TECHNISCHE UNIVERSITÄT DARMSTADT

# SafeSplit: A Novel Defense Against Client-Side Backdoor Attacks in Split Learning

NDSS 2025

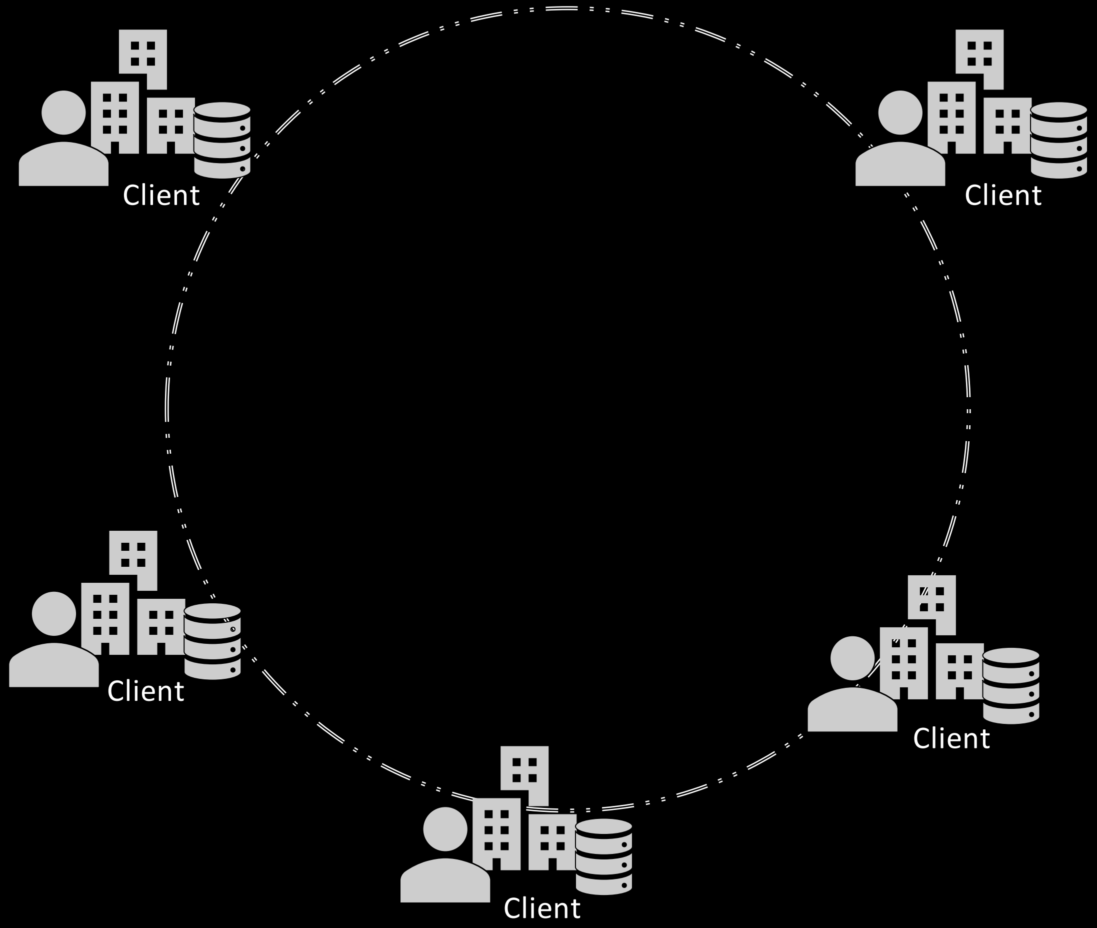
Phillip Rieger, Alessandro Pegoraro, Kavita Kumari, Tigist Abera, Jonathan Knauer, Ahmad-Reza Sadeghi

Technical University of Darmstadt

SYSTEM SECURITY LAB

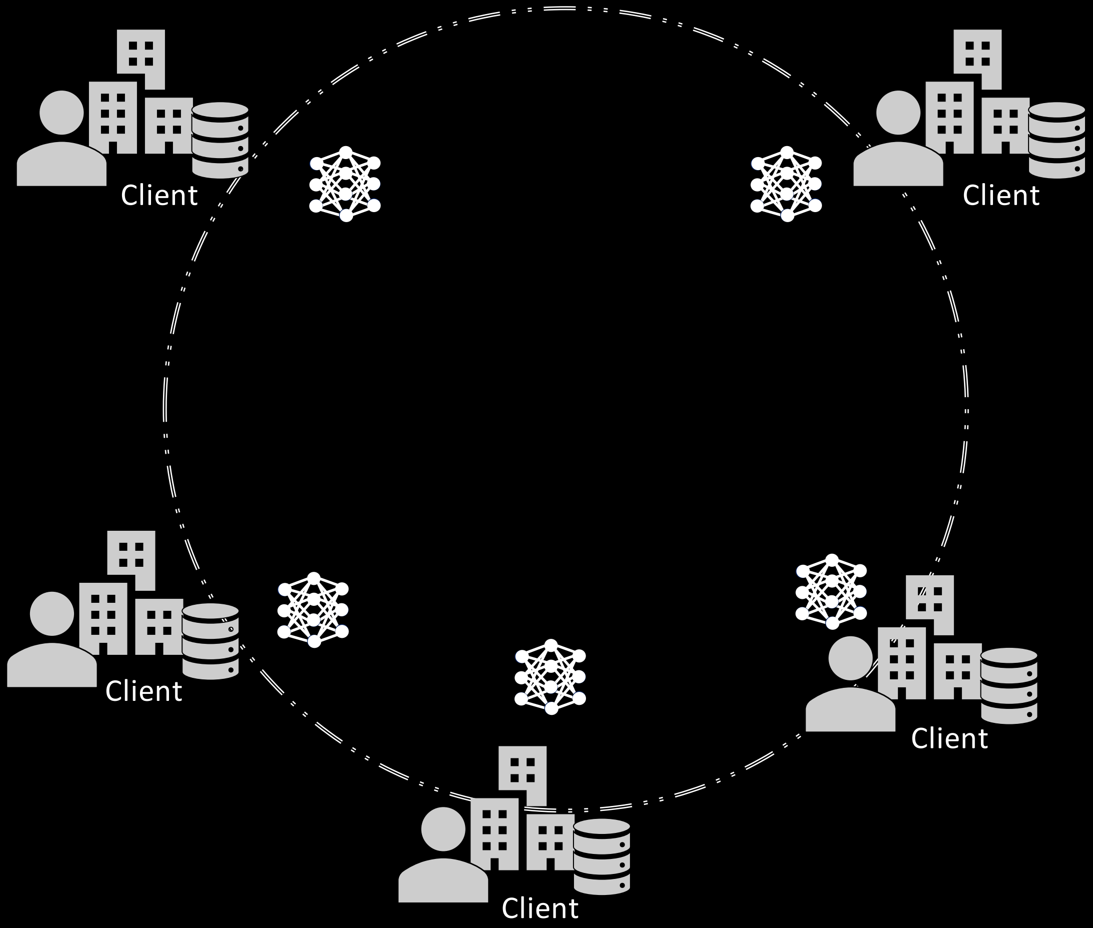
# Types of Collaborative Learning

## Federated Learning (FL)



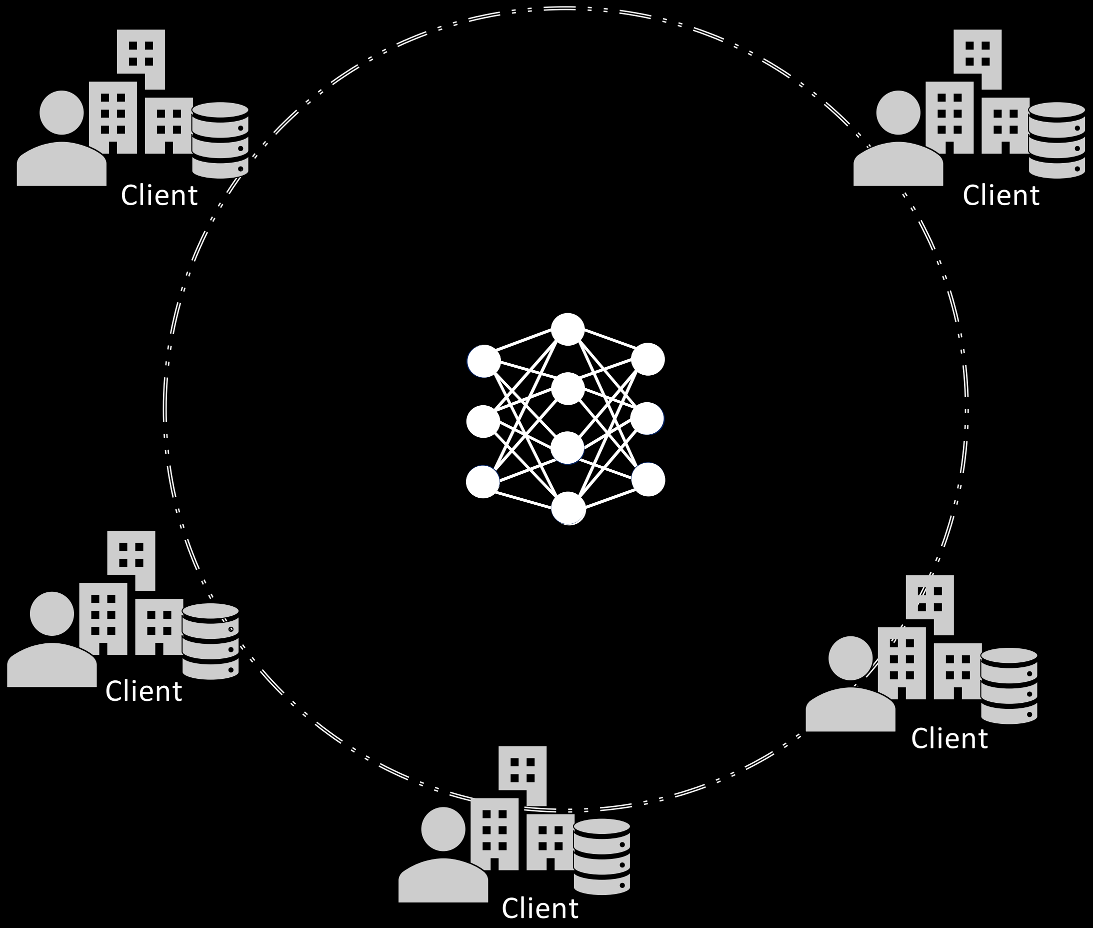
# Types of Collaborative Learning

## Federated Learning (FL)



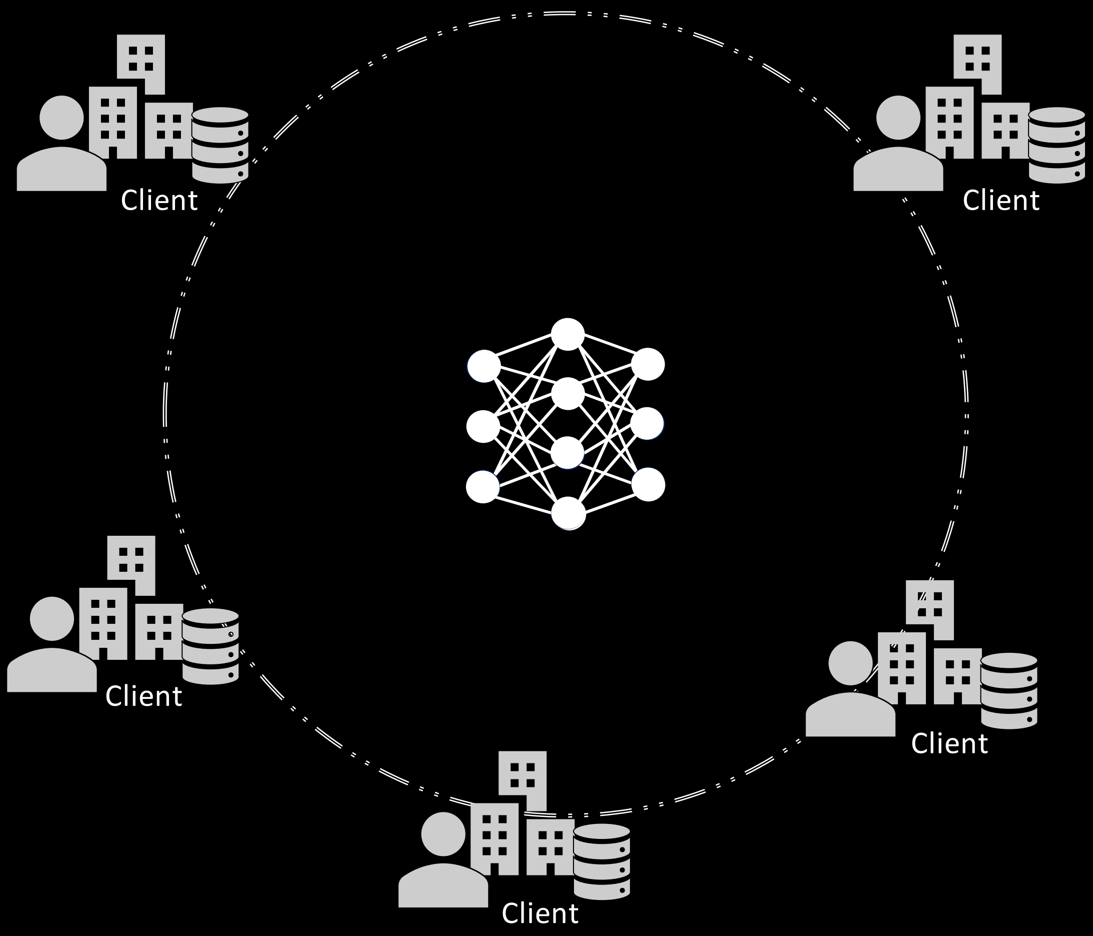
# Types of Collaborative Learning

## Federated Learning (FL)

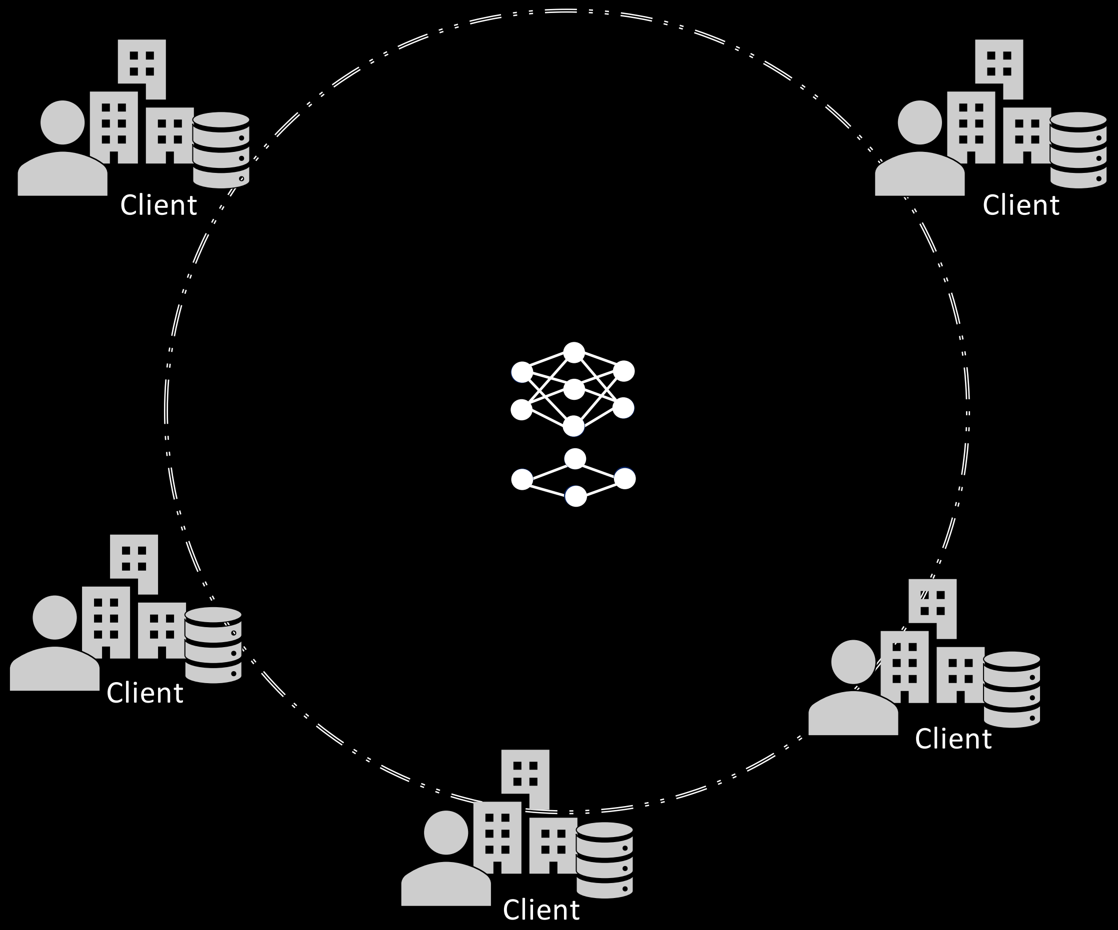


# Types of Collaborative Learning

Federated Learning (FL)

