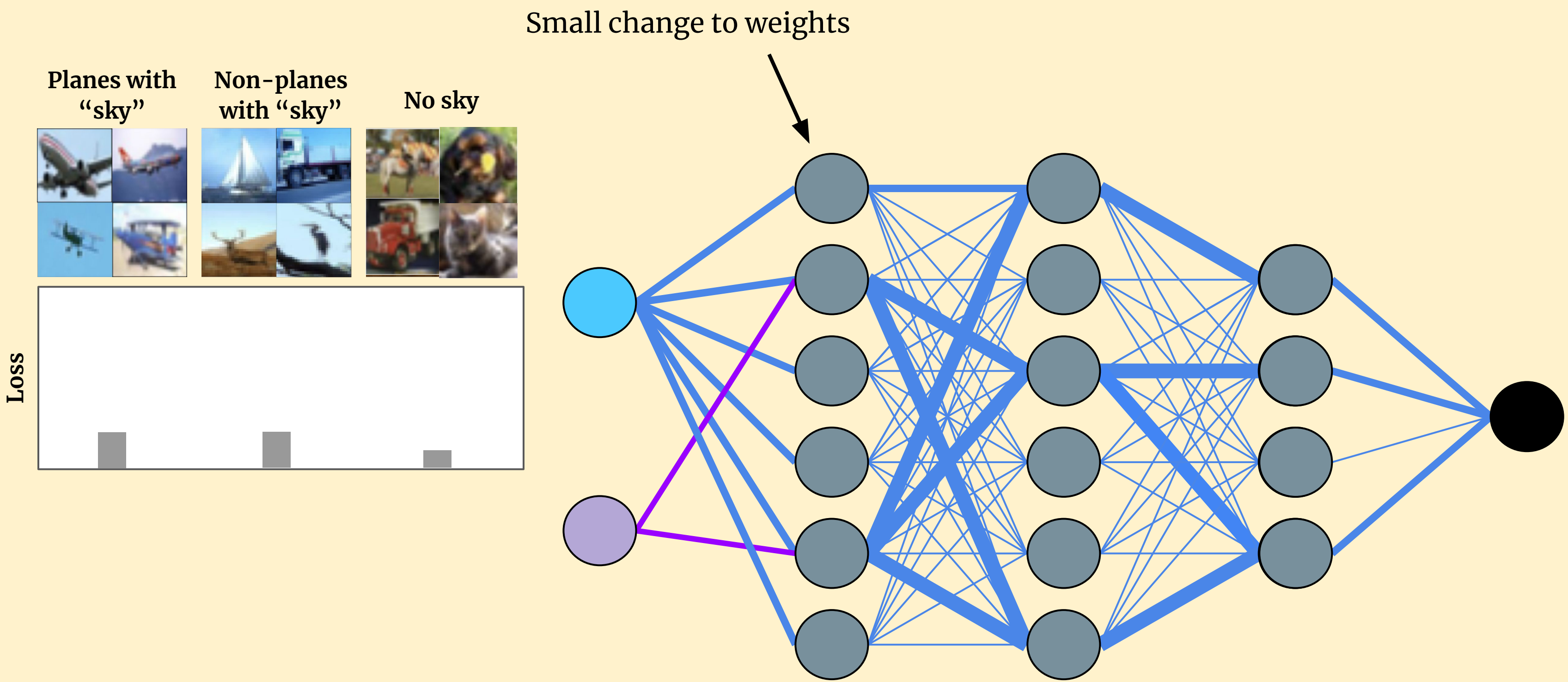
When "shape" is amplified, **"sky" is amplified too**.

Because it is larger + more pervasive, it still dominates the network's activations.



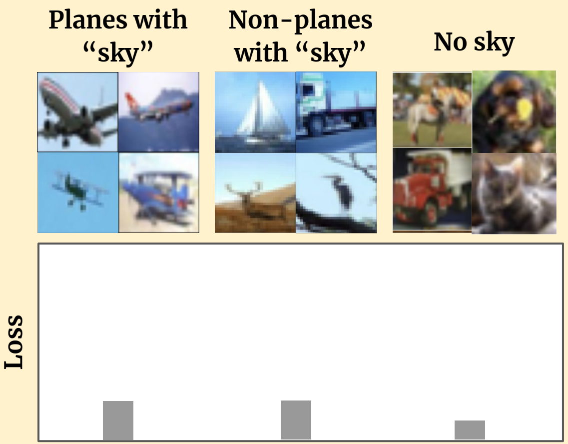
This causes large sensitivity to small changes in *how the network uses* “sky”.

- Small, targeted change to predict one group massively increases loss on the other.

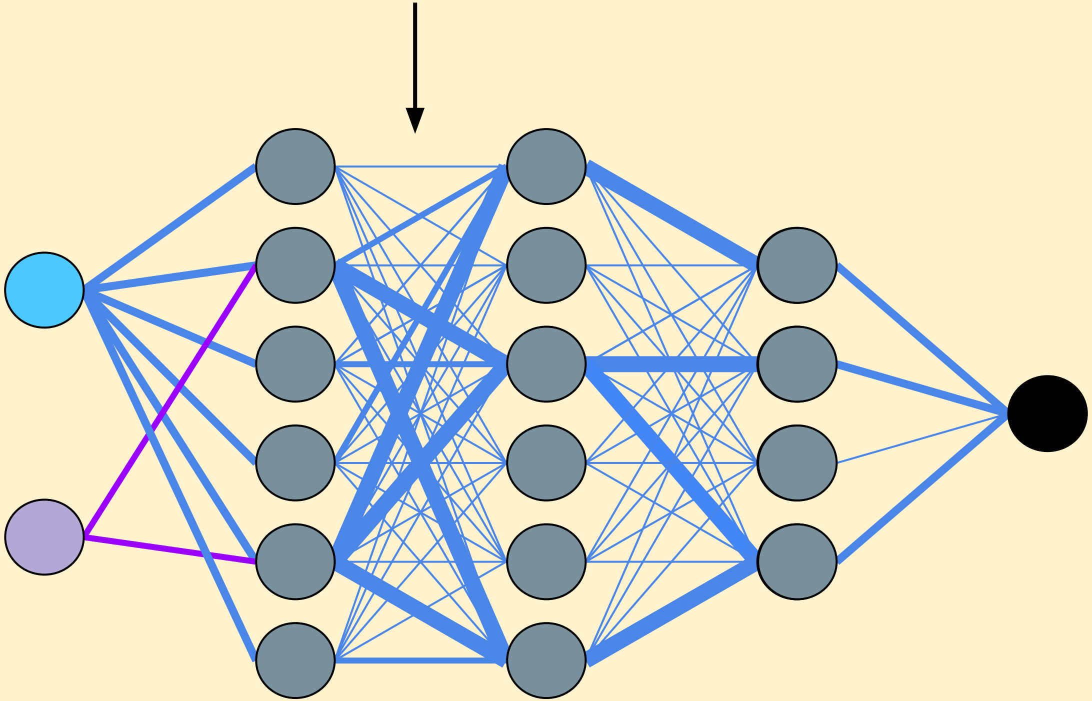


This causes large sensitivity to small changes in *how the network uses* "sky".

- Small, targeted change to predict one group massively increases loss on the other.

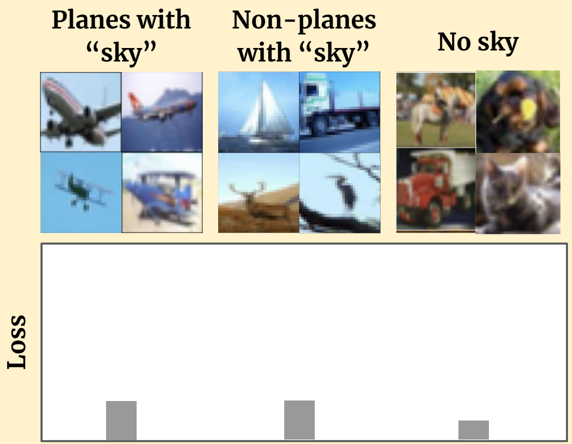


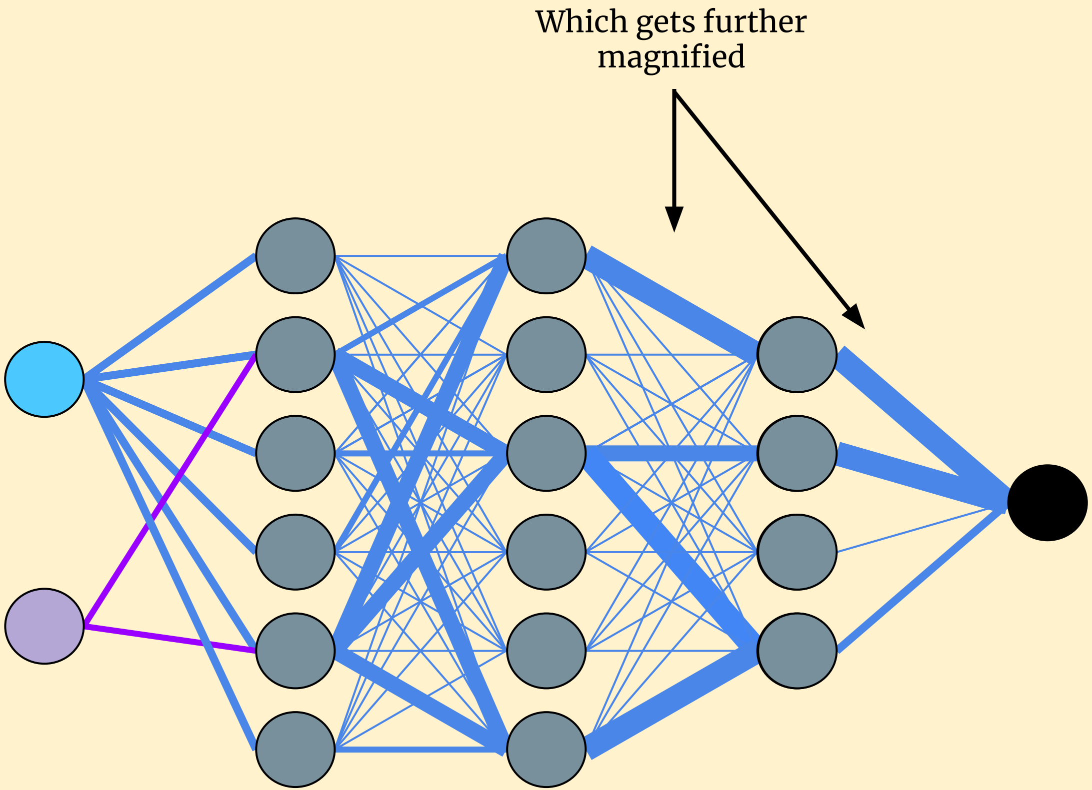
Changes how feature propagates...



This causes large sensitivity to small changes in *how the network uses* "sky".

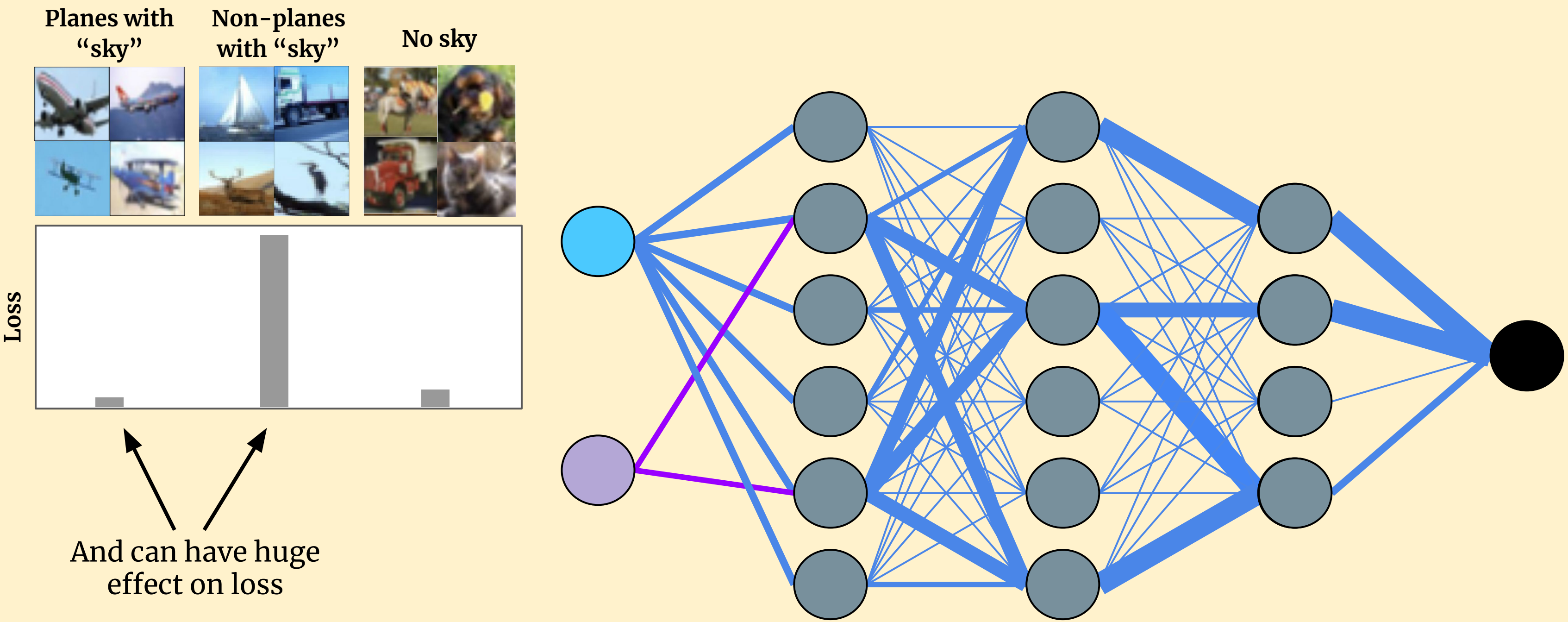
- Small, targeted change to predict one group massively increases loss on the other.





This causes large sensitivity to small changes in *how the network uses* “sky”.

- Small, targeted change to predict one group massively increases loss on the other.