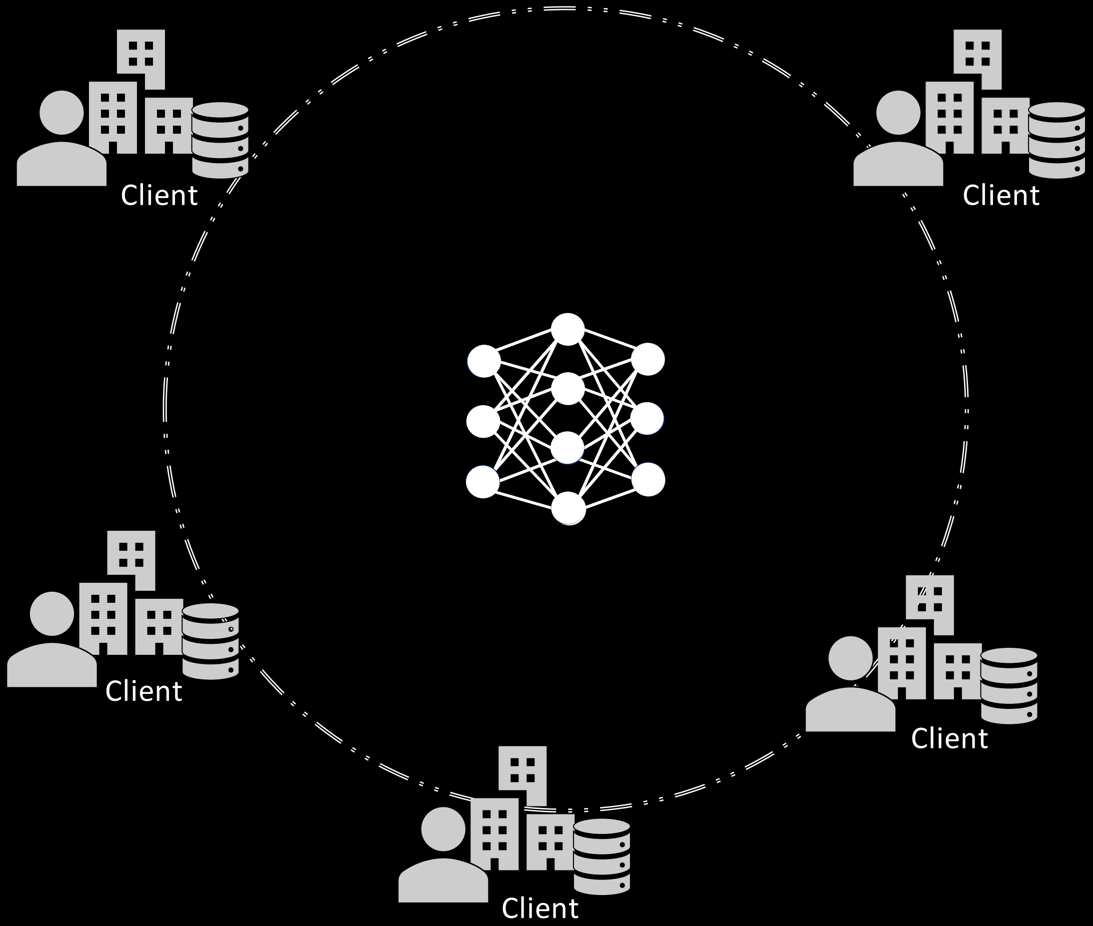



Split Learning

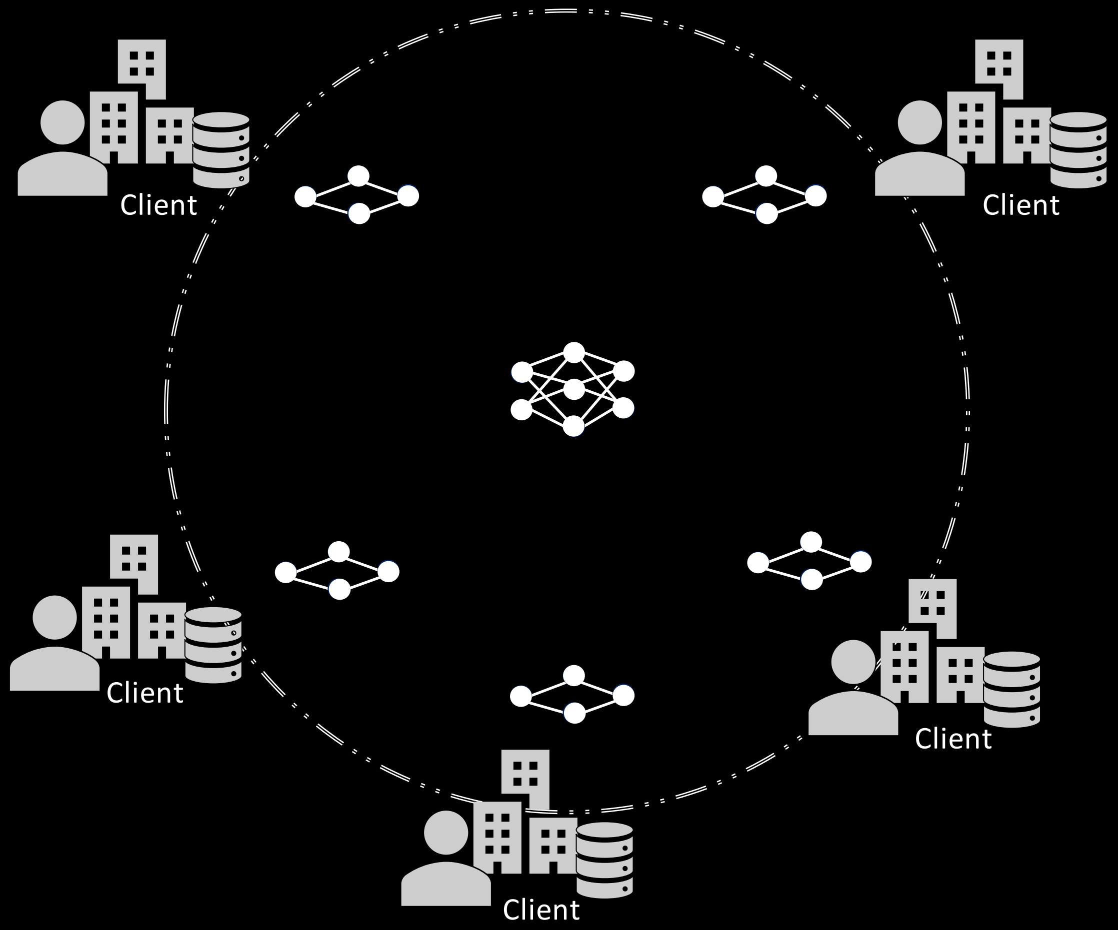


# Types of Collaborative Learning

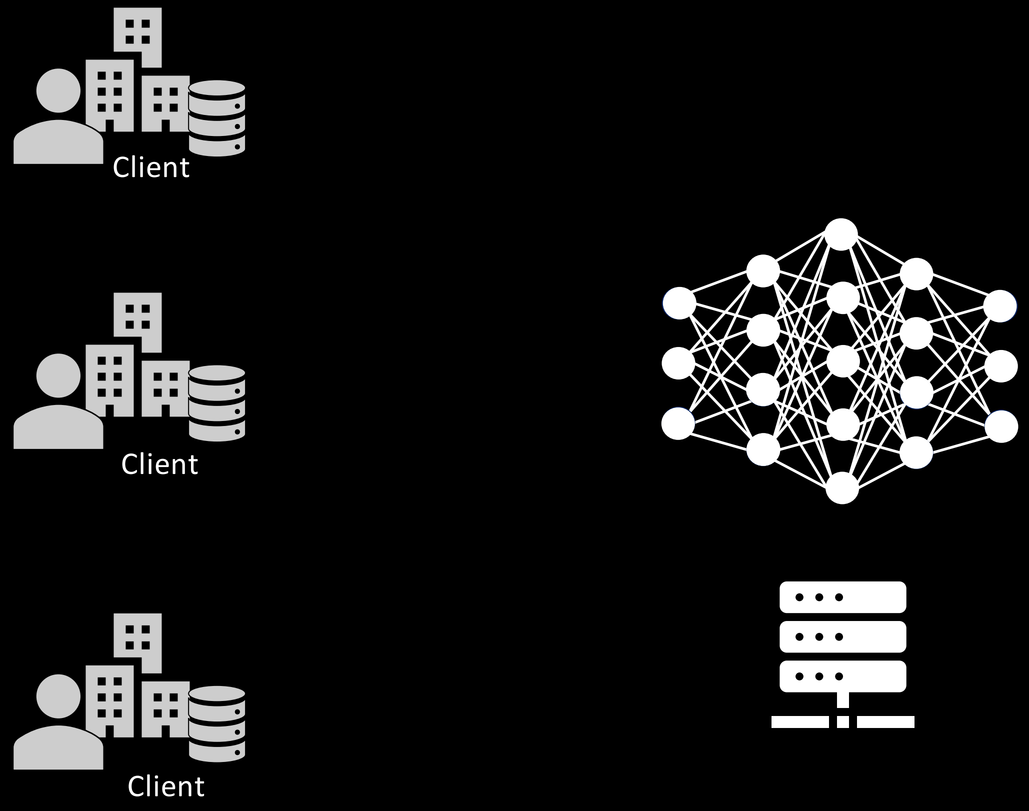
## Federated Learning (FL)