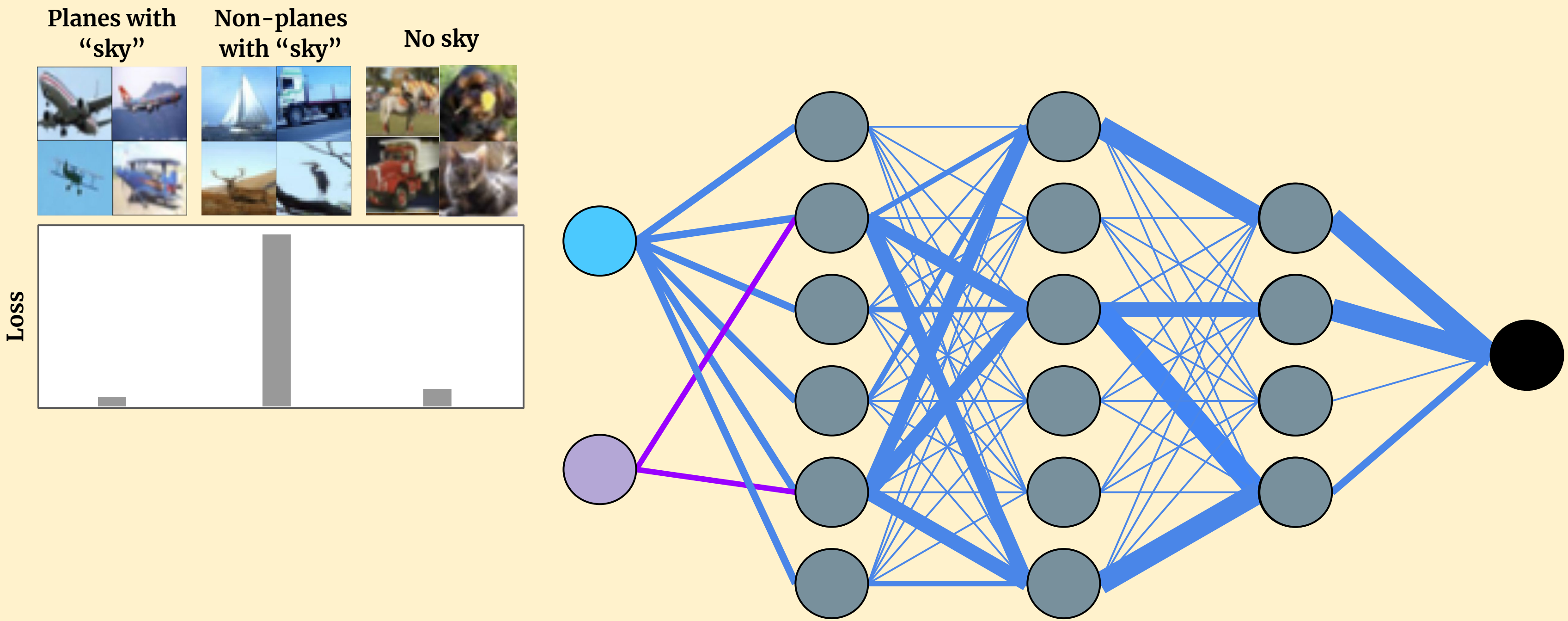


This causes large sensitivity to small changes in *how the network uses* "sky".

- Small, targeted change to predict one group massively increases loss on the other.

In other words, **loss on outliers becomes very *sharp* w.r.t. parameters.**

- ("growth in sensitivity" was previously noted, e.g. weight/Jacobian norm[5, 6])



[5] On linear stability of sgd and input-smoothness of neural networks. Ma and Ying, 2021.
[6] On the lipschitz constant of deep networks and double descent. Gamba et al. 2023.

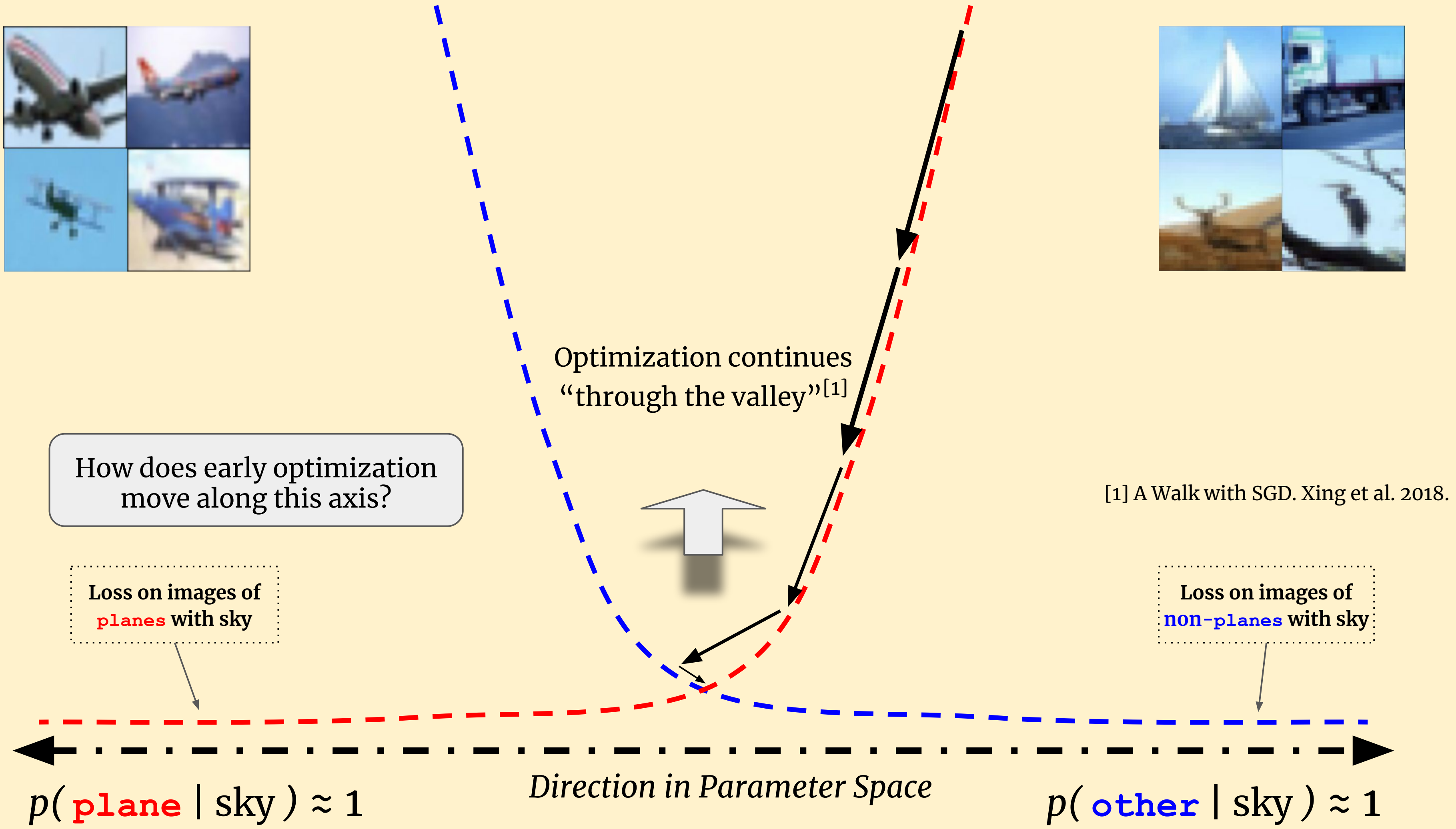
This story is pretty abstract.

Let's visualize something more concrete:

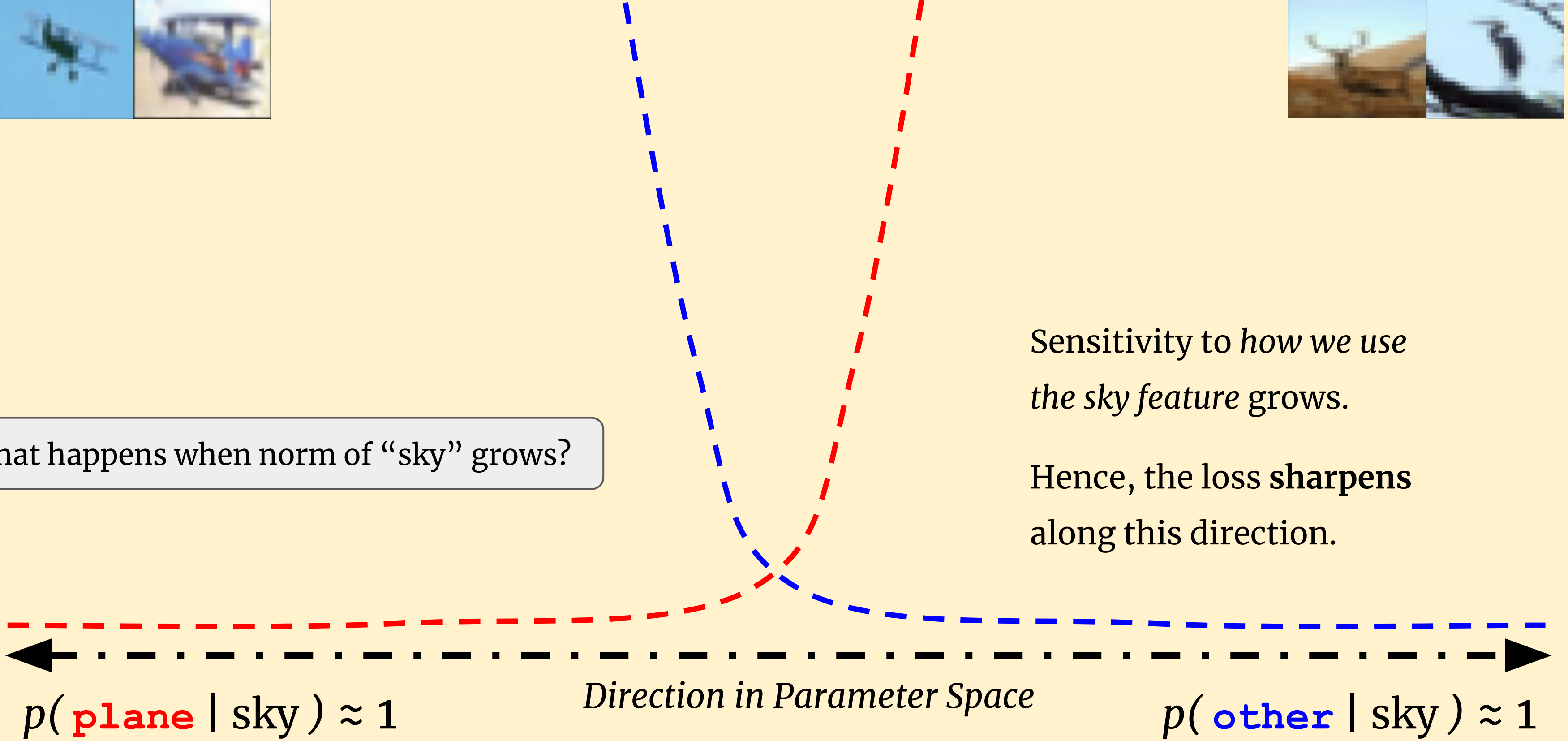
The (hypothetical) loss in a 1D parameter space.

Direction in Parameter Space

$p($ plane $|$ sky $) \approx 1$ $p($ other $|$ sky $) \approx 1$



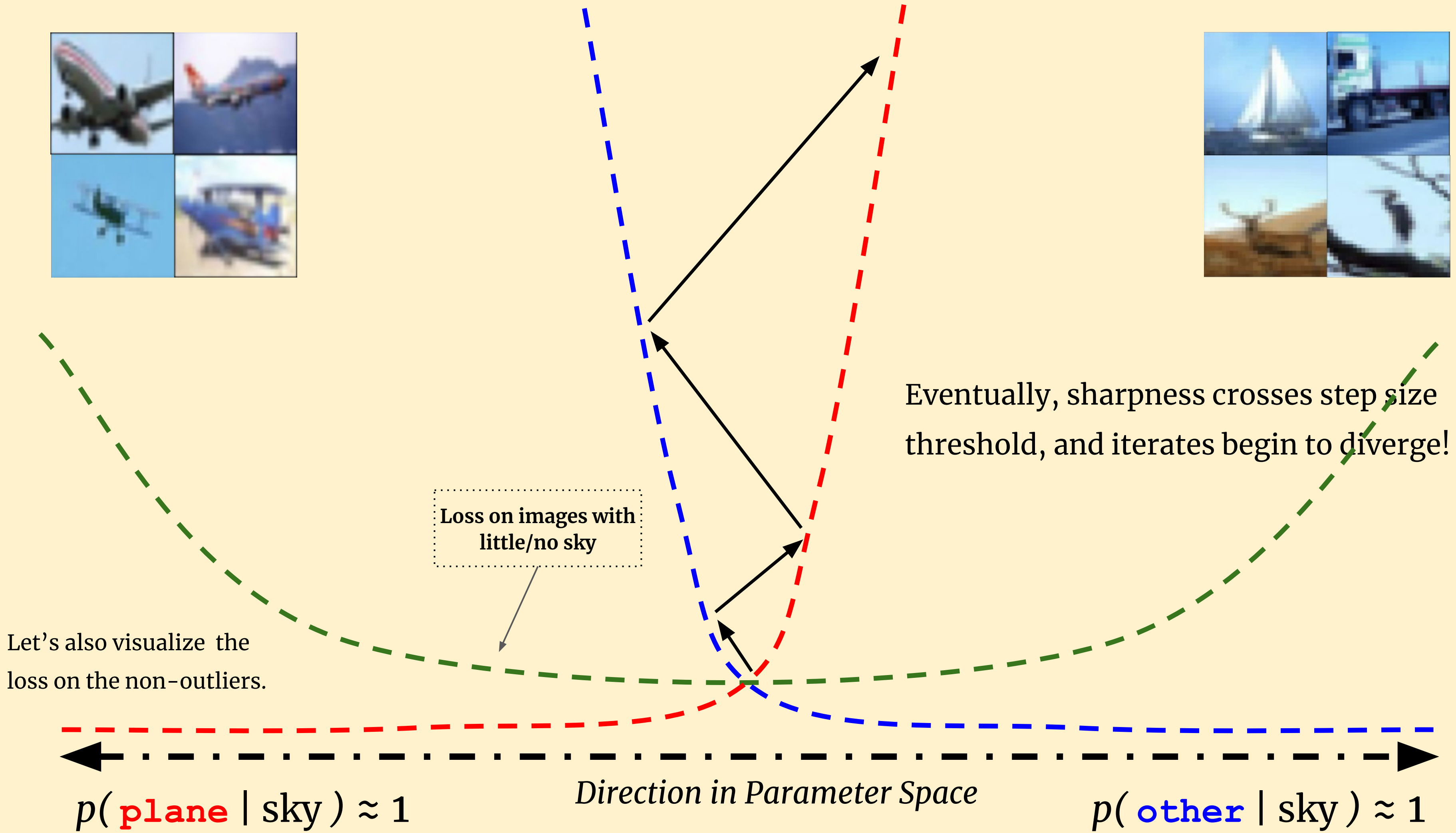
Optimization continues
“through the valley”[1]
How does early optimization
move along this axis?
[1] A Walk with SGD. Xing et al. 2018.
Loss on images of
planes with sky
Loss on images of
non-planes with sky
Direction in Parameter Space
p(plane | sky) ≈ 1
p(other | sky) ≈ 1