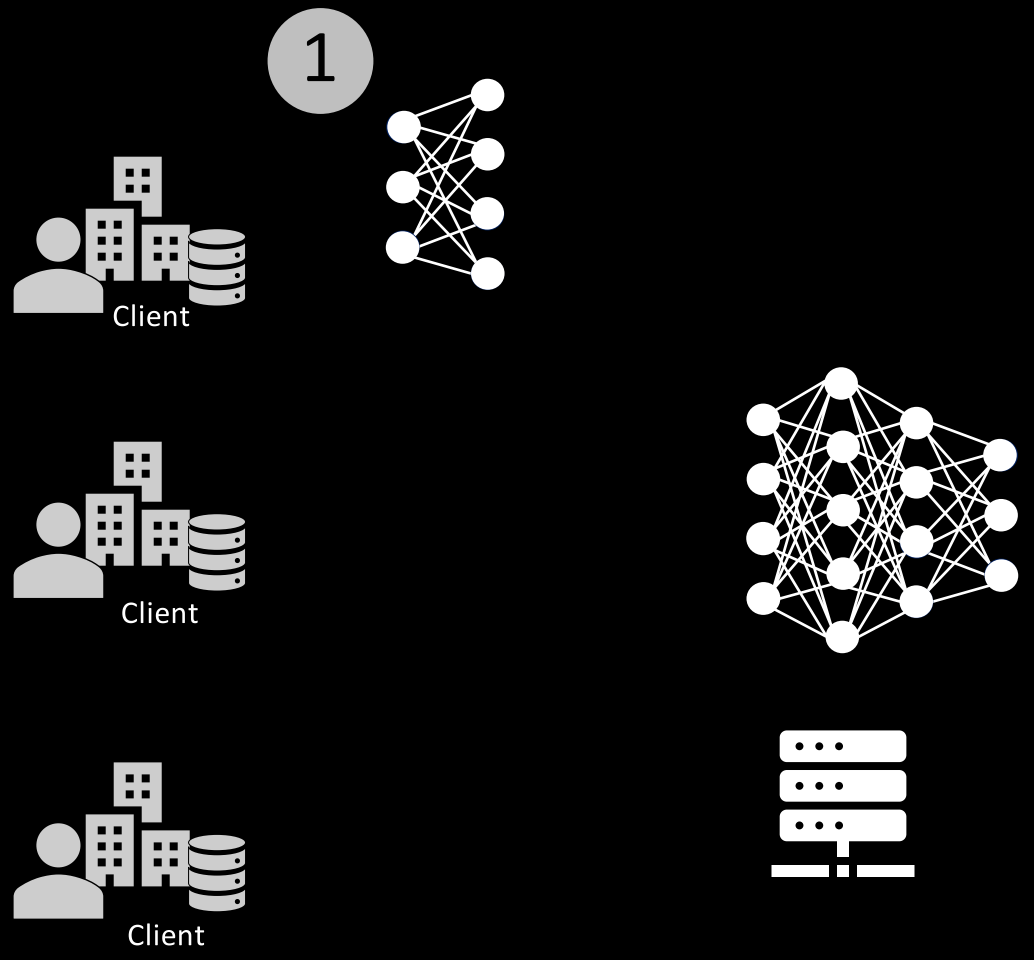

## Split Learning



# Split Learning – Concept

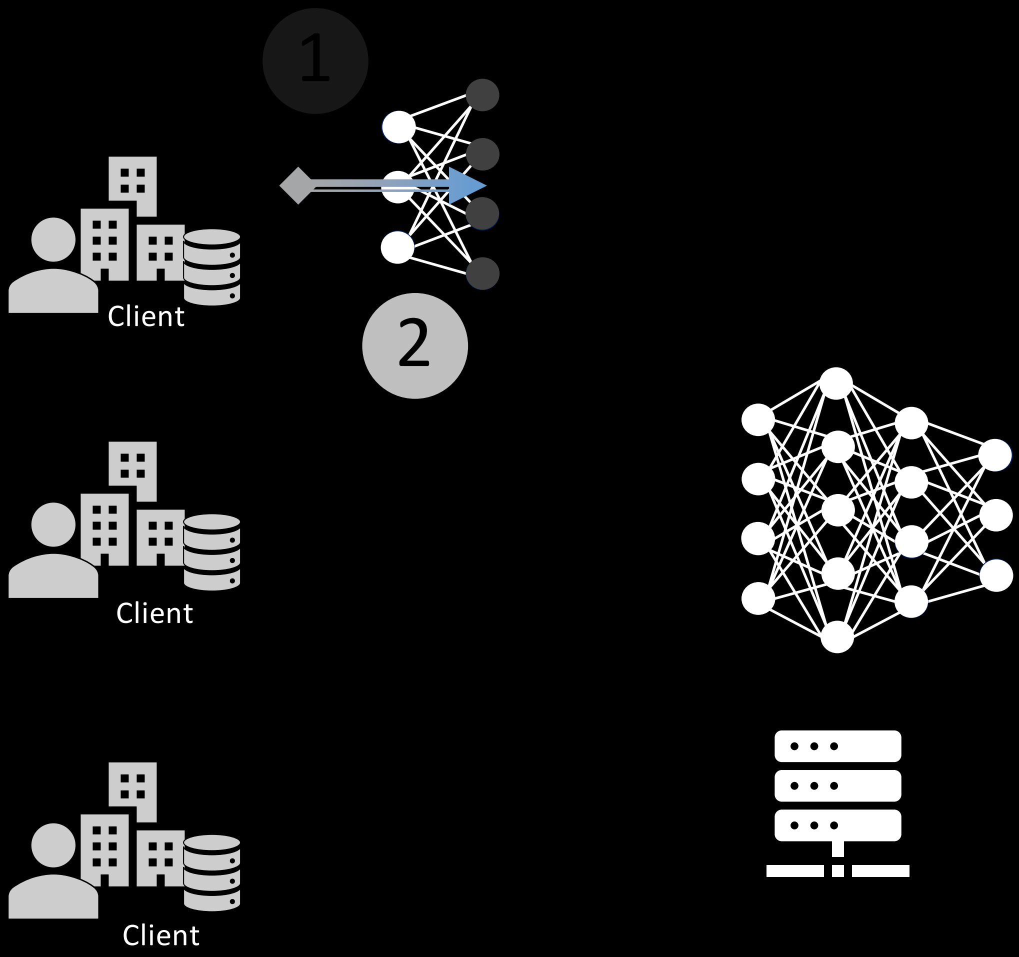


# Split Learning – Concept




1) Split DNN

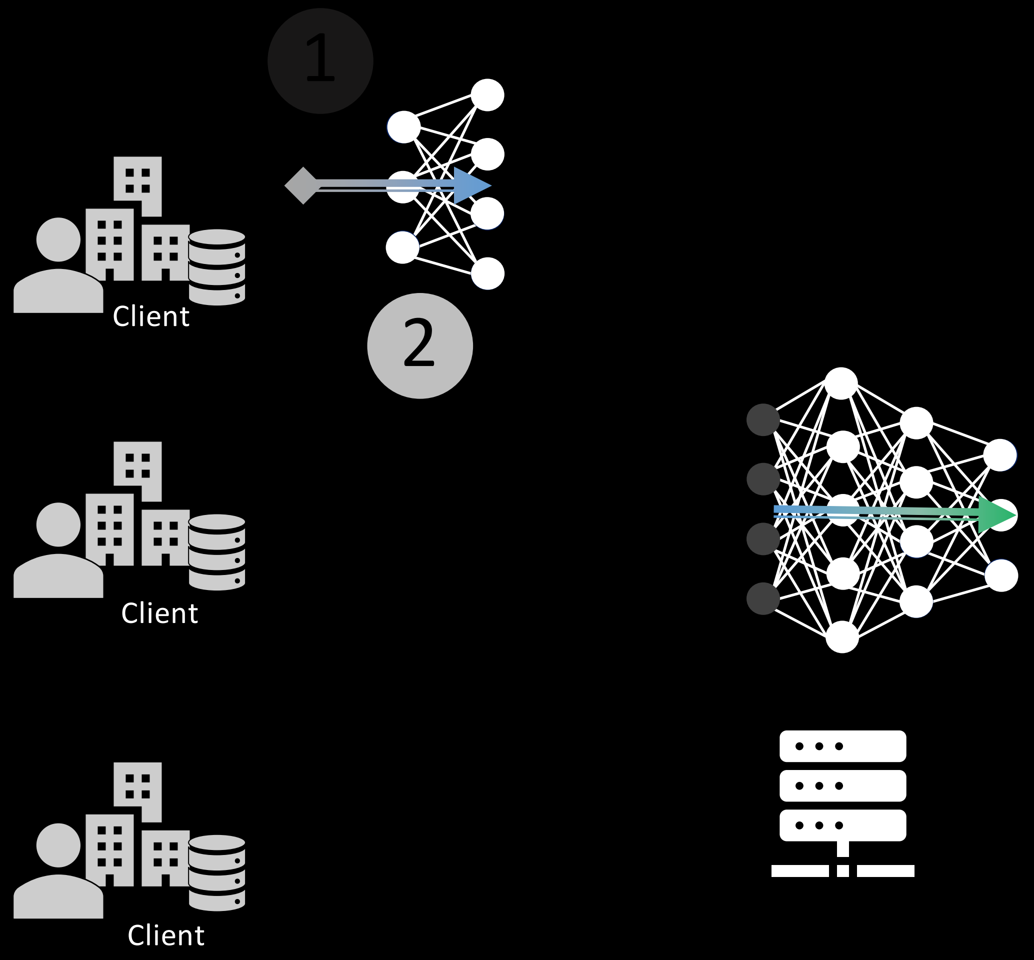
# Split Learning – Concept




1) Split DNN

2) Do local prediction & transmit hidden states

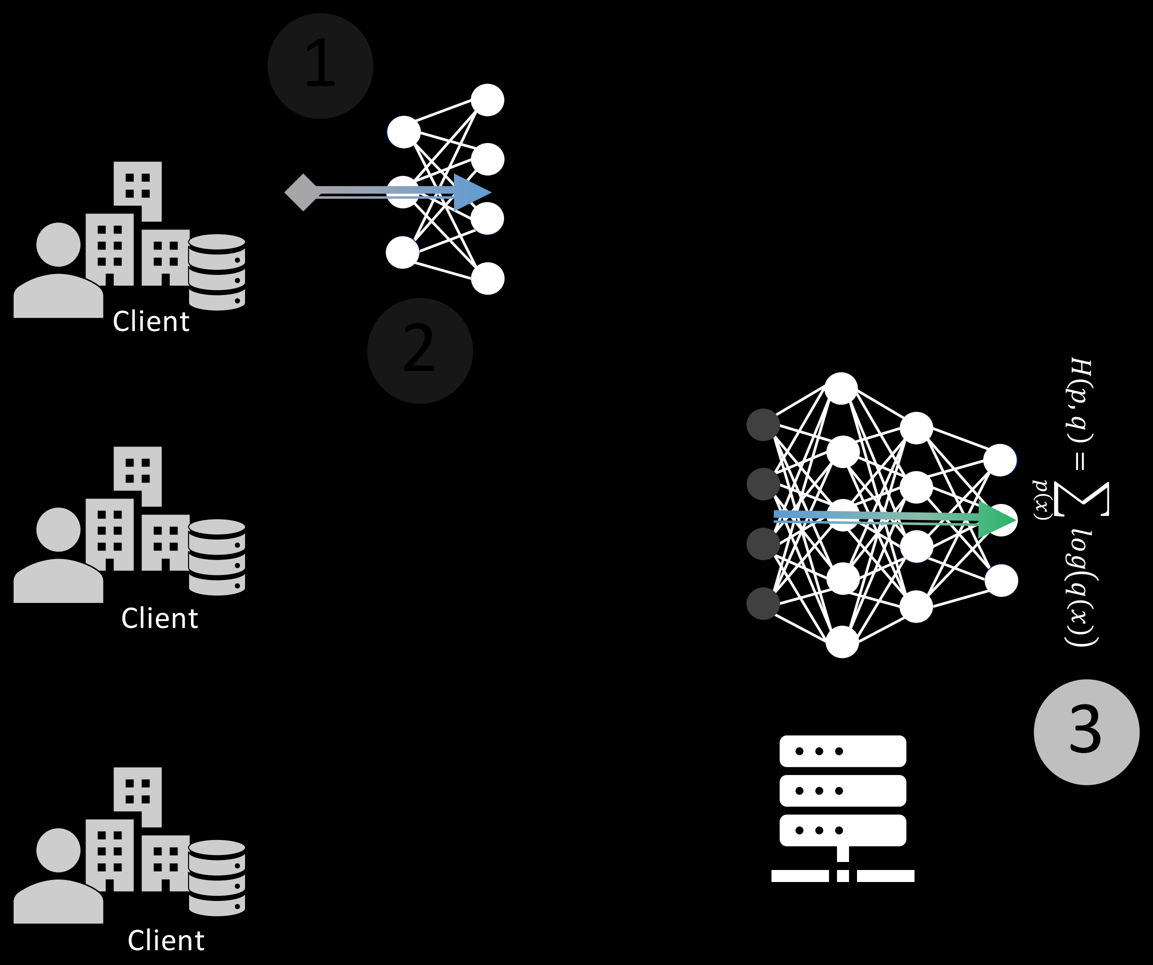
# Split Learning – Concept




1) Split DNN

2) Do local prediction & transmit hidden states

# Split Learning – Concept

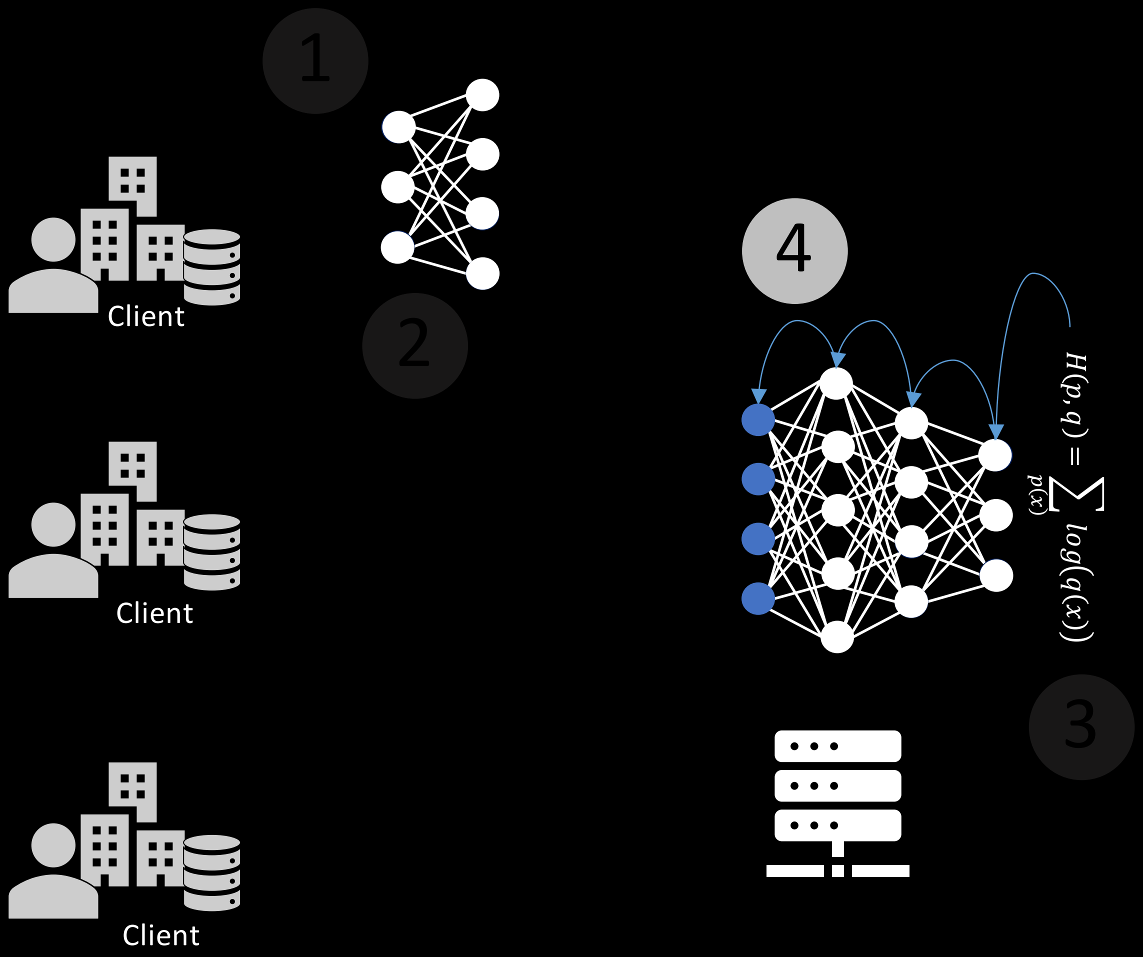



1) Split DNN

2) Do local prediction & transmit hidden states

3) Calculate loss

# Split Learning – Concept



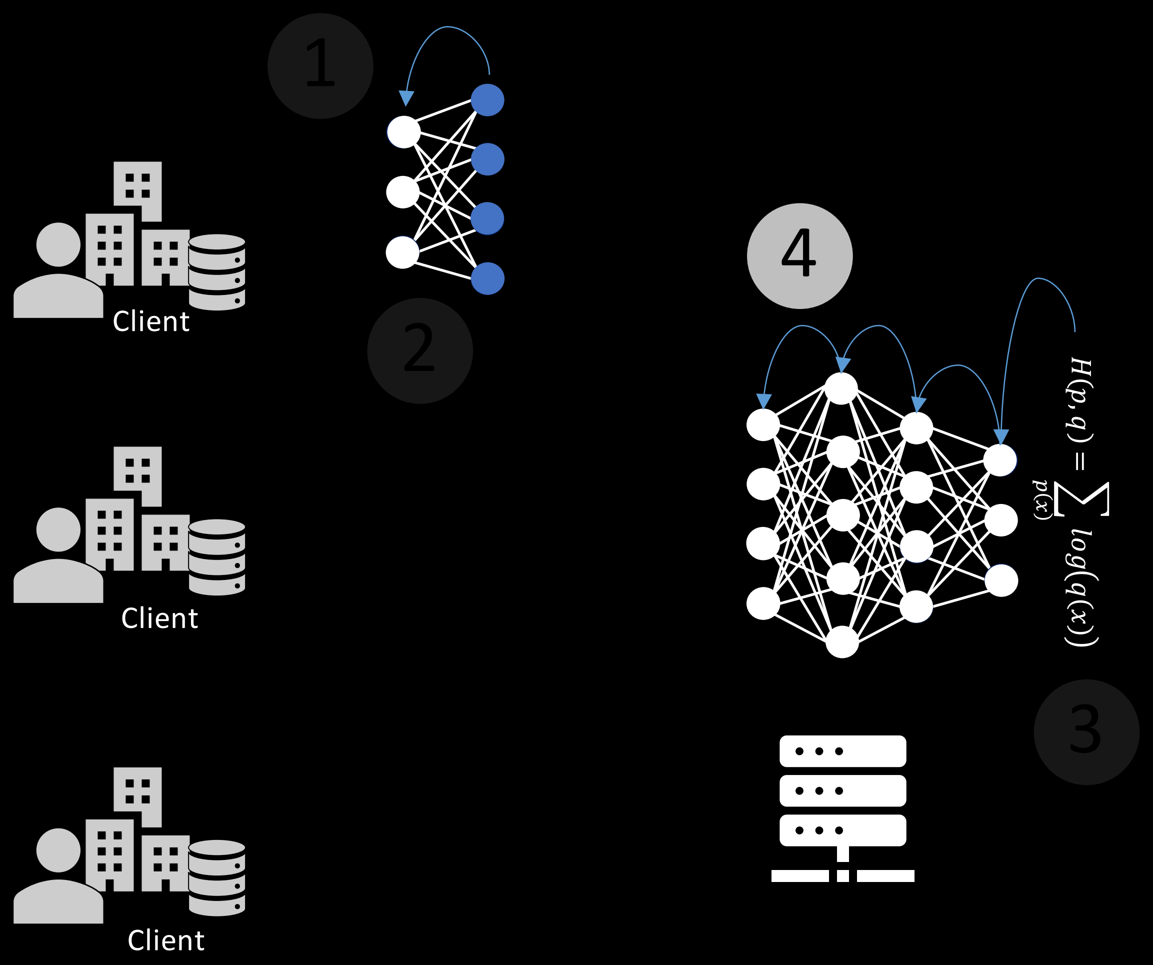

1) Split DNN

2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

# Split Learning – Concept



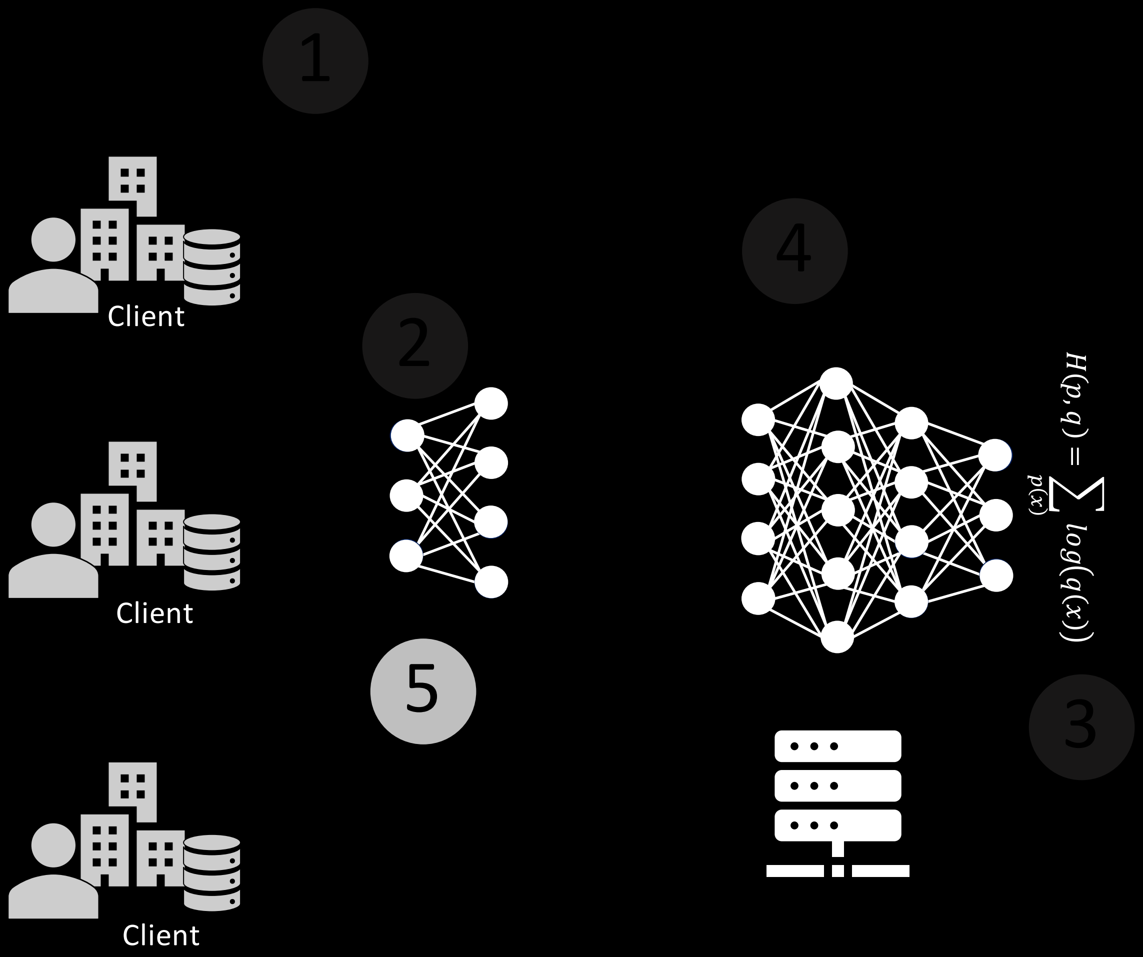

1) Split DNN

2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

# Split Learning – Concept




1) Split DNN

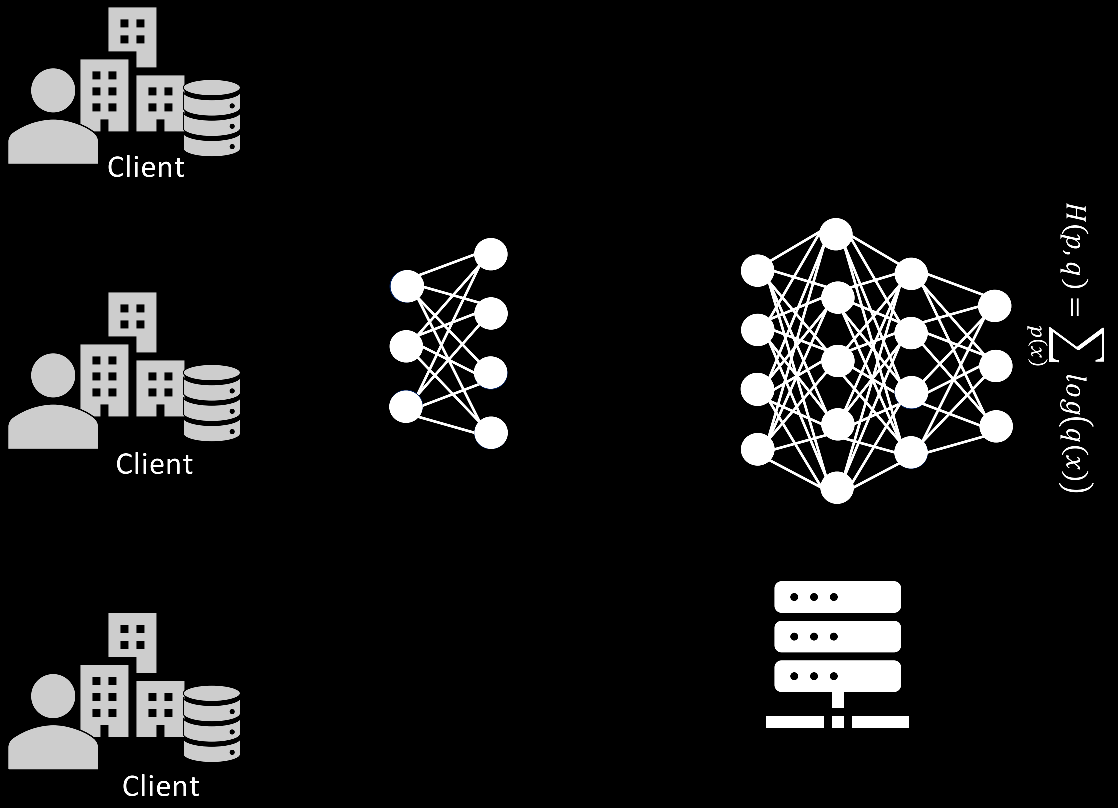
2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

5) Forward client-model

# Split Learning – U-Shape




1) Split DNN

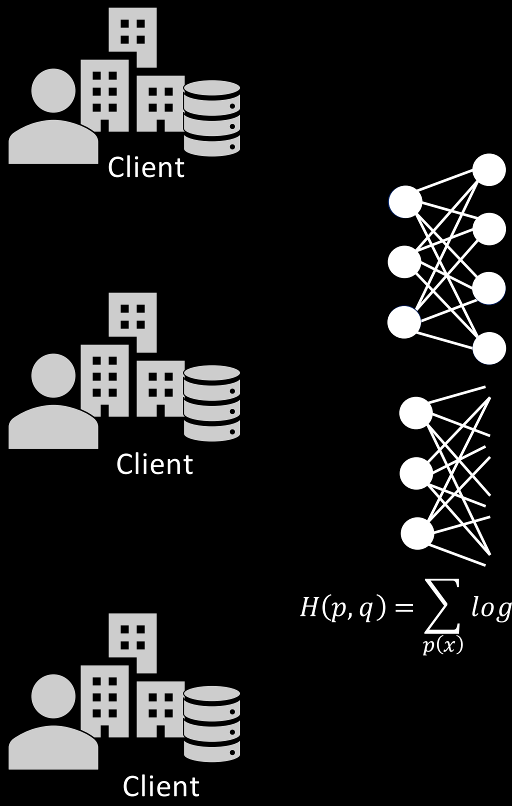
2) Do local prediction & transmit hidden states

3) Calculate loss

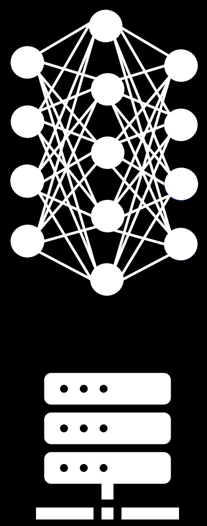
4) Distributed backpropagation

5) Forward client-model

# Split Learning – U-Shape




$$H(p,q) = \sum_{p(x)} log\big(q(x)\big)$$



1) Split DNN

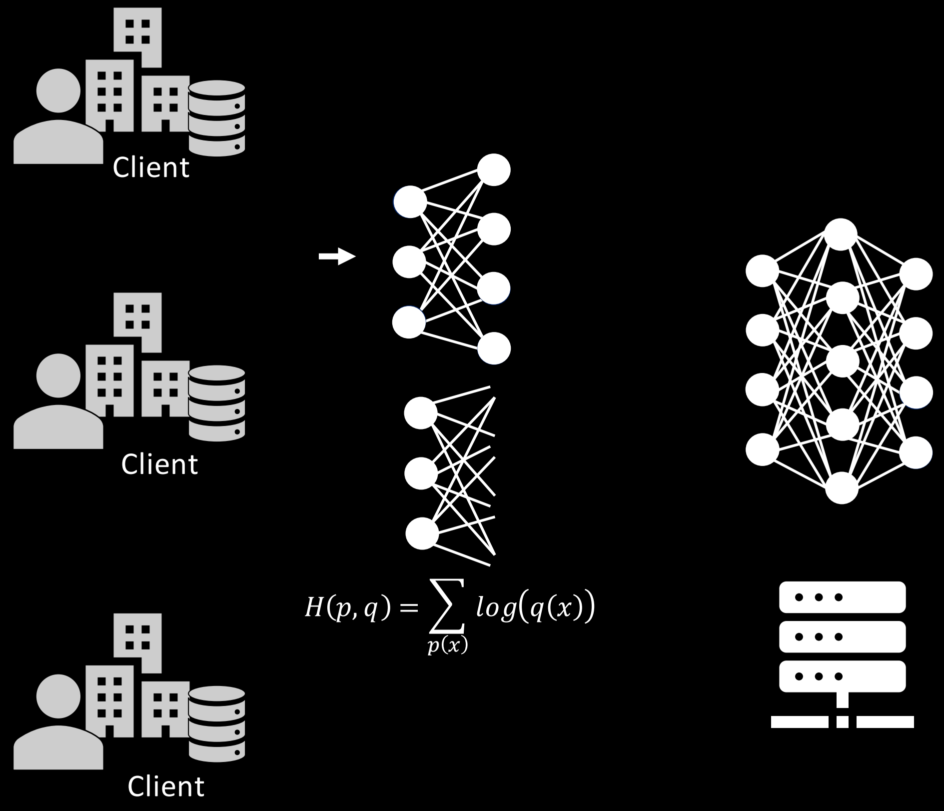
2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

5) Forward client-model

# Split Learning – U-Shape




1) Split DNN

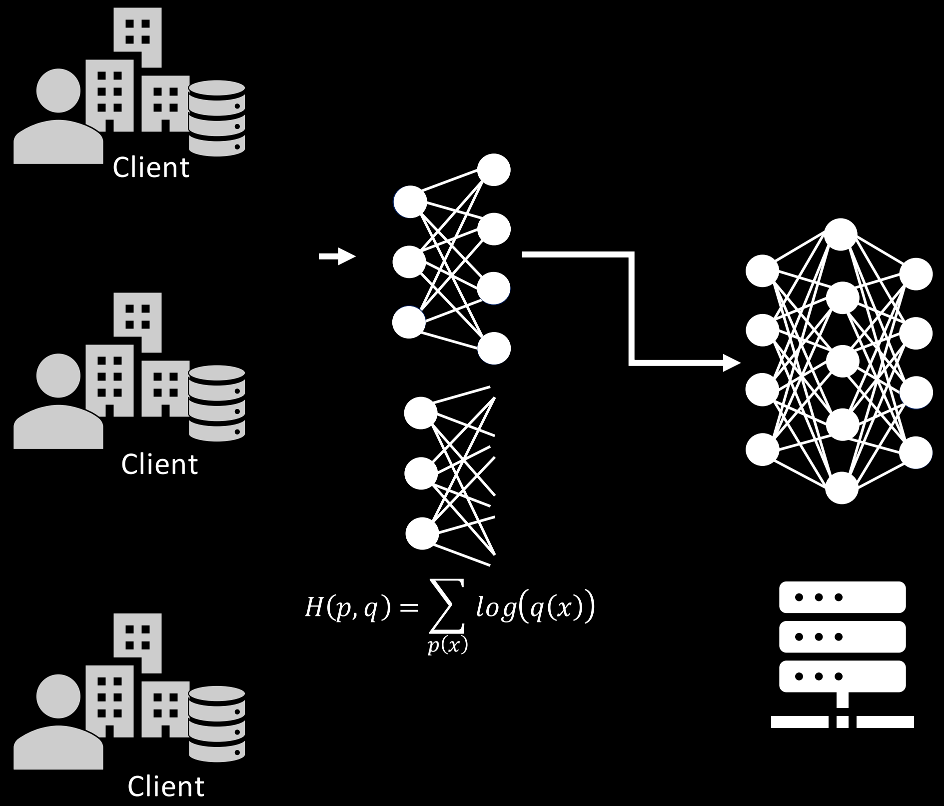
2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

5) Forward client-model

# Split Learning – U-Shape




1) Split DNN

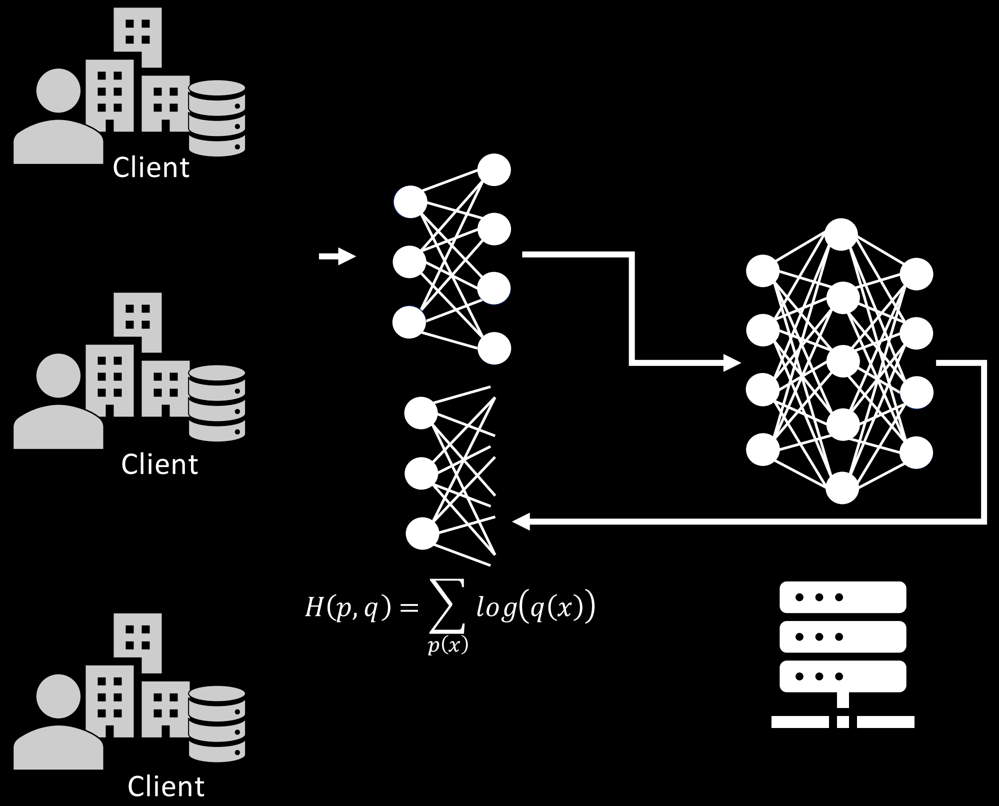
2) Do local prediction & transmit hidden states

3) Calculate loss

4) Distributed backpropagation

5) Forward client-model

# Split Learning – U-Shape




1) Split DNN

2) Do local prediction & transmit hidden states

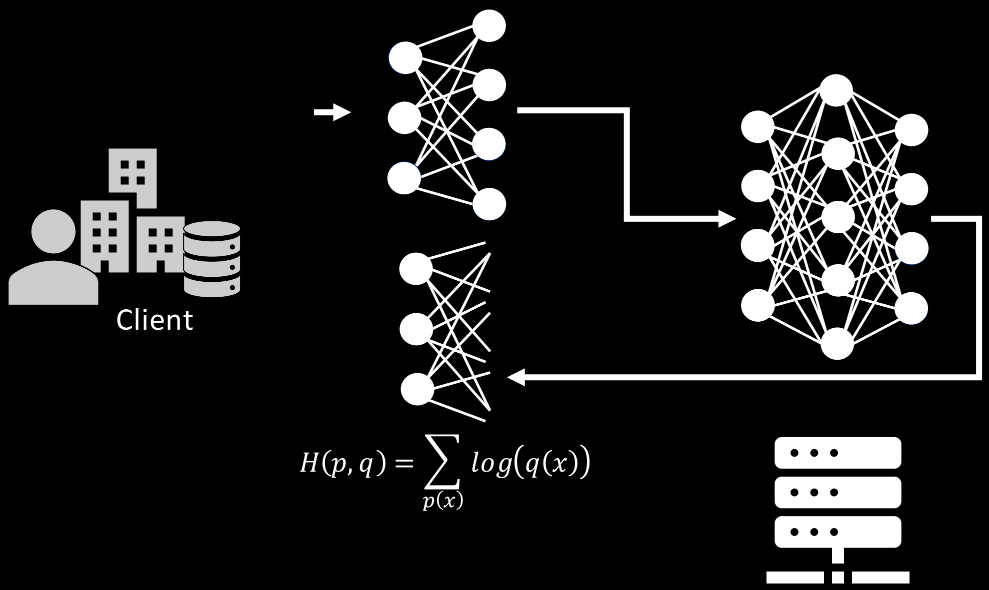
3) Calculate loss

4) Distributed backpropagation

5) Forward client-model

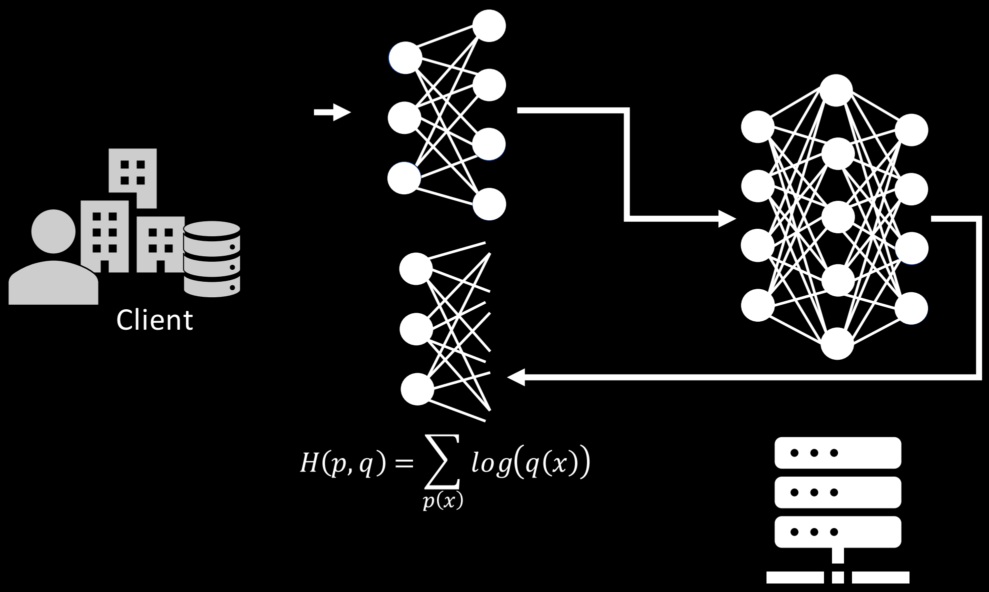
# Split Learning – U-Shape

Advantage: Improved Privacy, Labels remain on client Side




$$H(p,q) = \sum_{p(x)} log\big(q(x)\big)$$

# Split Learning – U-Shape




$$H(p,q) = \sum_{p(x)} log\big(q(x)\big)$$

Advantage: Improved Privacy, Labels remain on client Side

Challenge: Malicious clients can tamper loss function

# Backdoor Example

- Trigger: Pixel-pattern
  [Bagdasaryan et al. AISTATS 2020]

Trigger: Pixel-pattern
Target Label: Bird

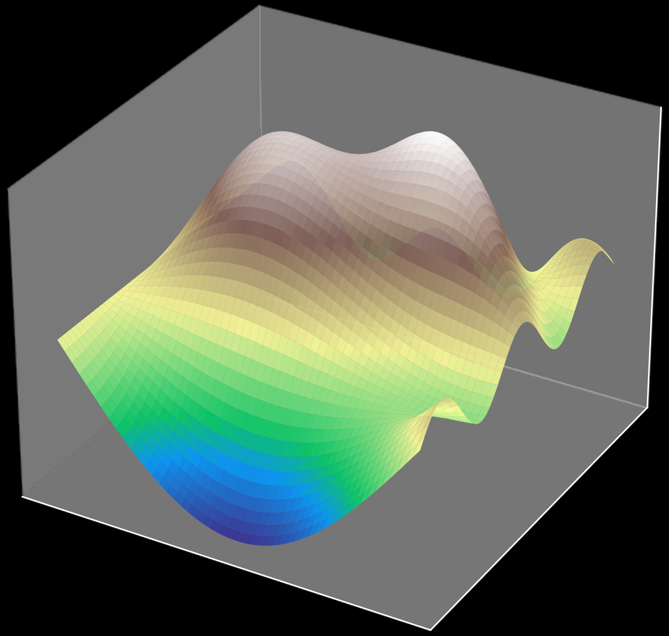
# Backdoor Example

- Trigger: Pixel-pattern
  [Bagdasaryan et al. AISTATS 2020]