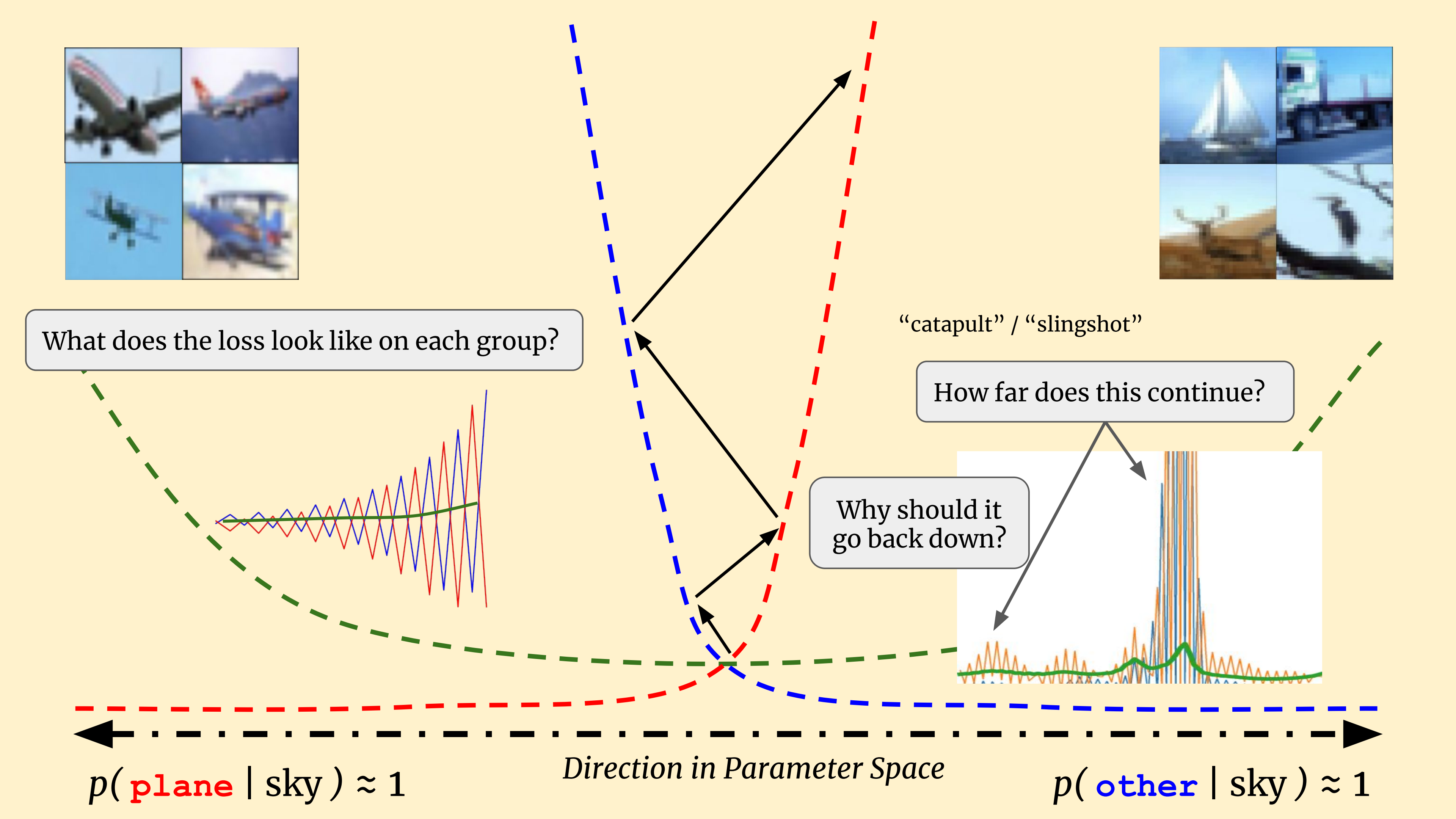


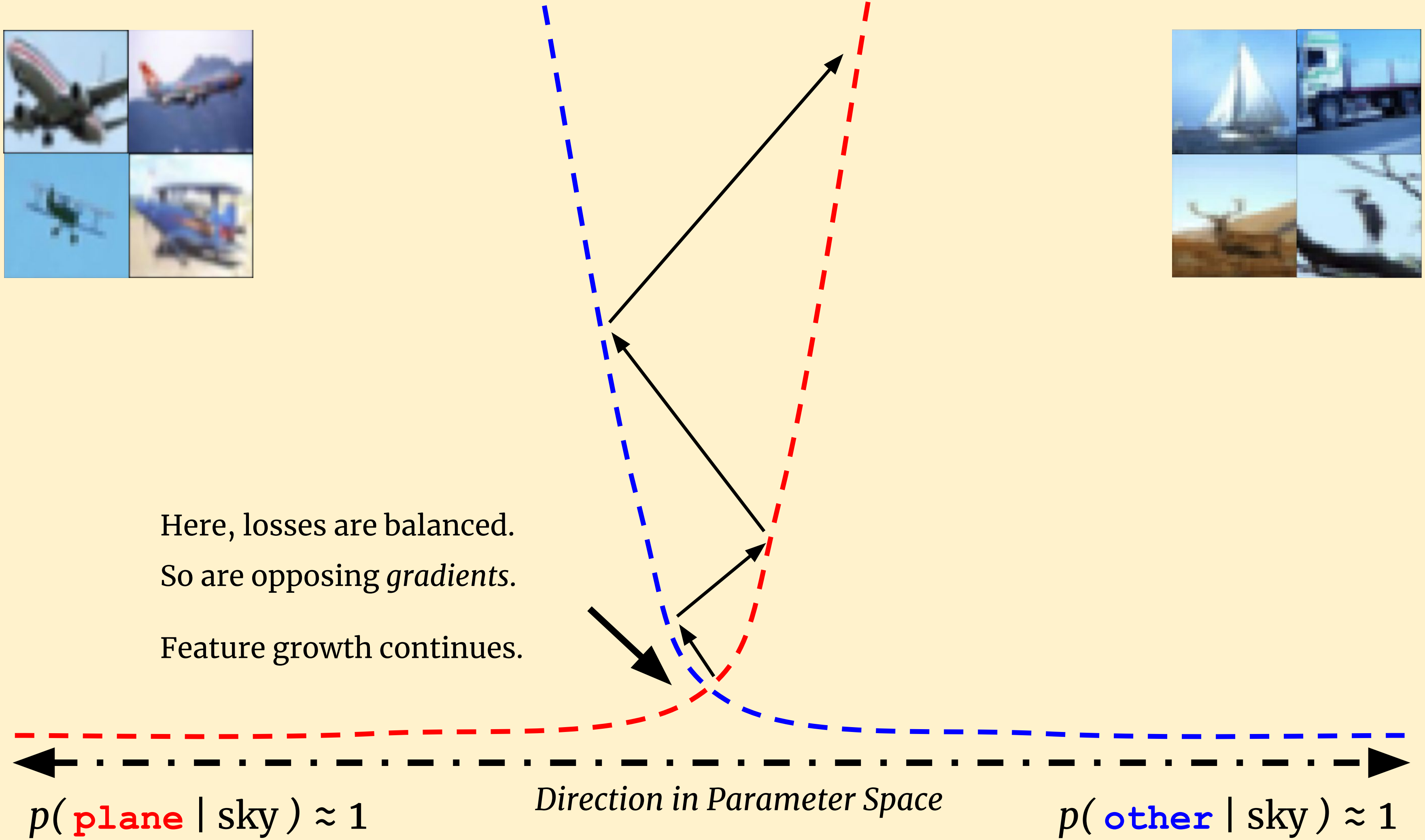
What happens when norm of “sky” grows?