Trigger: Pixel-pattern
Target Label: Bird

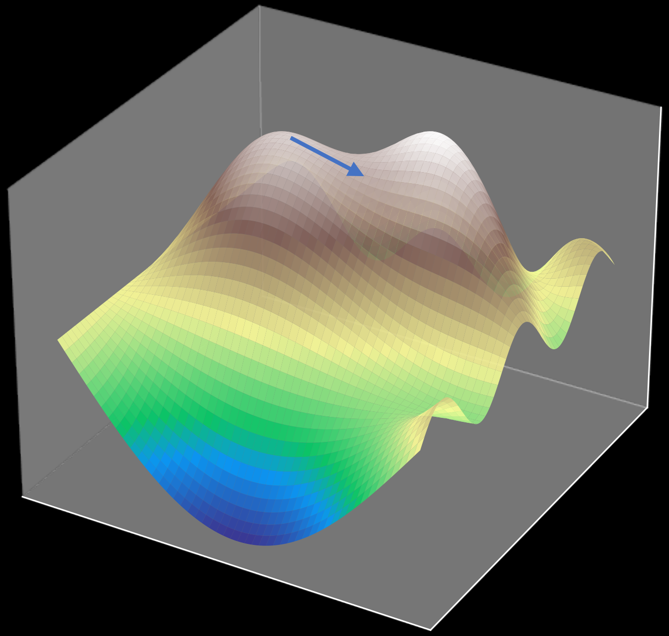
No Trigger
Label: Car

# Security Challenges in Split Learning



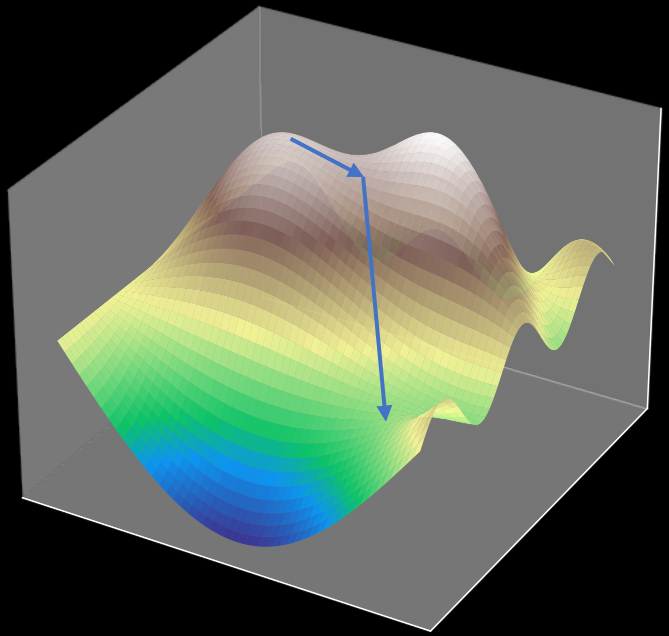
Centralized Learning

# Security Challenges in Split Learning



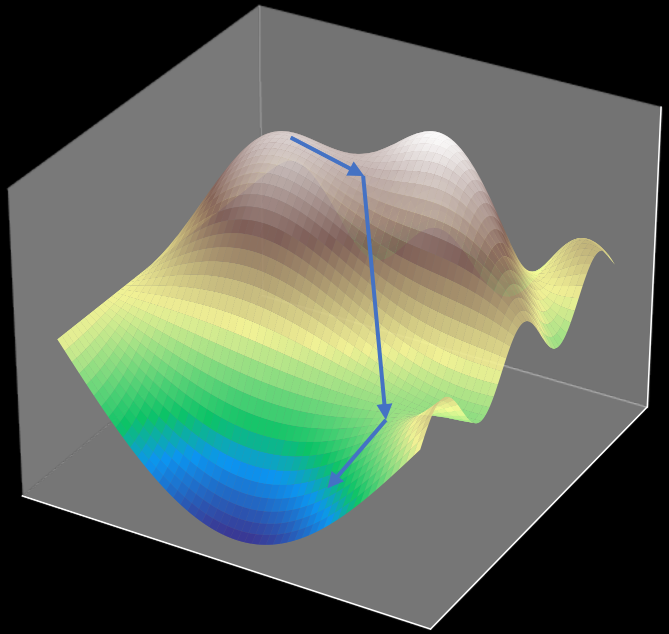
Centralized Learning

# Security Challenges in Split Learning



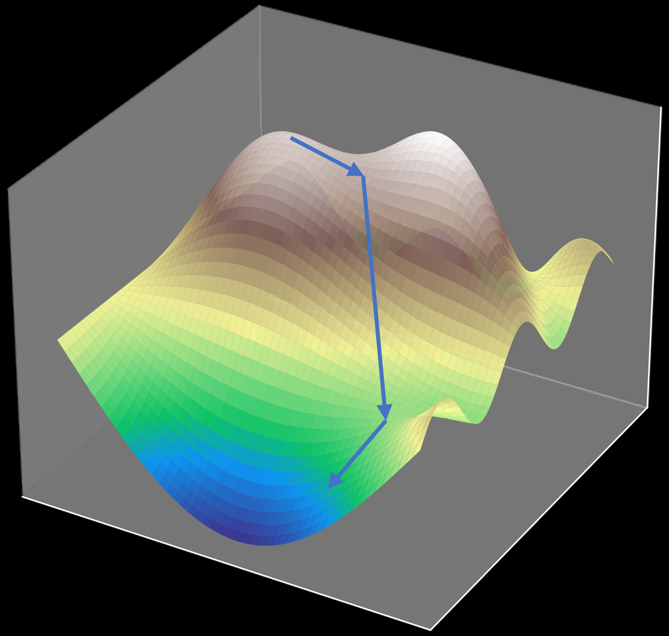
Centralized Learning

# Security Challenges in Split Learning

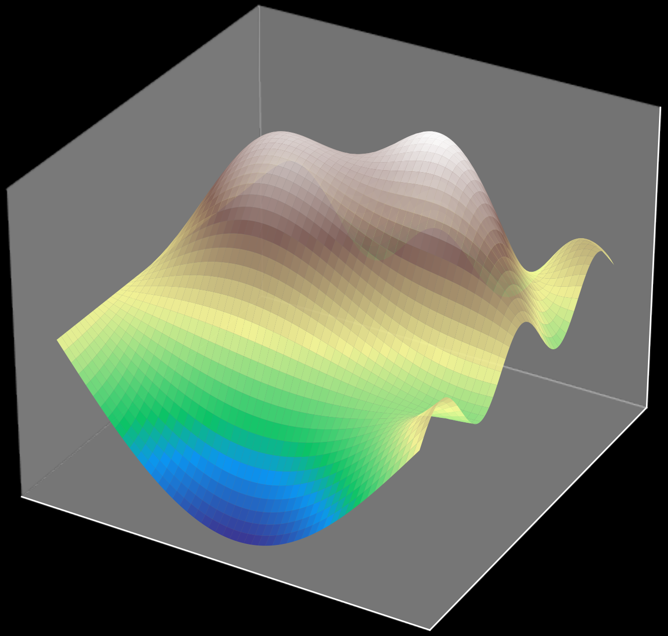


Centralized Learning

# Security Challenges in Split Learning

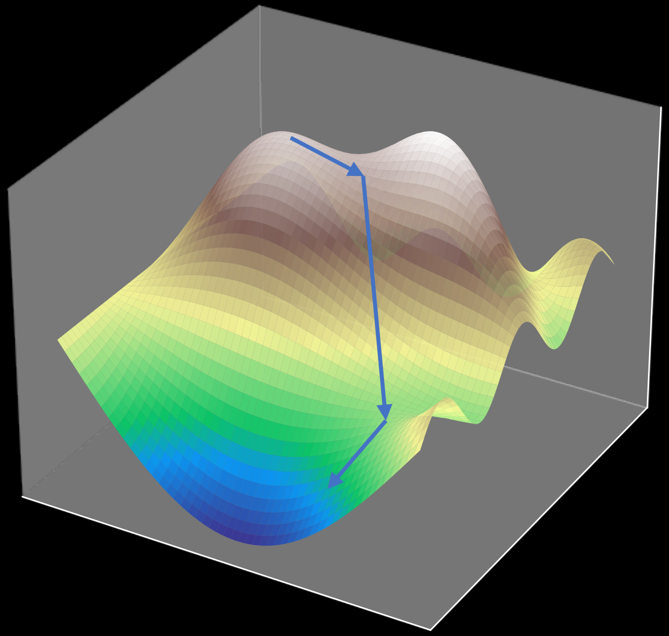


Centralized Learning

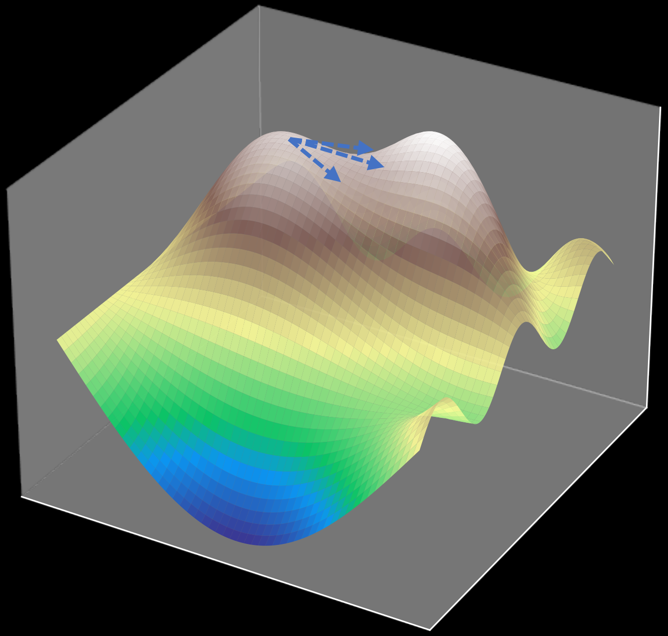


Federated Learning

# Security Challenges in Split Learning

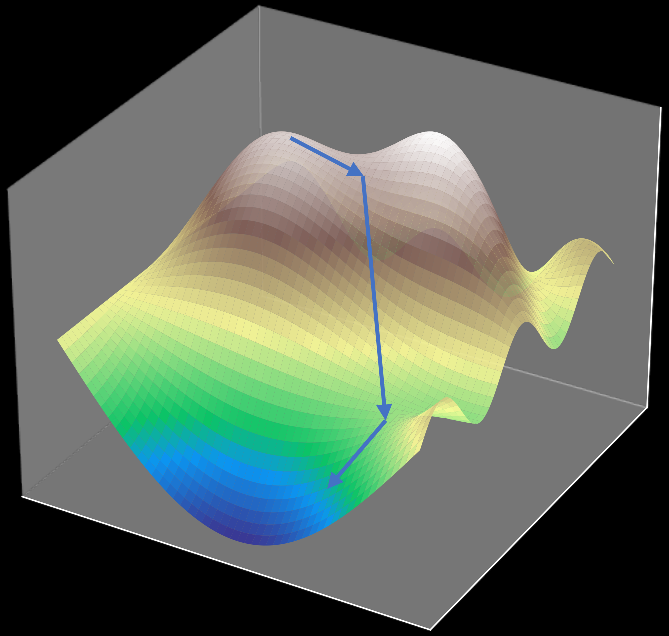


Centralized Learning

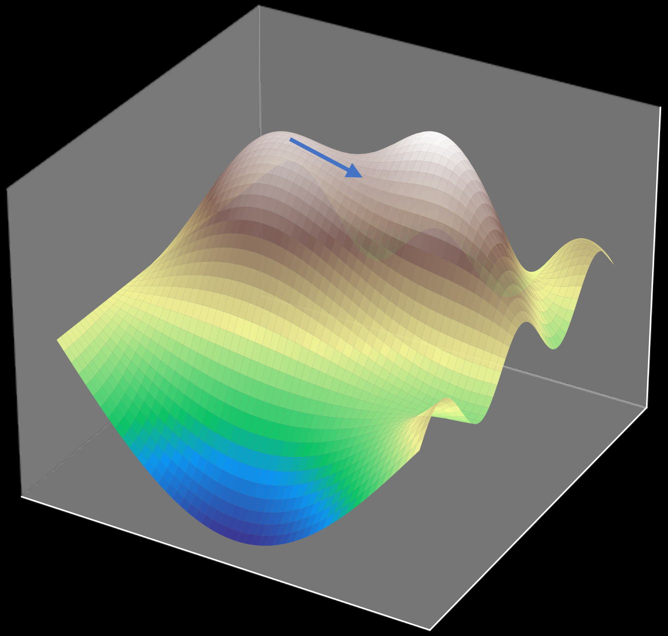


Federated Learning

# Security Challenges in Split Learning

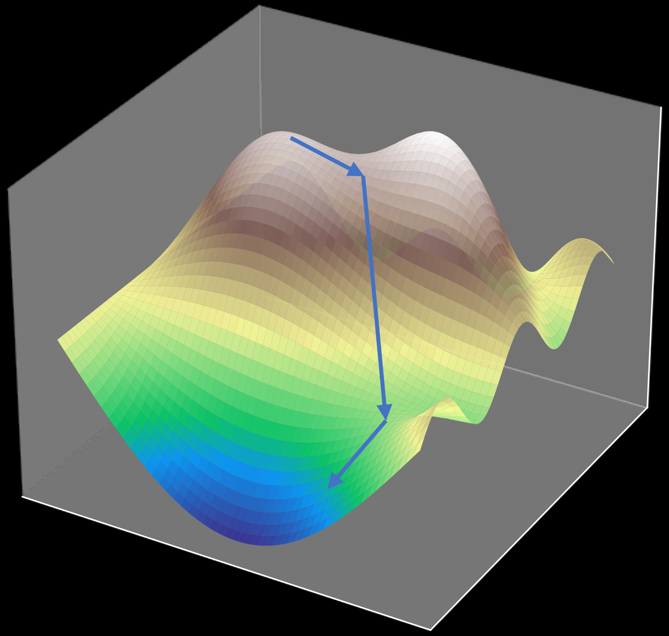


Centralized Learning

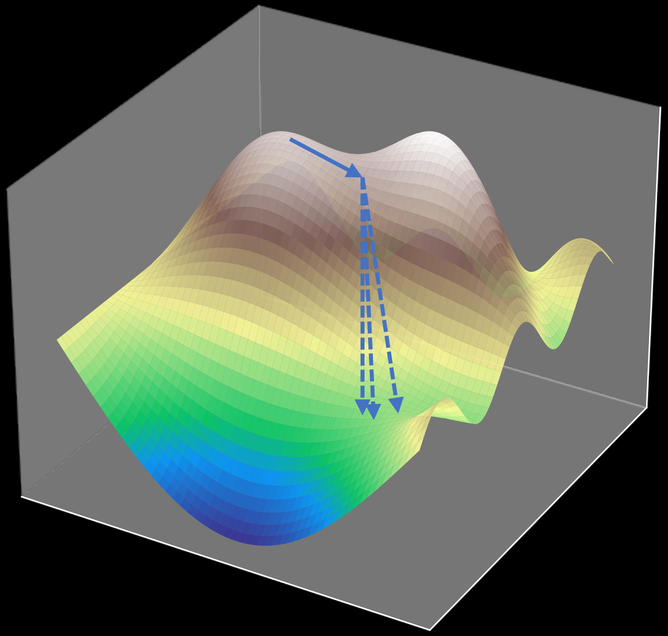


Federated Learning

# Security Challenges in Split Learning

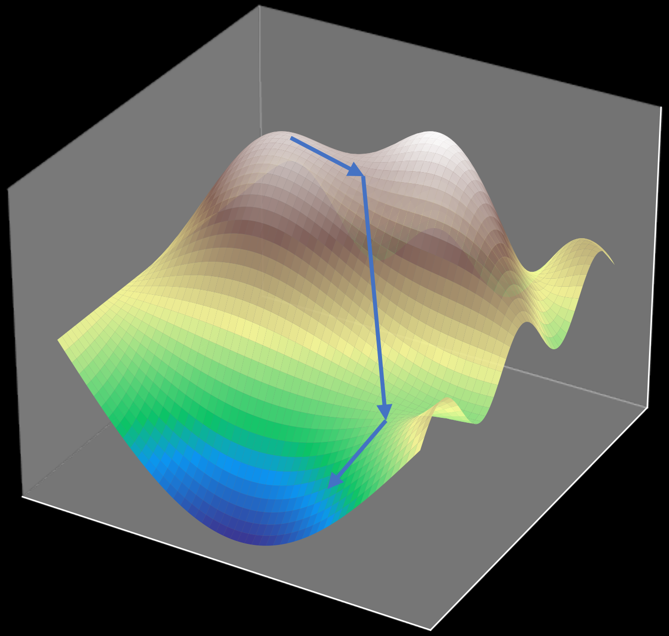


Centralized Learning

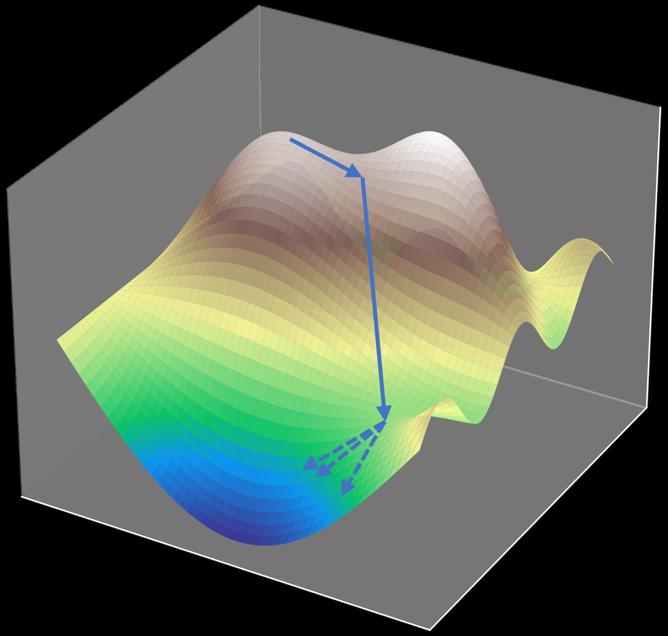


Federated Learning