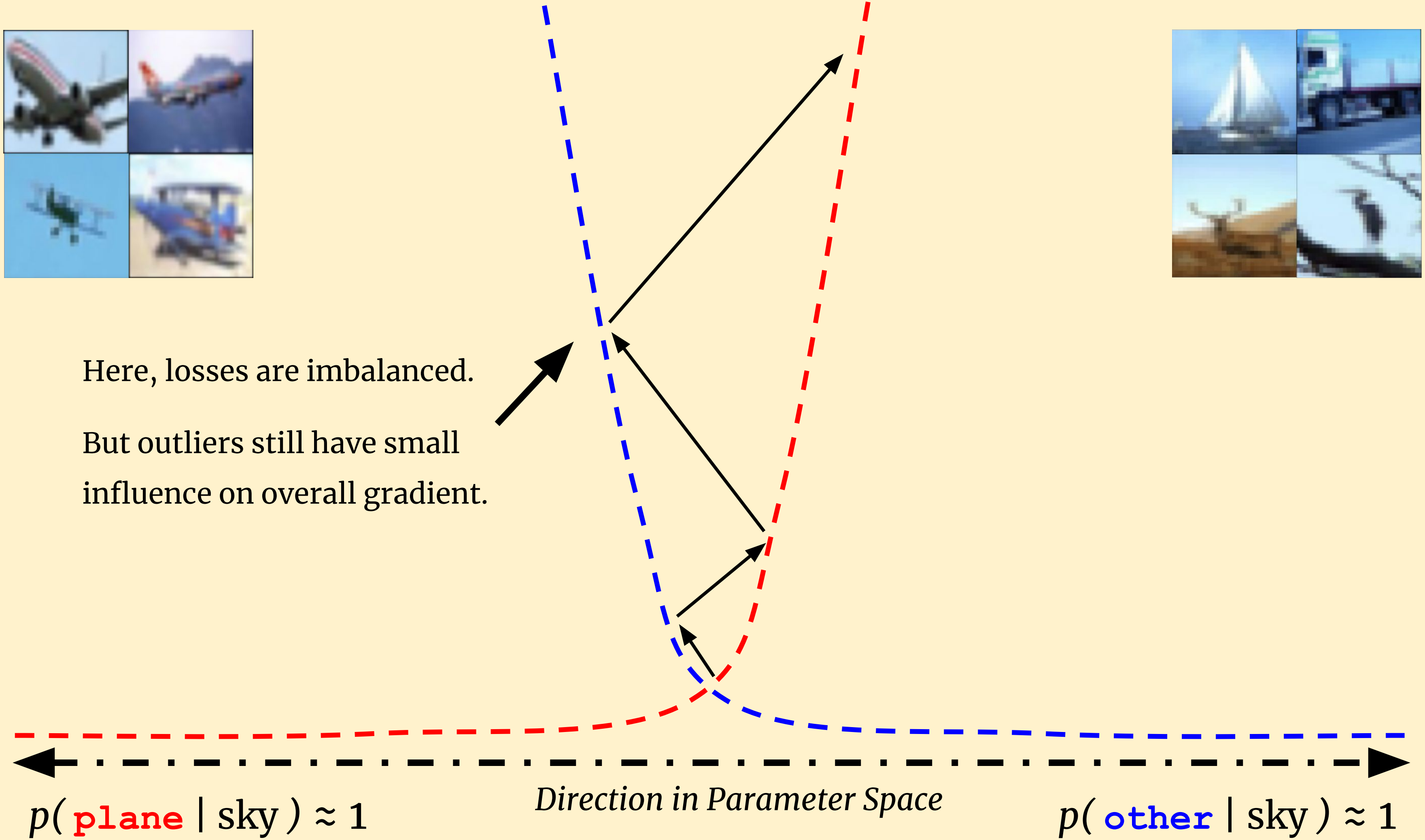
Sensitivity to *how we use the sky feature* grows.

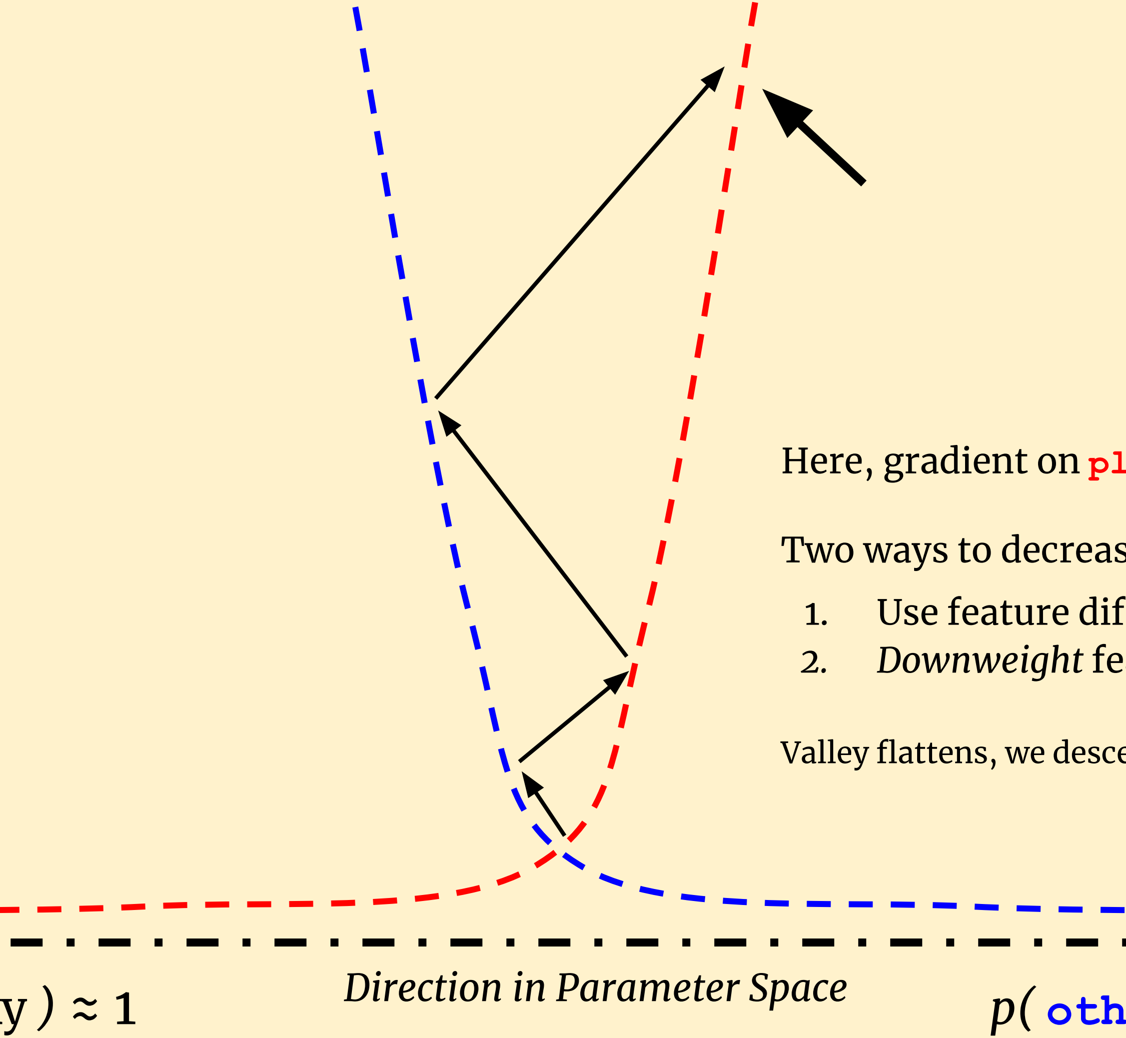
Hence, the loss **sharpens** along this direction.


Eventually, sharpness crosses step size threshold, and iterates begin to diverge!
Loss on images with little/no sky
Let's also visualize the loss on the non-outliers.
Direction in Parameter Space
p(plane | sky) ≈ 1
p(other | sky) ≈ 1


What does the loss look like on each group?
"catapult" / "slingshot"
How far does this continue?
Why should it go back down?
Direction in Parameter Space
p(plane | sky) ≈ 1
p(other | sky) ≈ 1


Here, losses are balanced.
So are opposing *gradients*.
Feature growth continues.
Direction in Parameter Space
$p($ plane $|$ sky $) \approx 1$
$p($ other $|$ sky $) \approx 1$


Here, losses are imbalanced.
But outliers still have small
influence on overall gradient.
Direction in Parameter Space
p(plane | sky) ≈ 1
p(other | sky) ≈ 1



Here, gradient on `plane` dominates.

Two ways to decrease loss:

1. Use feature differently.
2. *Downweight* feature.

Valley flattens, we descend again.

Experimental Verification

The value of a theory (even a non-rigorous one) is in its ability to make predictions.

So far we've described:

1. Initial phase of fitting a "linear" model. ← (previously observed)
2. Growth in activation magnitude among images with this feature. ← (least well understood)
3. Upon reaching Edge of Stability, predictions *oscillate* between "sky = `plane`" and "sky = `other`".
4. Oscillation results in shrinking of activation magnitude.

What does this story imply, *behaviorally*? Can we test it more directly?