# Security Challenges in Split Learning

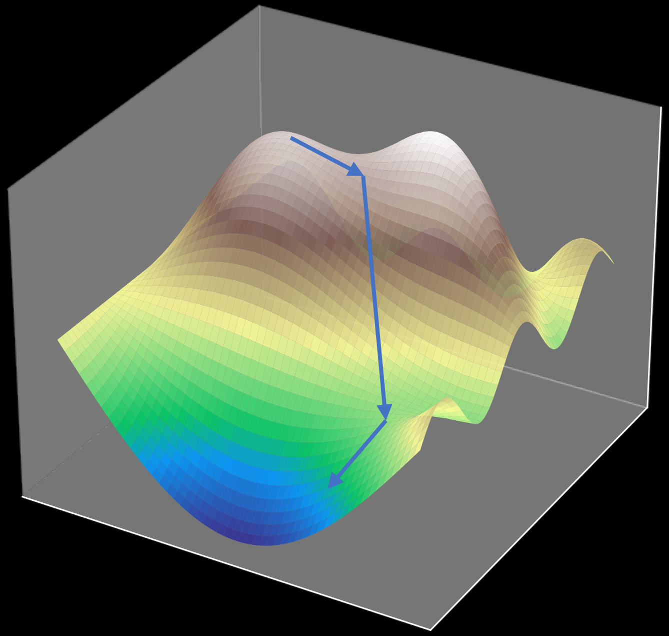


Centralized Learning

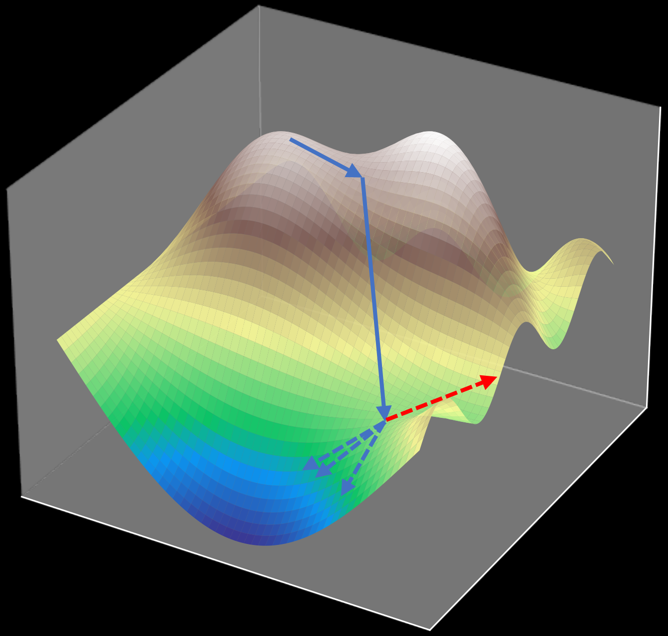


Federated Learning

# Security Challenges in Split Learning

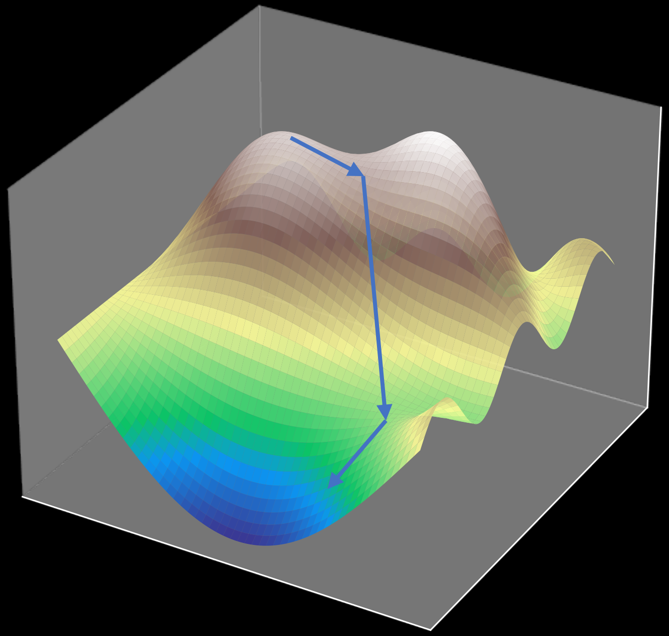


Centralized Learning

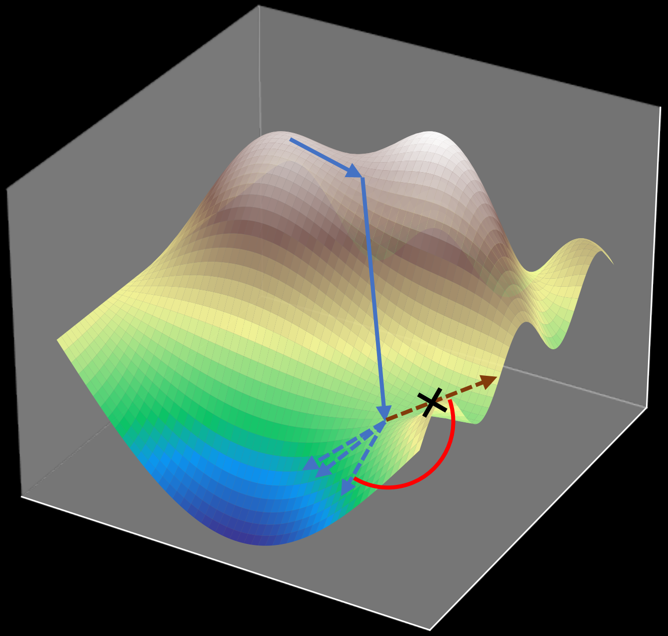


Federated Learning

# Security Challenges in Split Learning

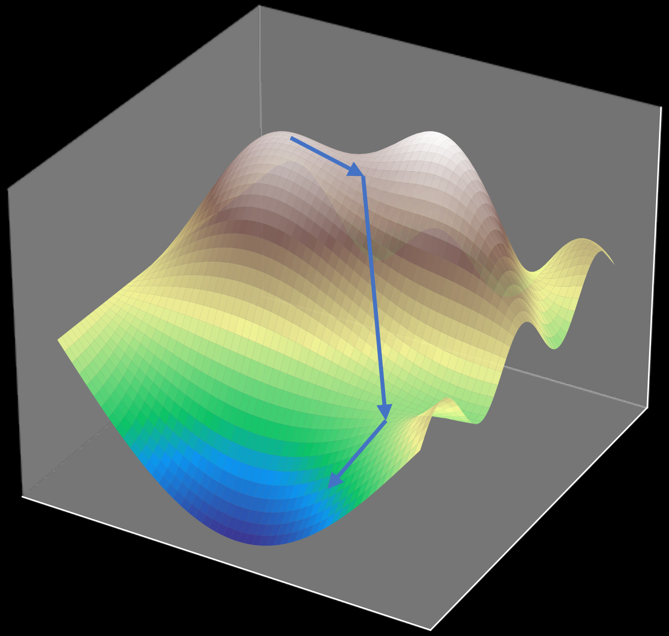


Centralized Learning

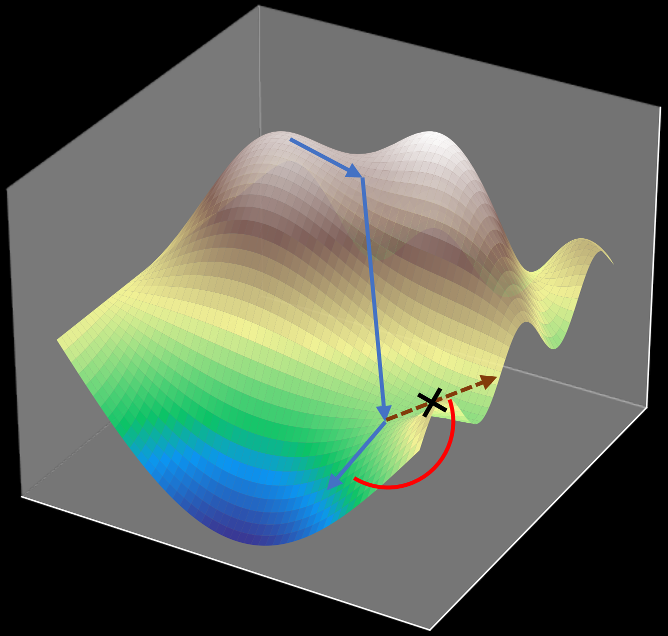


Federated Learning

# Security Challenges in Split Learning

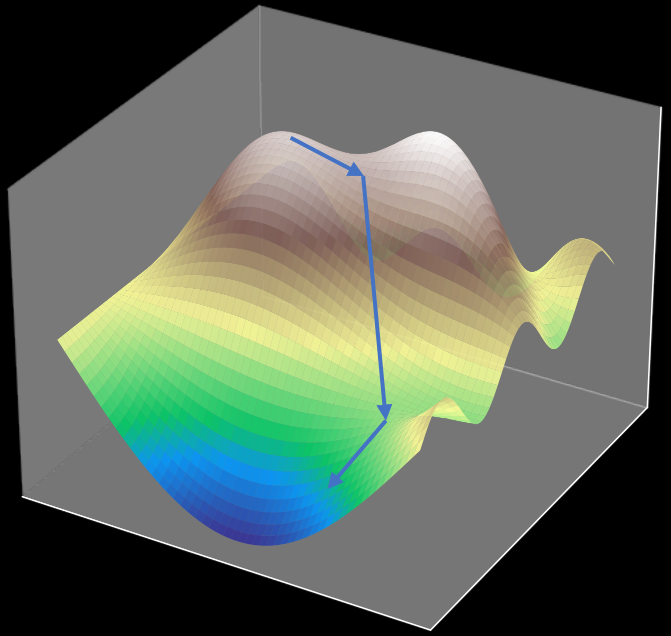


Centralized Learning

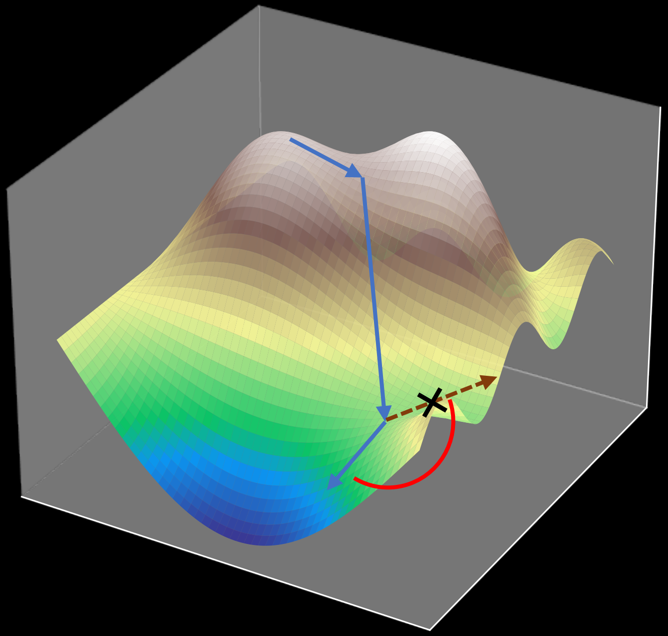


Federated Learning

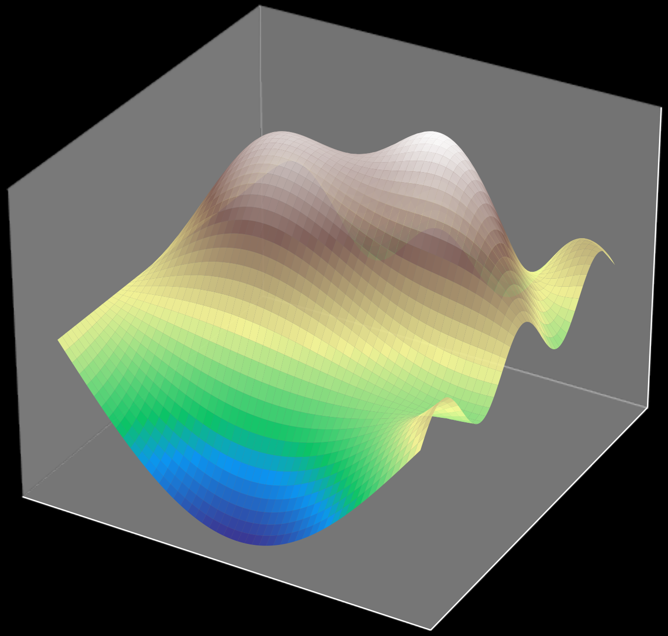
# Security Challenges in Split Learning



Centralized Learning

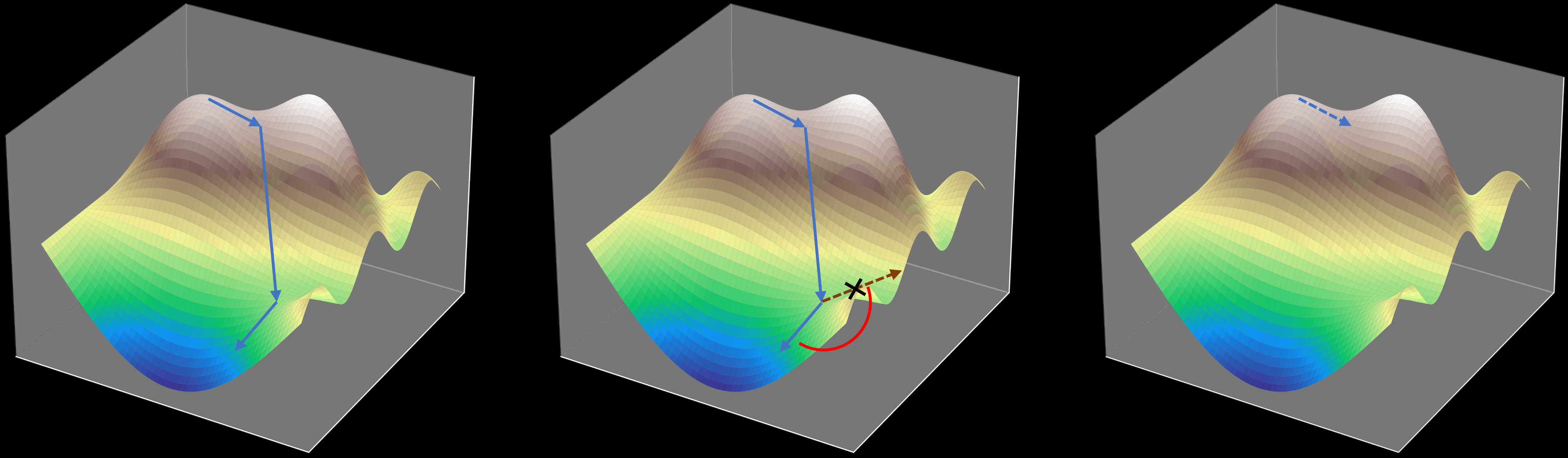


Federated Learning



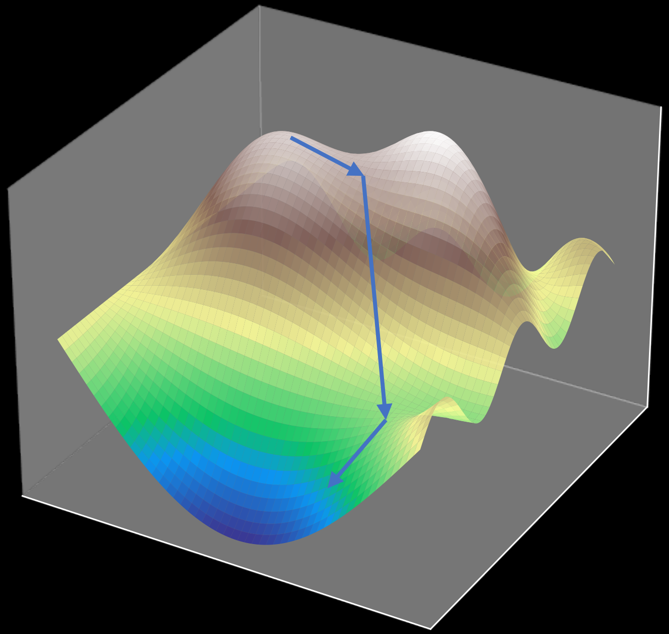
Split Learning

# Security Challenges in Split Learning

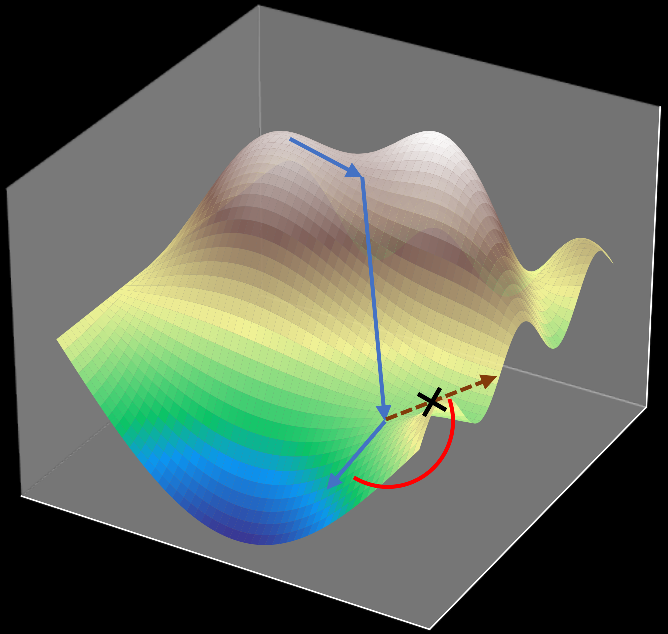


Centralized Learning Federated Learning Split Learning

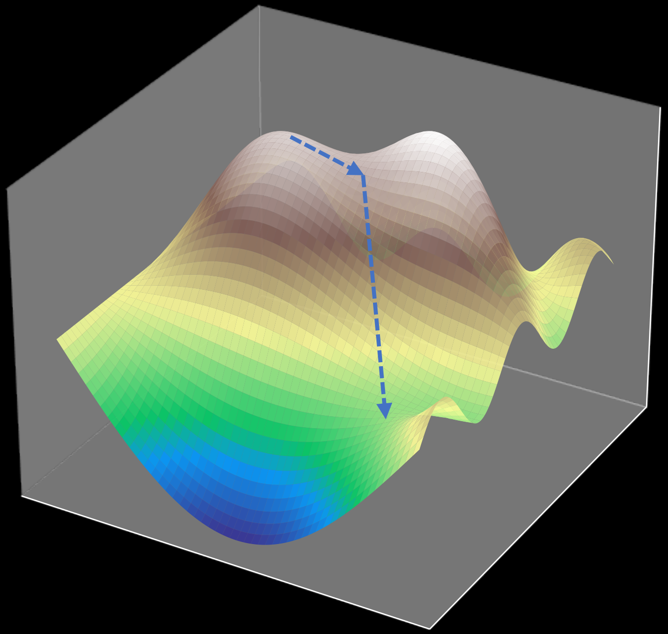
# Security Challenges in Split Learning



Centralized Learning

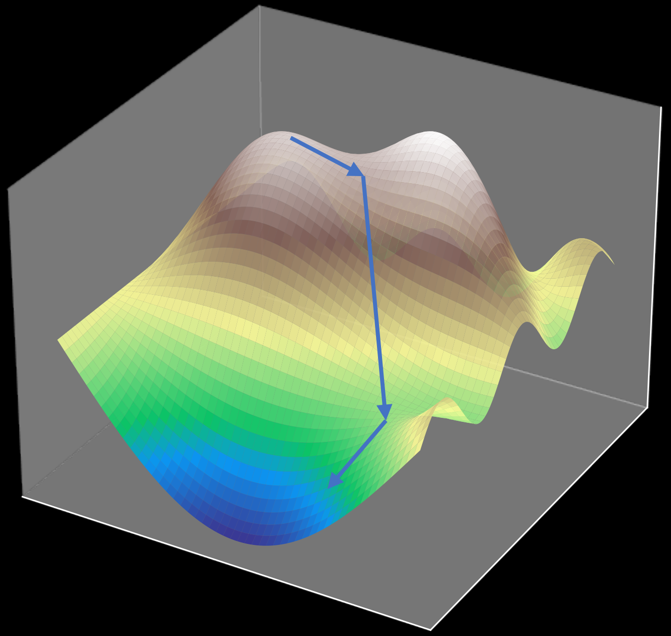


Federated Learning

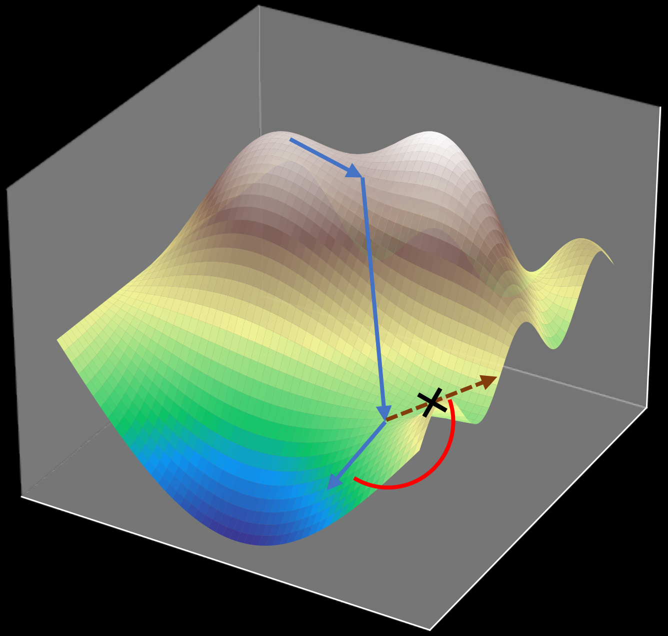


Split Learning