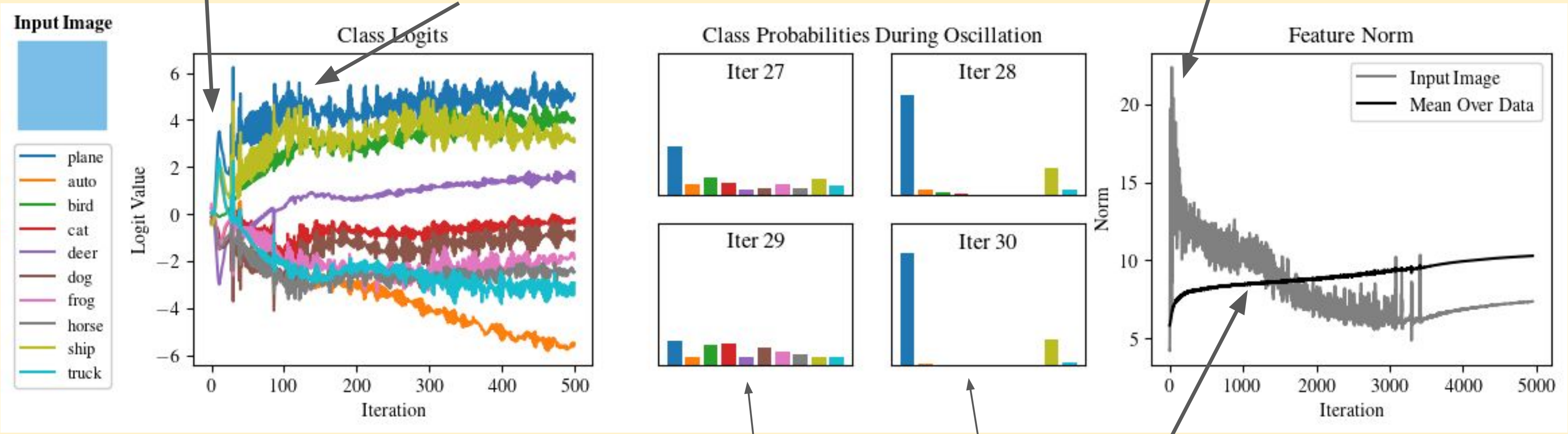
Experimental Verification

To avoid confounders, we'll pass a pure "sky" image through a ResNet-18.

Smooth initial "linear" phase

Then, logit oscillations

Growth in embedding norm increases sharpness



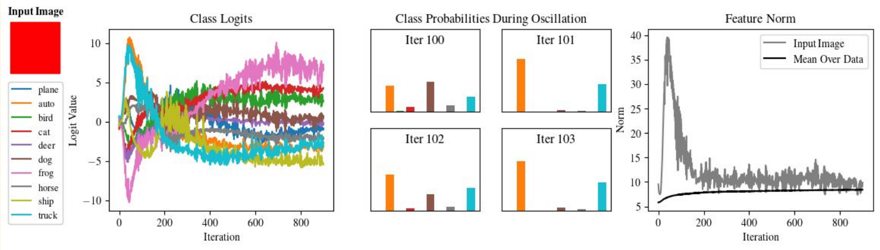
Sky = `other`

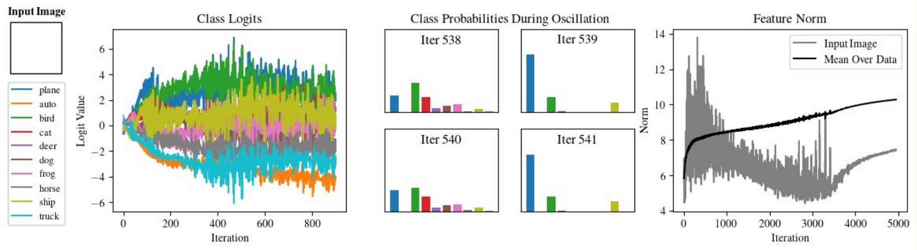
Sky = `plane` (and a bit `ship`)

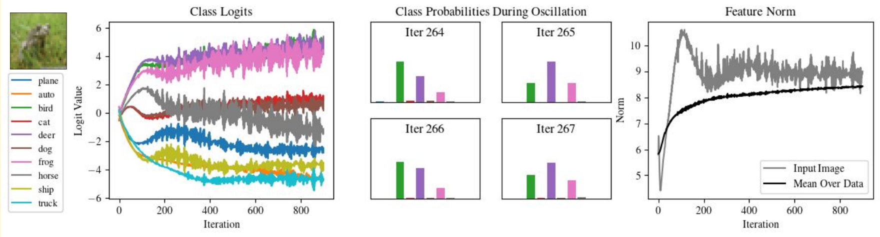
Slow decay during oscillation

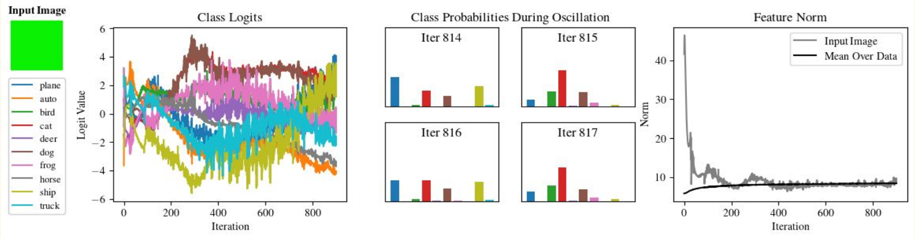
Experimental Verification

(Doesn't happen as cleanly for all archs/colors, but it's pretty consistent.)





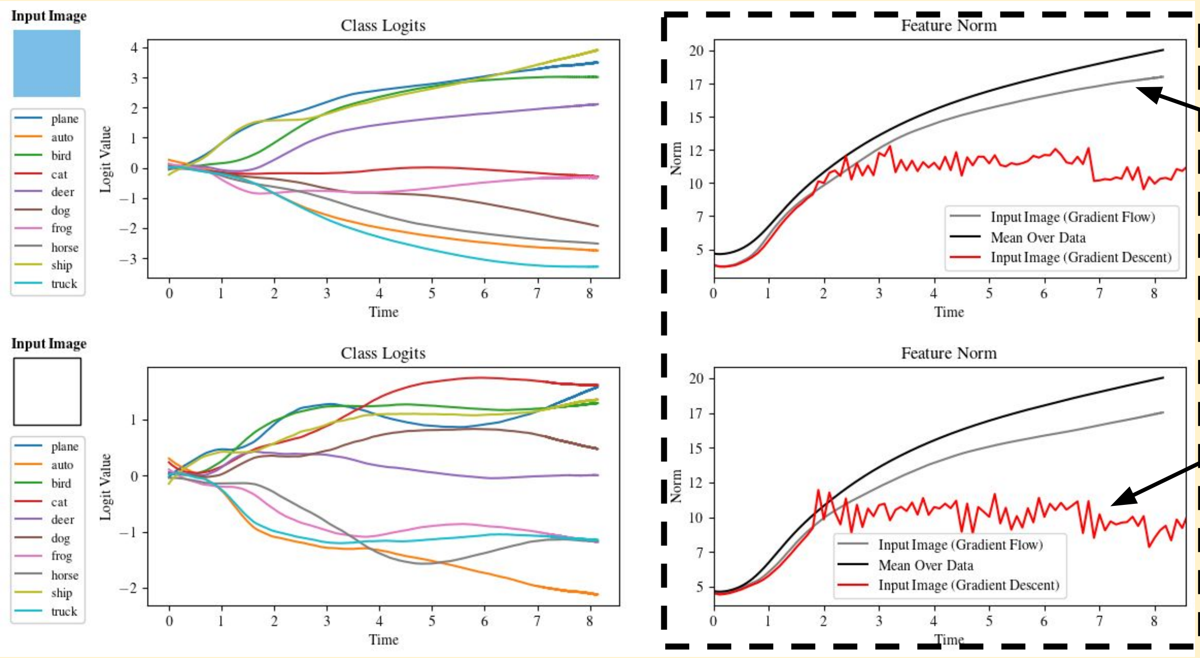




Experimental Verification

Oscillation seems valuable for downweighting the “simple” but “incomplete” features.