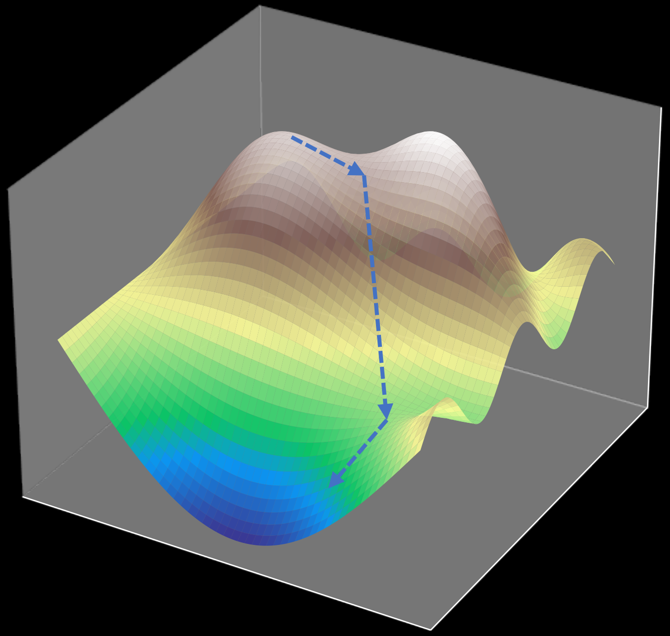
# Security Challenges in Split Learning



Centralized Learning

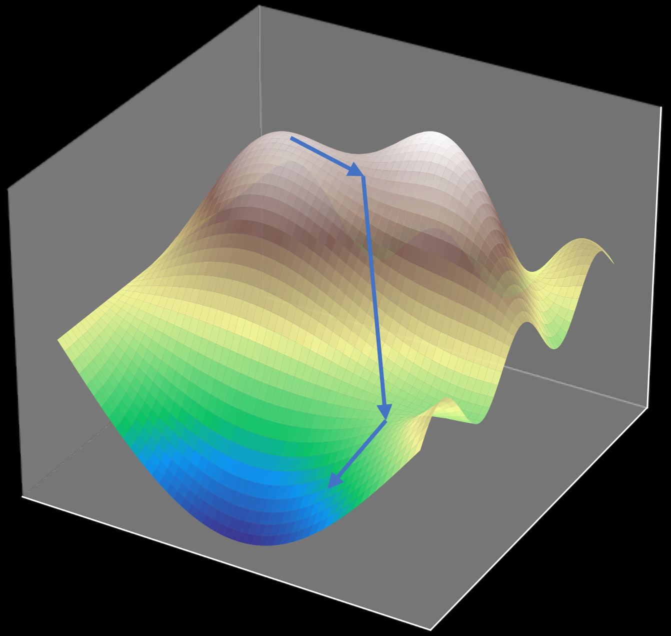


Federated Learning

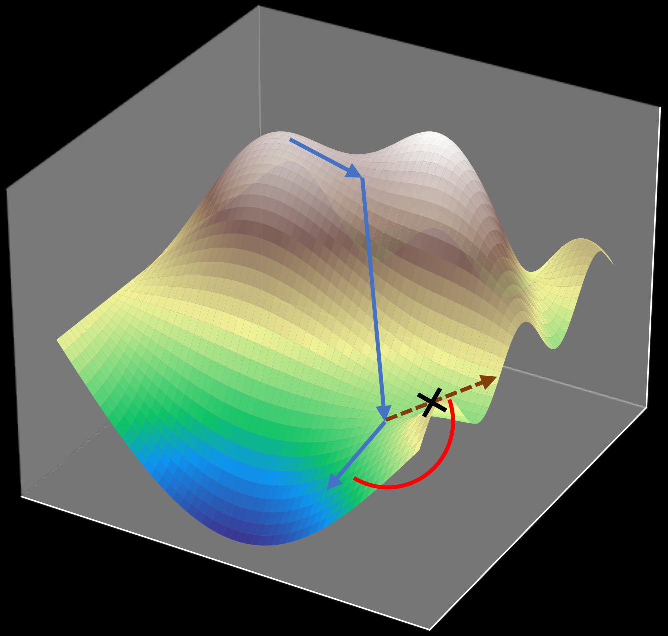


Split Learning

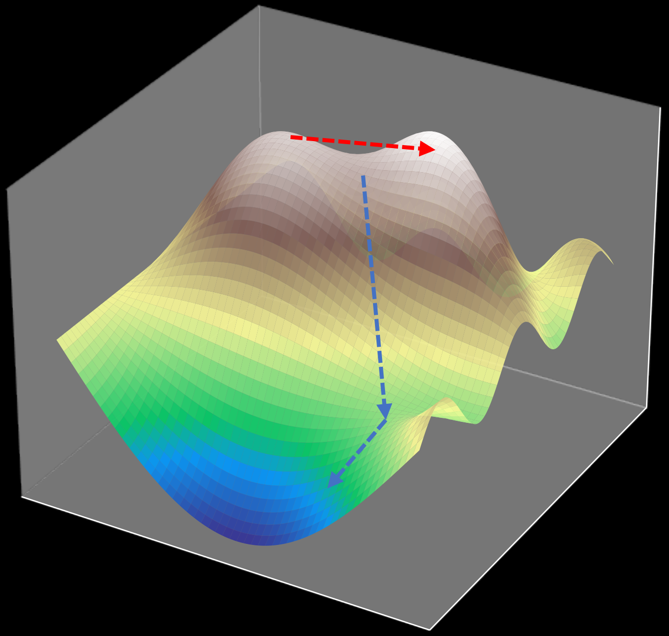
# Security Challenges in Split Learning



Centralized Learning

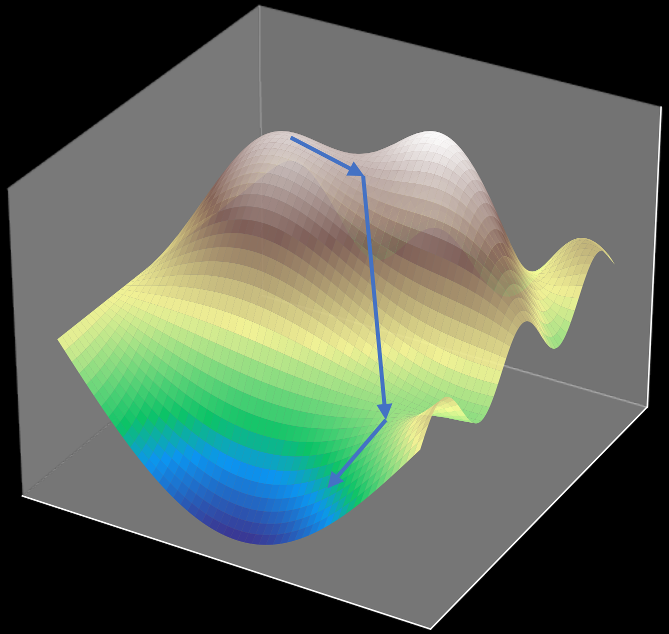


Federated Learning

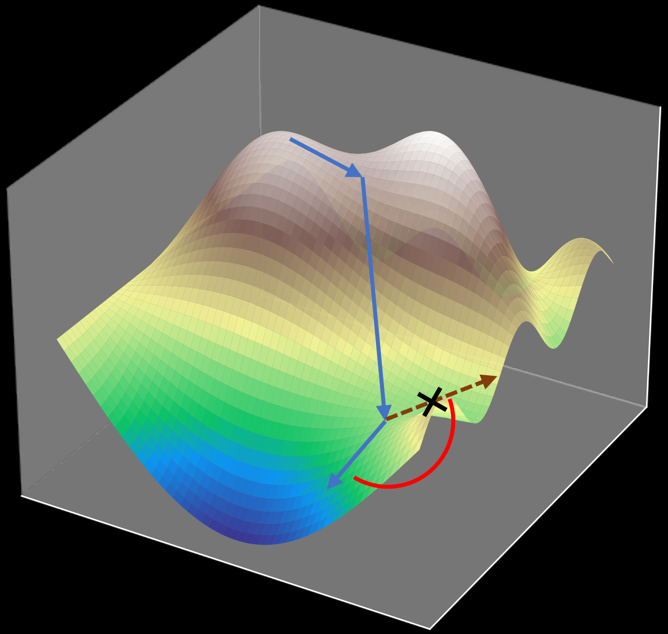


Split Learning

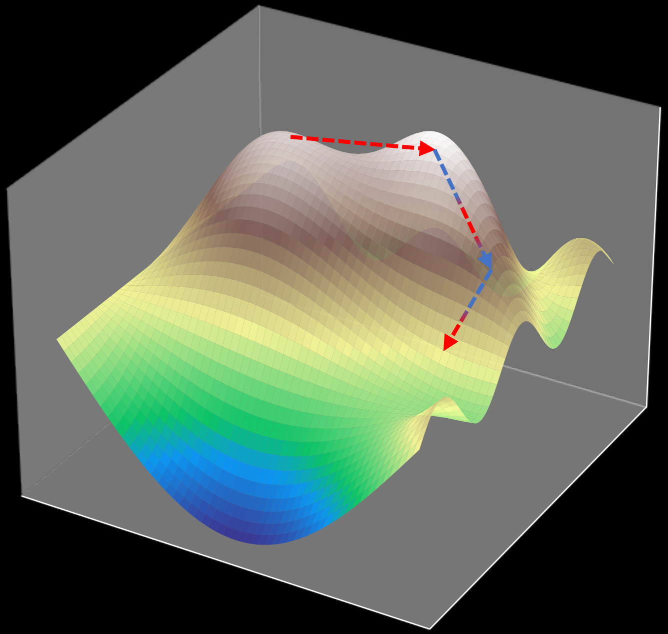
# Security Challenges in Split Learning



Centralized Learning

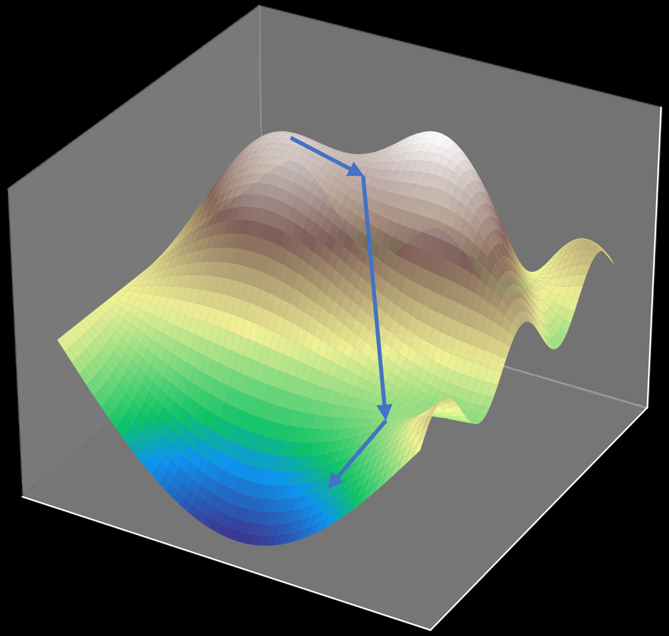


Federated Learning

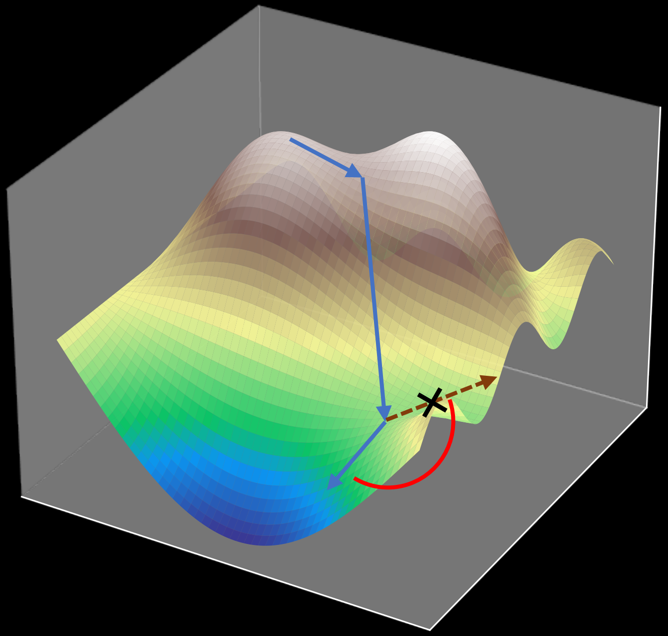


Split Learning

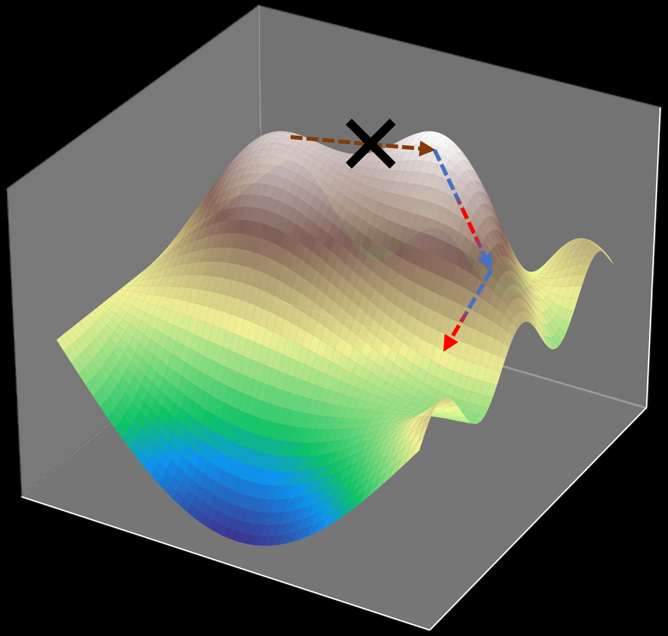
# Security Challenges in Split Learning



Centralized Learning

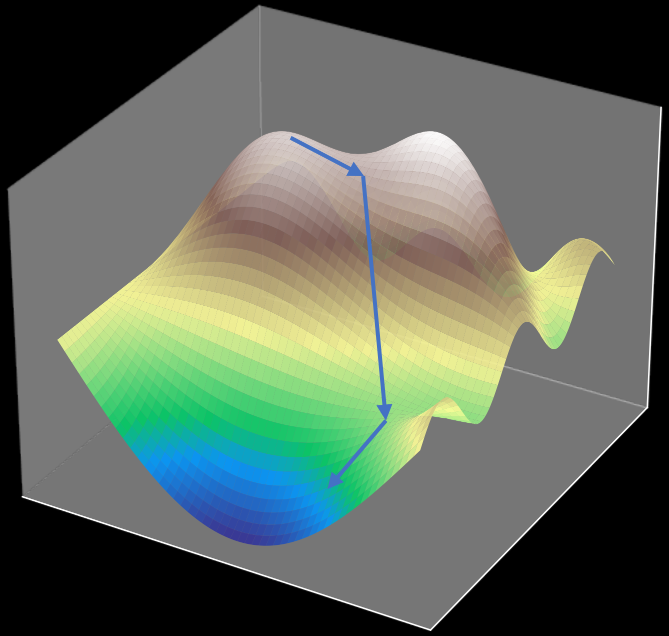


Federated Learning

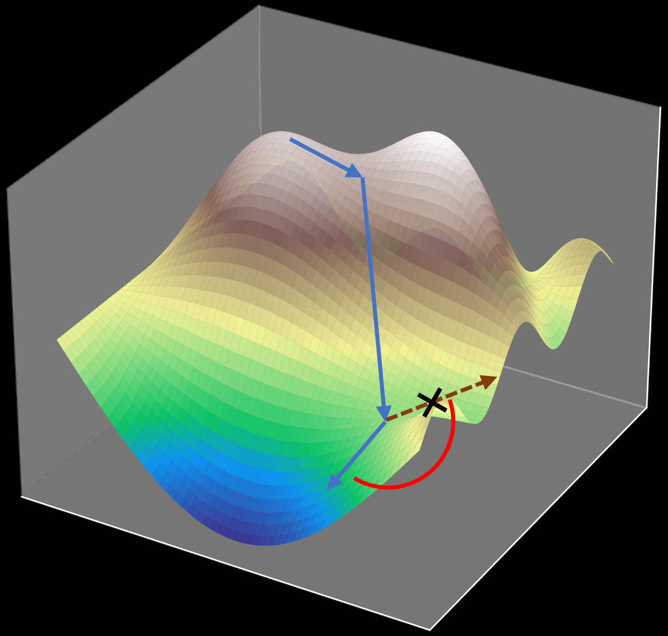


Split Learning

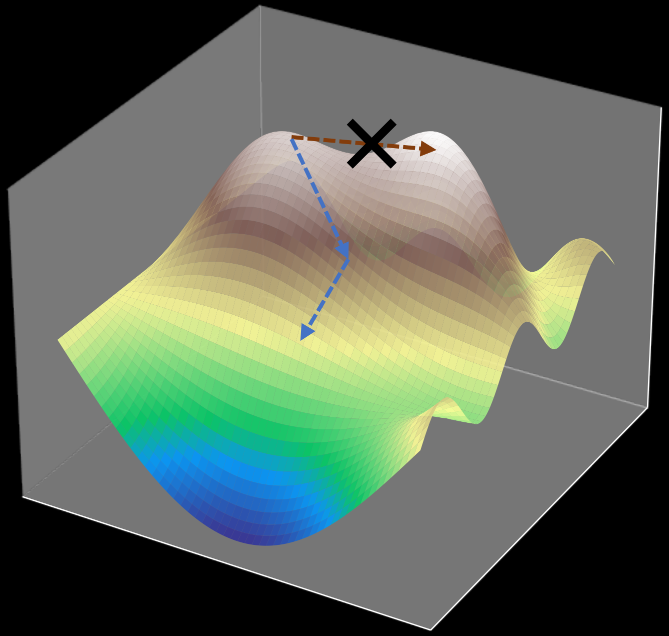
# Security Challenges in Split Learning



Centralized Learning



Federated Learning

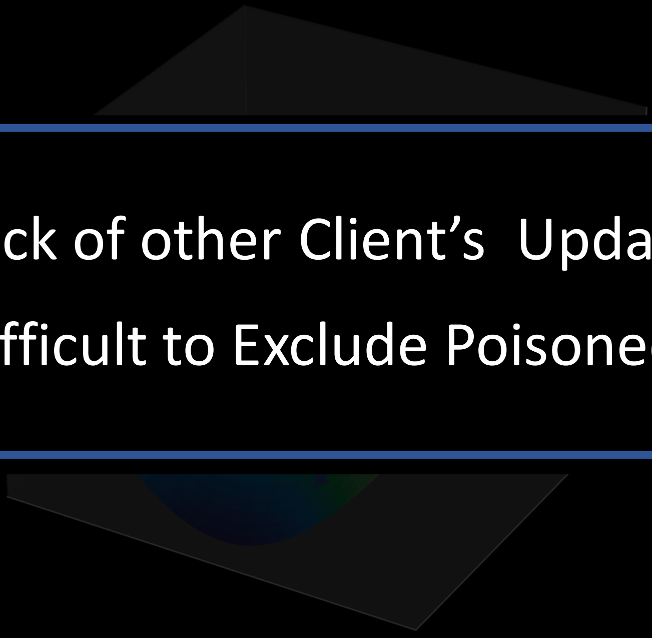


Split Learning

# Security Challenges in Split Learning

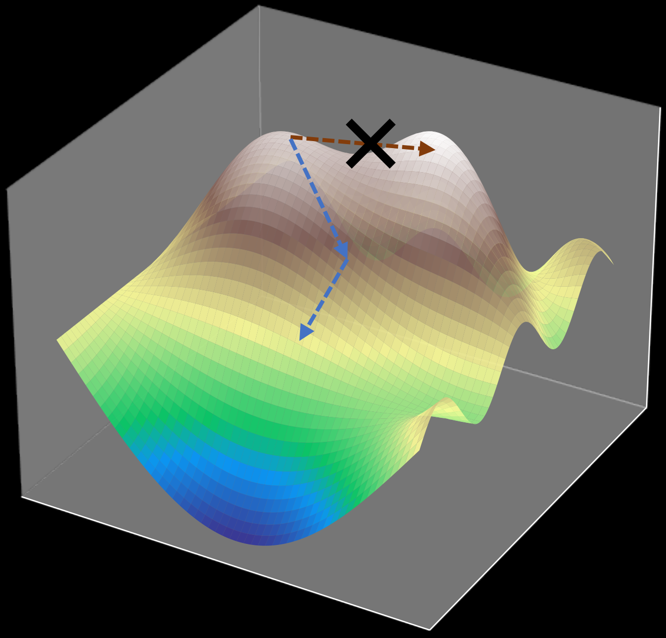
Lack of other Client's  Updates for Comparison