- *Gradient Flow doesn’t oscillate*. Maybe that’s part of why it generalizes poorly?



Under gradient flow, feature norm grows continuously.

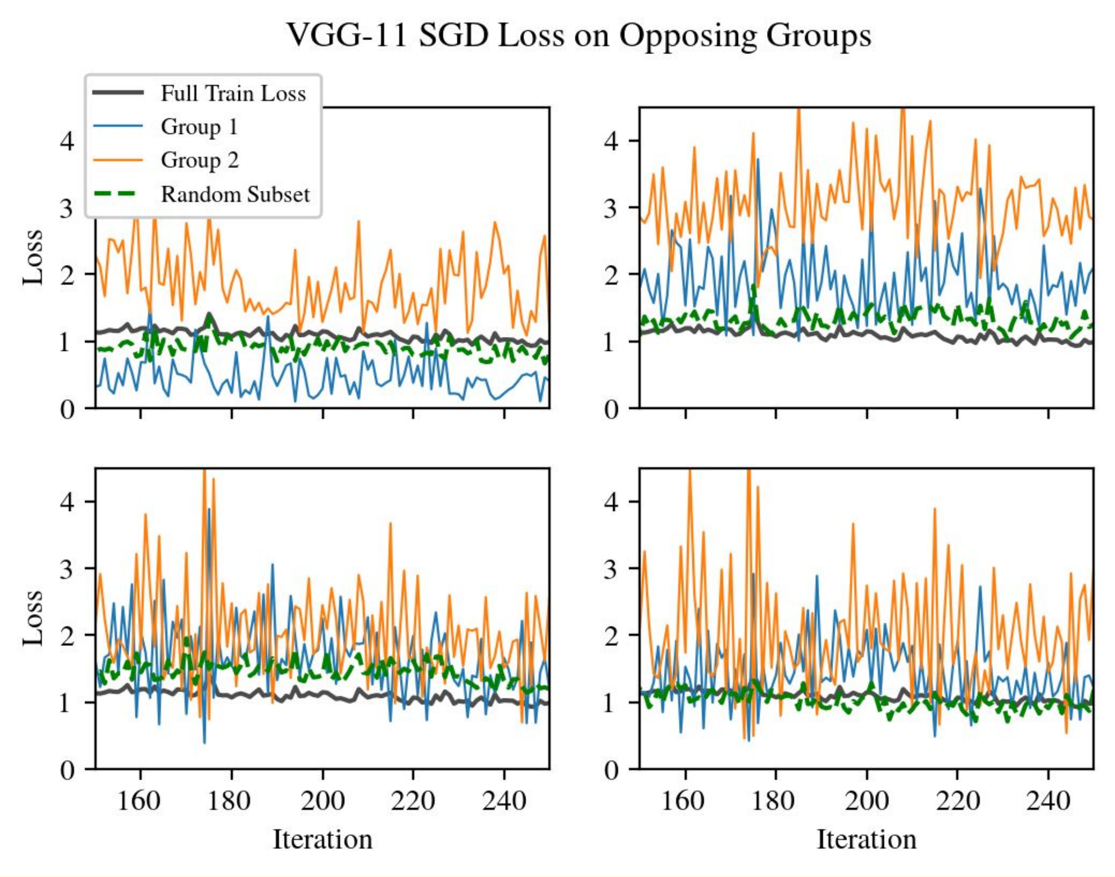
Gradient descent matches flow initially, but *norm starts decreasing* once oscillation begins

Does this Occur for SGD?

Long story short, **Yes**.

Alternations are not every step.

Groups are not always opposite.



Opposing Signals have clear *potential* connections to existing tools in stochastic optimization, for both training speed and generalization:

- *Batch Normalization*
- *Adaptive Gradient Methods*
- *Sharpness-Aware Minimization*
- *Large Initial Learning Rate*

Maybe these methods work because of how they handle Opposing Signals?

- **Could this help us design new improvements to SGD?**

Lots of unanswered questions.
Very happy to discuss further.

Outliers with Opposing Signals Have an Outsized Effect on Neural Network Optimization

Elan Rosenfeld & Andrej Risteski
https://arxiv.org/abs/2311.04163

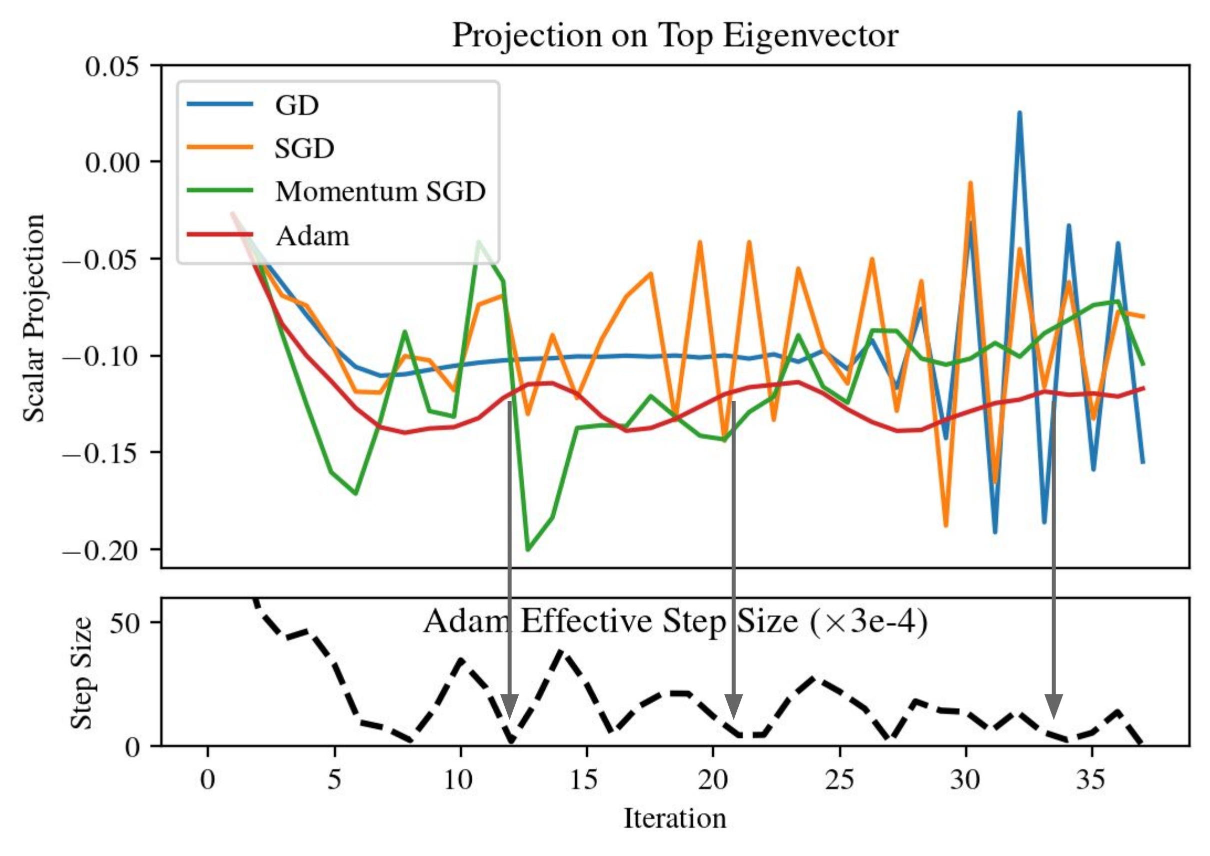
Implications for Stochastic Optimization

A Case Study of Adam vs. SGD

Adam looks markedly different!

Prevents steps that would *approach the local minimum*

Effective step size **drops sharply** when approaching valley floor.



Remember this?