Difficult to Exclude Poisoned Models after Detection



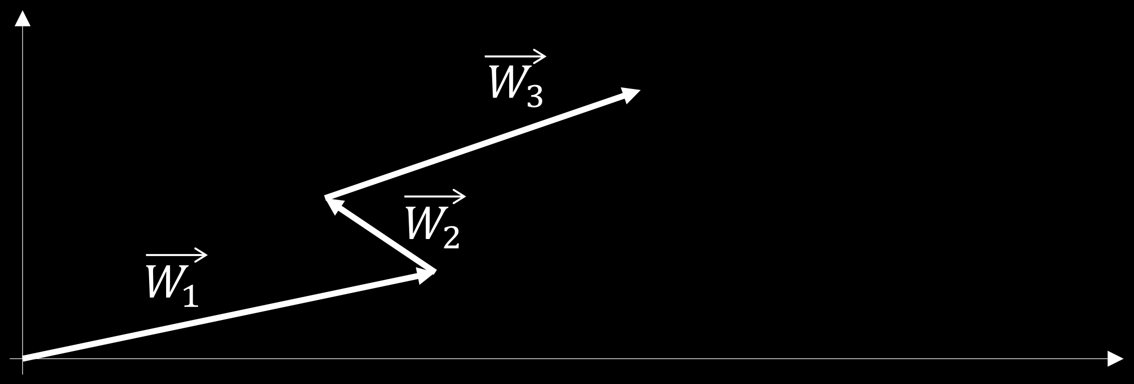
Centralized Learning

Federated Learning



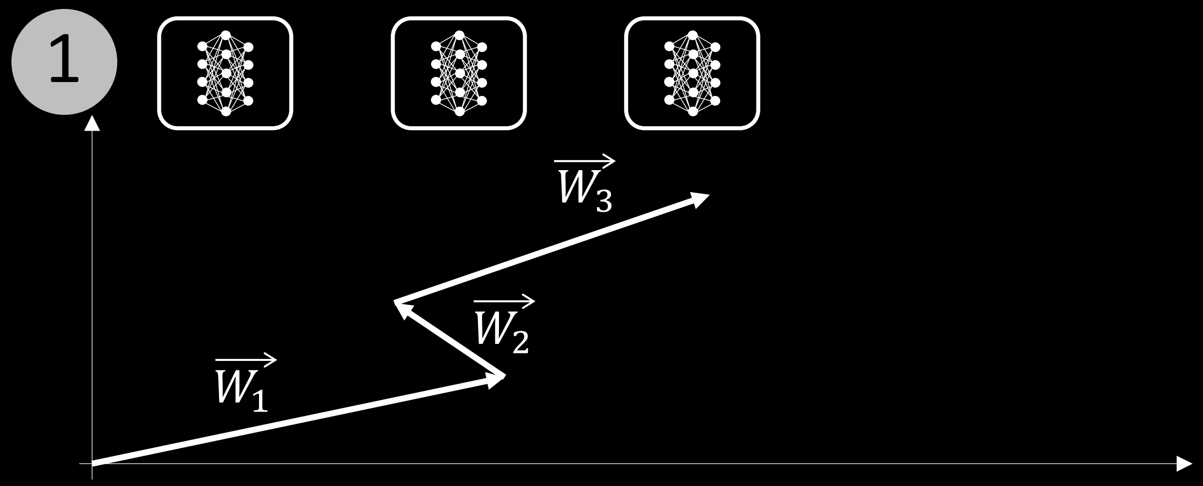
Split Learning

# Circular Analysis



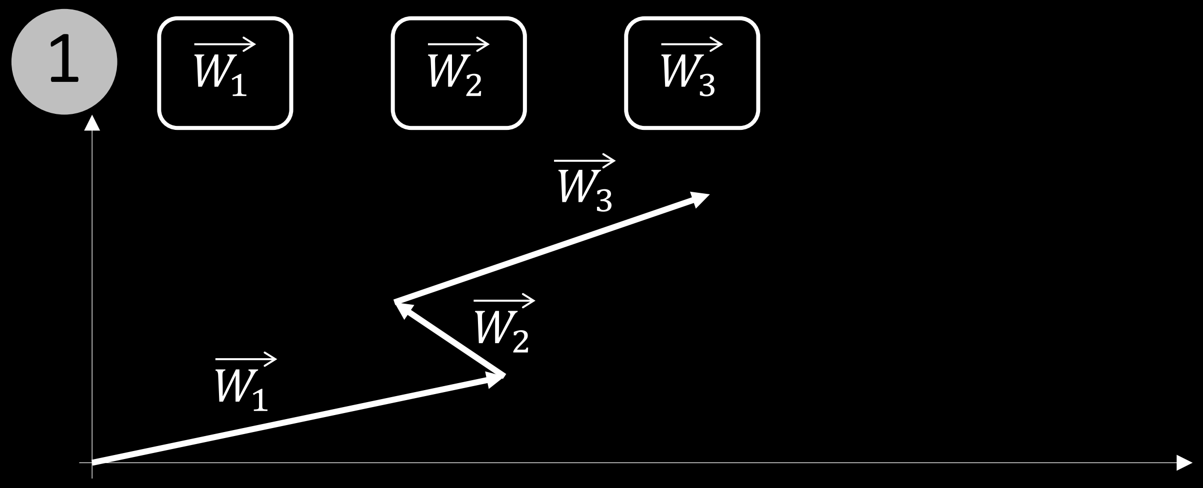
# Circular Analysis

1) Determine Backbone Updates

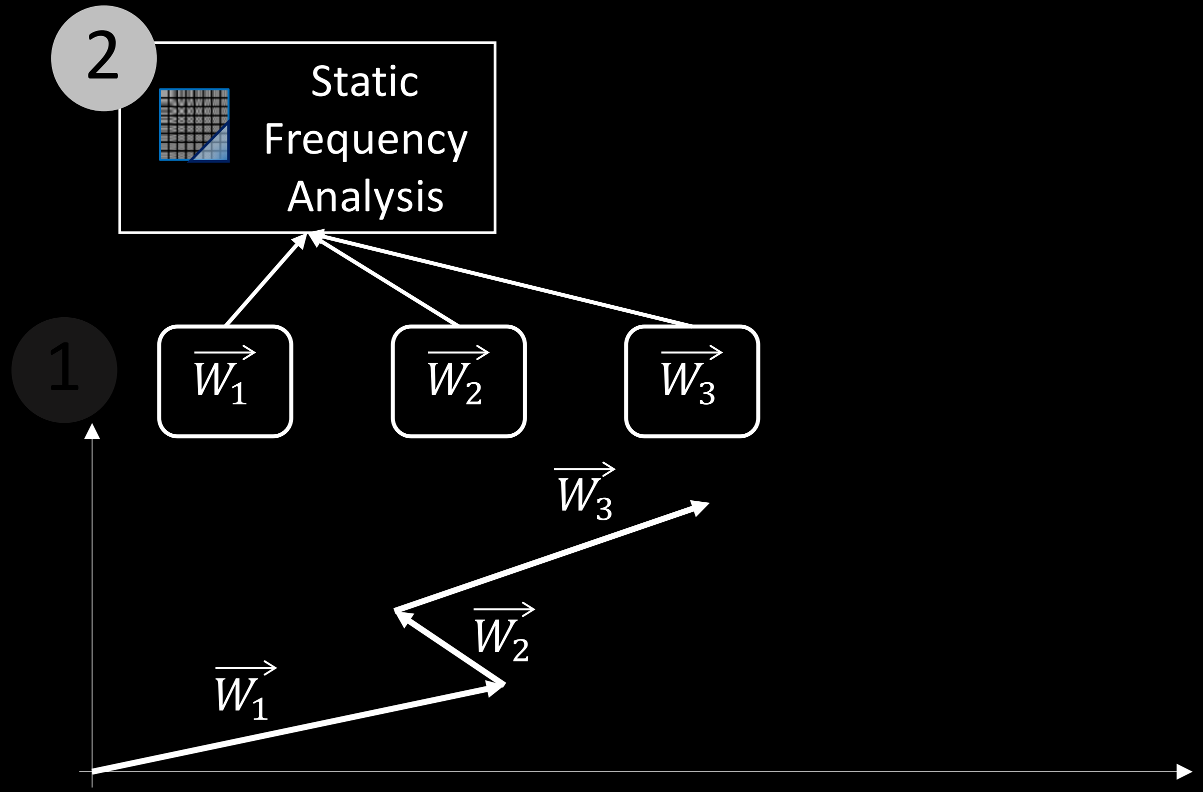


# Circular Analysis

1) Determine Backbone Updates



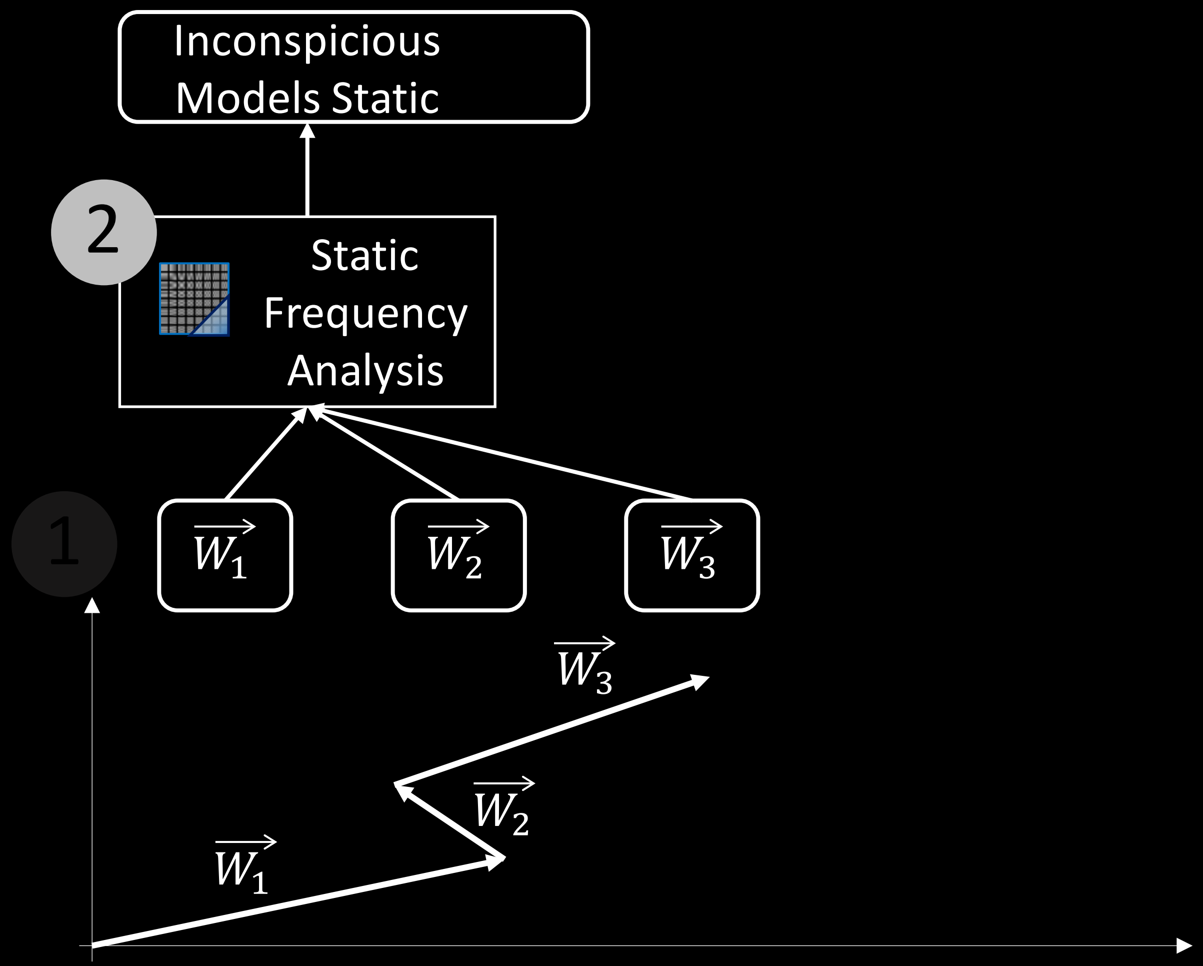
# Circular Analysis




1) Determine Backbone Updates

2) Static Frequency Analysis

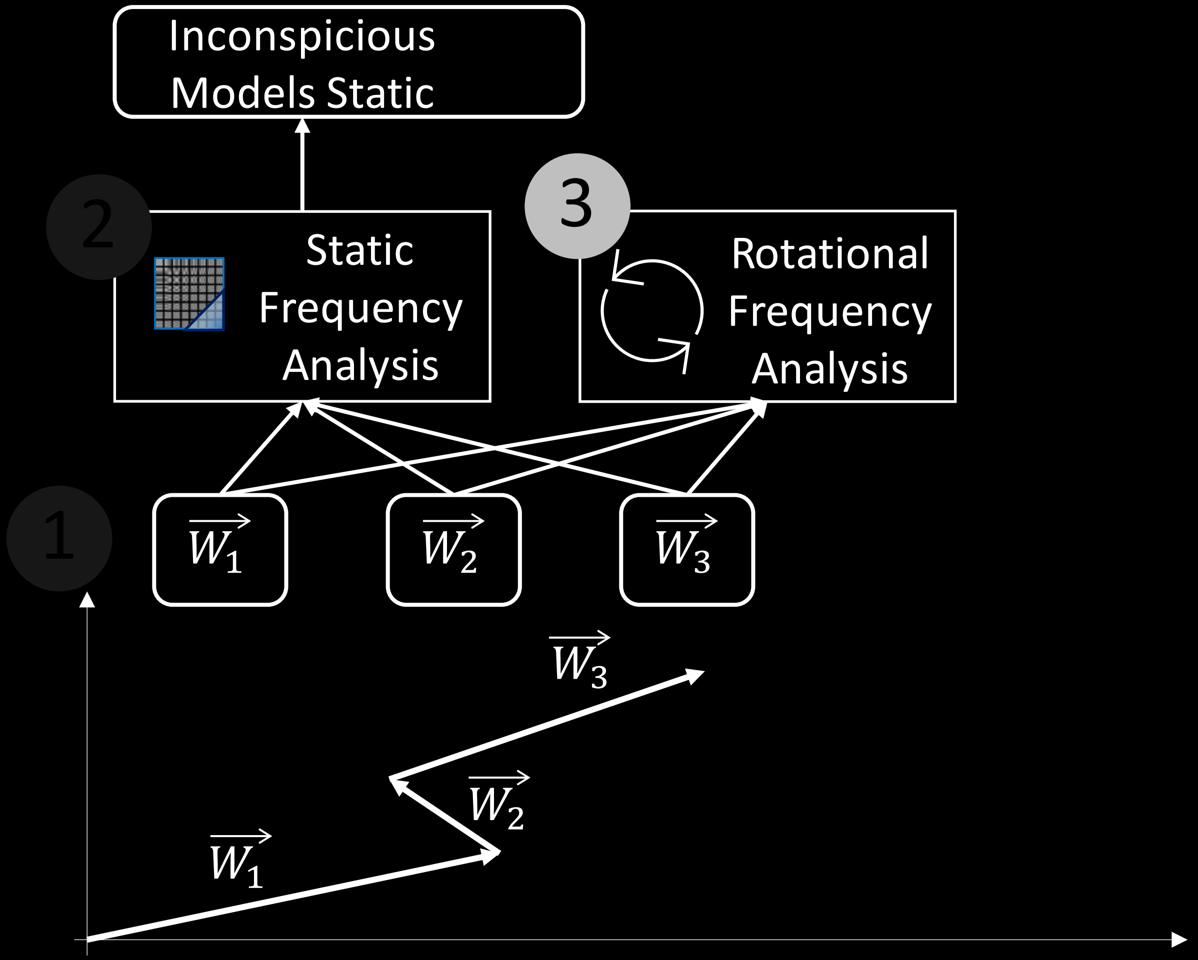
# Circular Analysis




1) Determine Backbone Updates

2) Static Frequency Analysis

# Circular Analysis

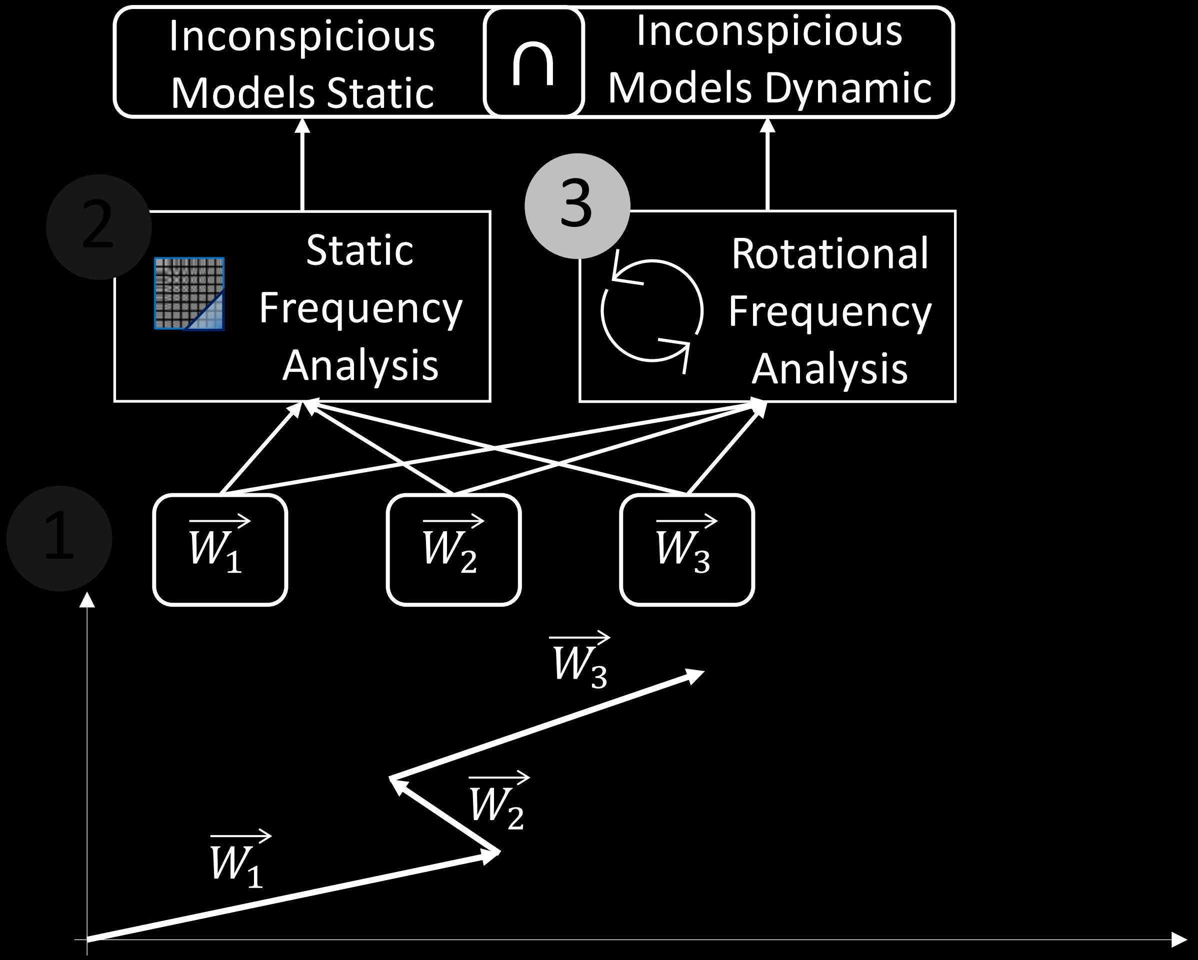



1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

# Circular Analysis

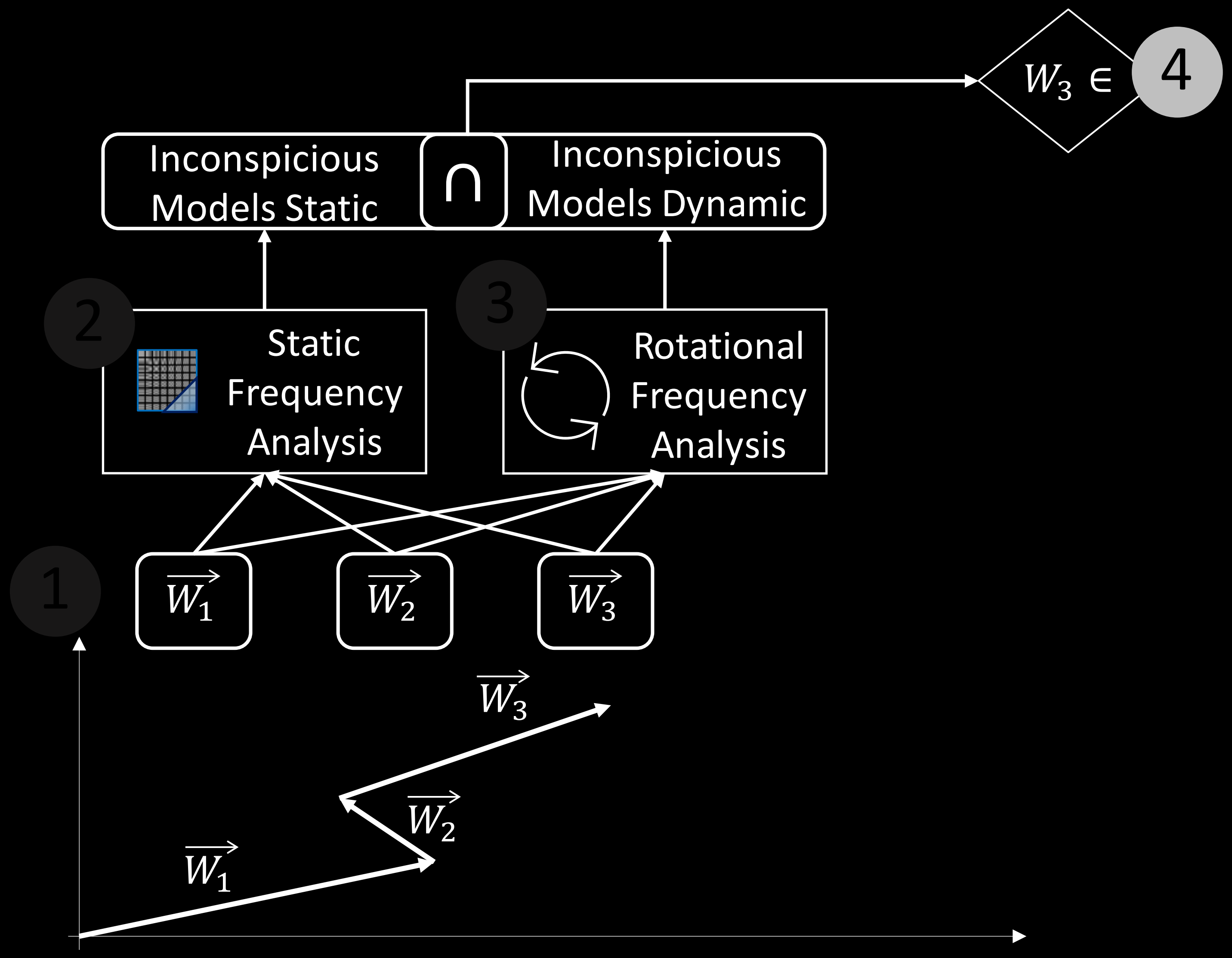



1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

# Circular Analysis



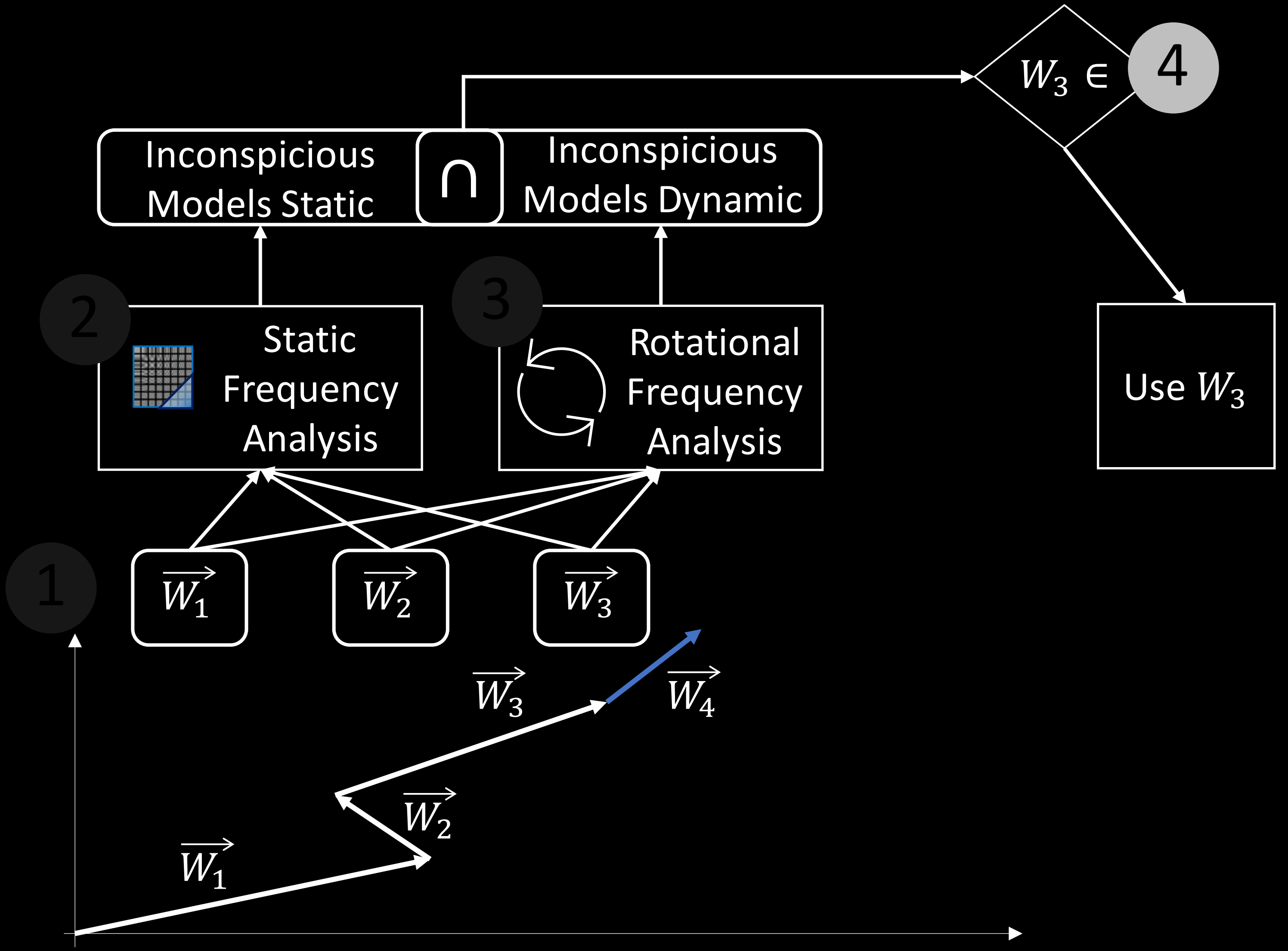

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



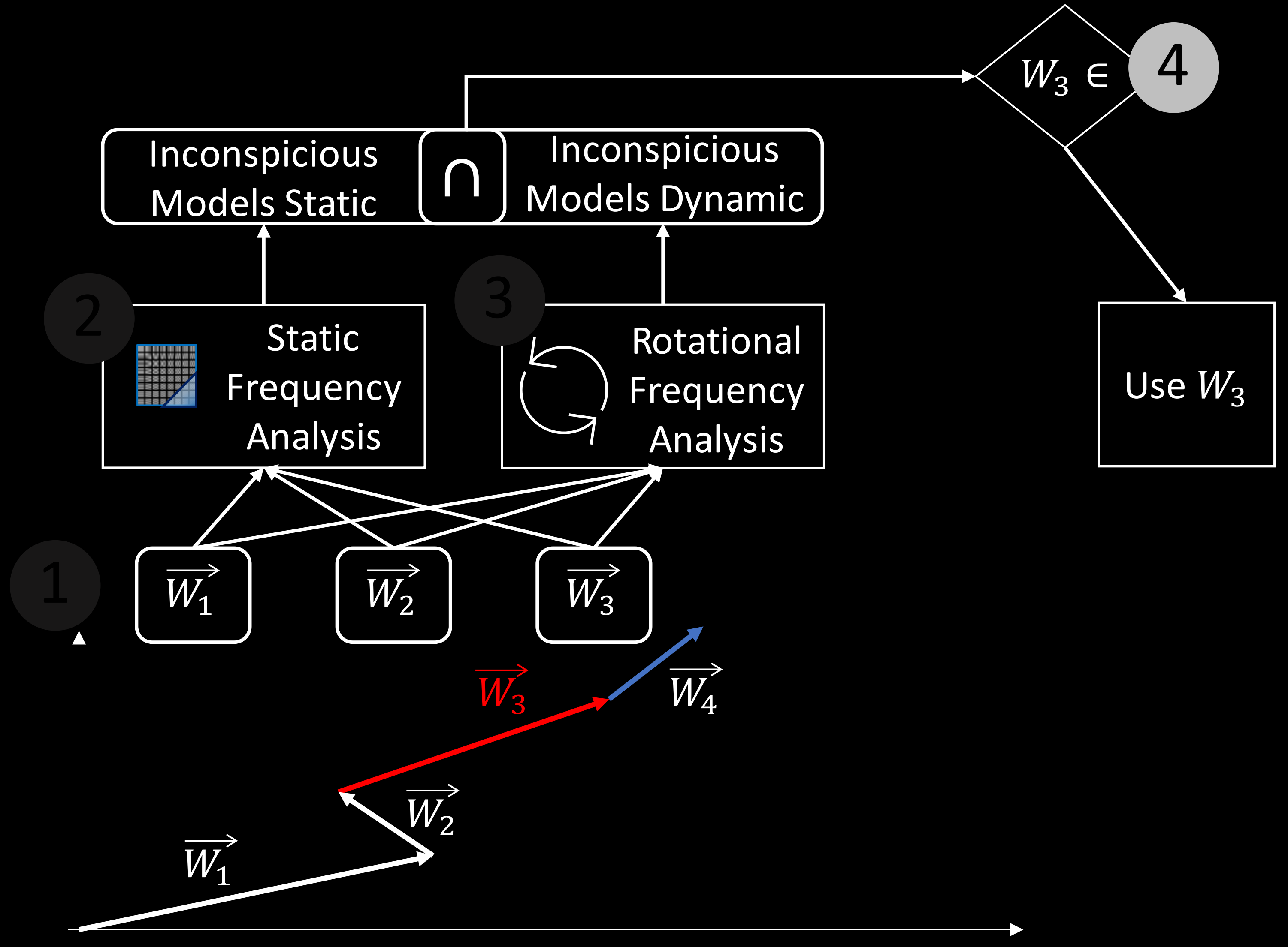

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



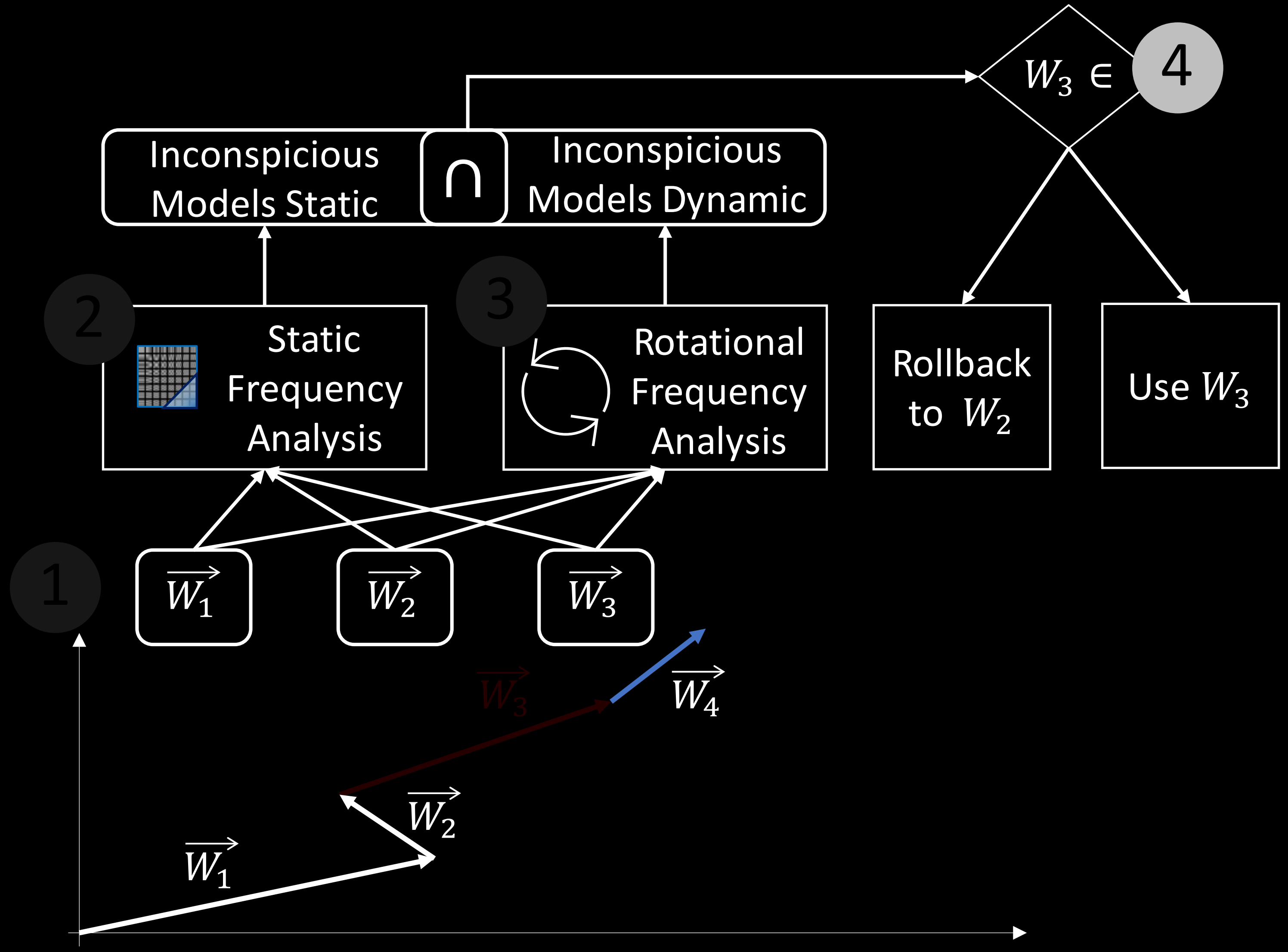

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



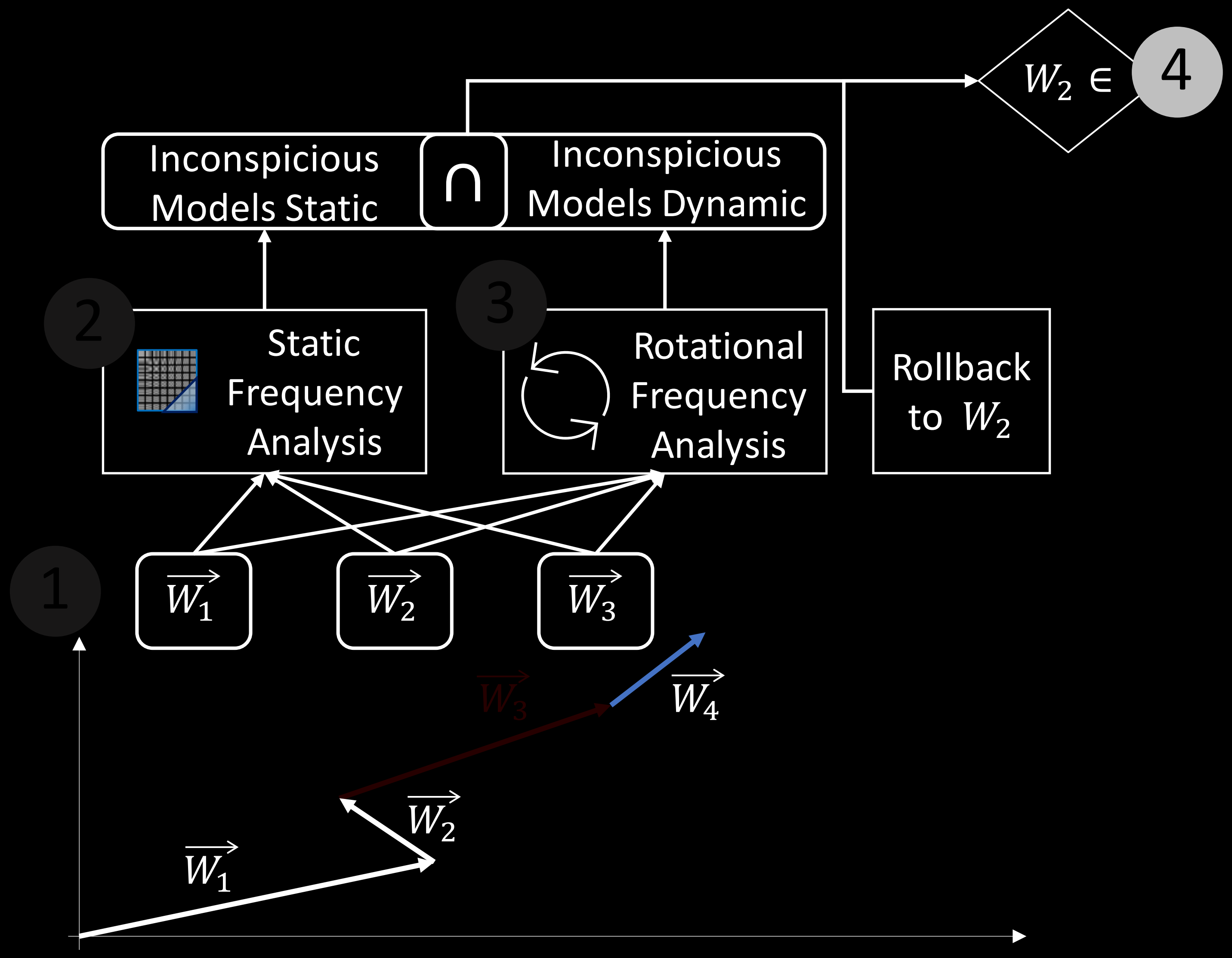

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



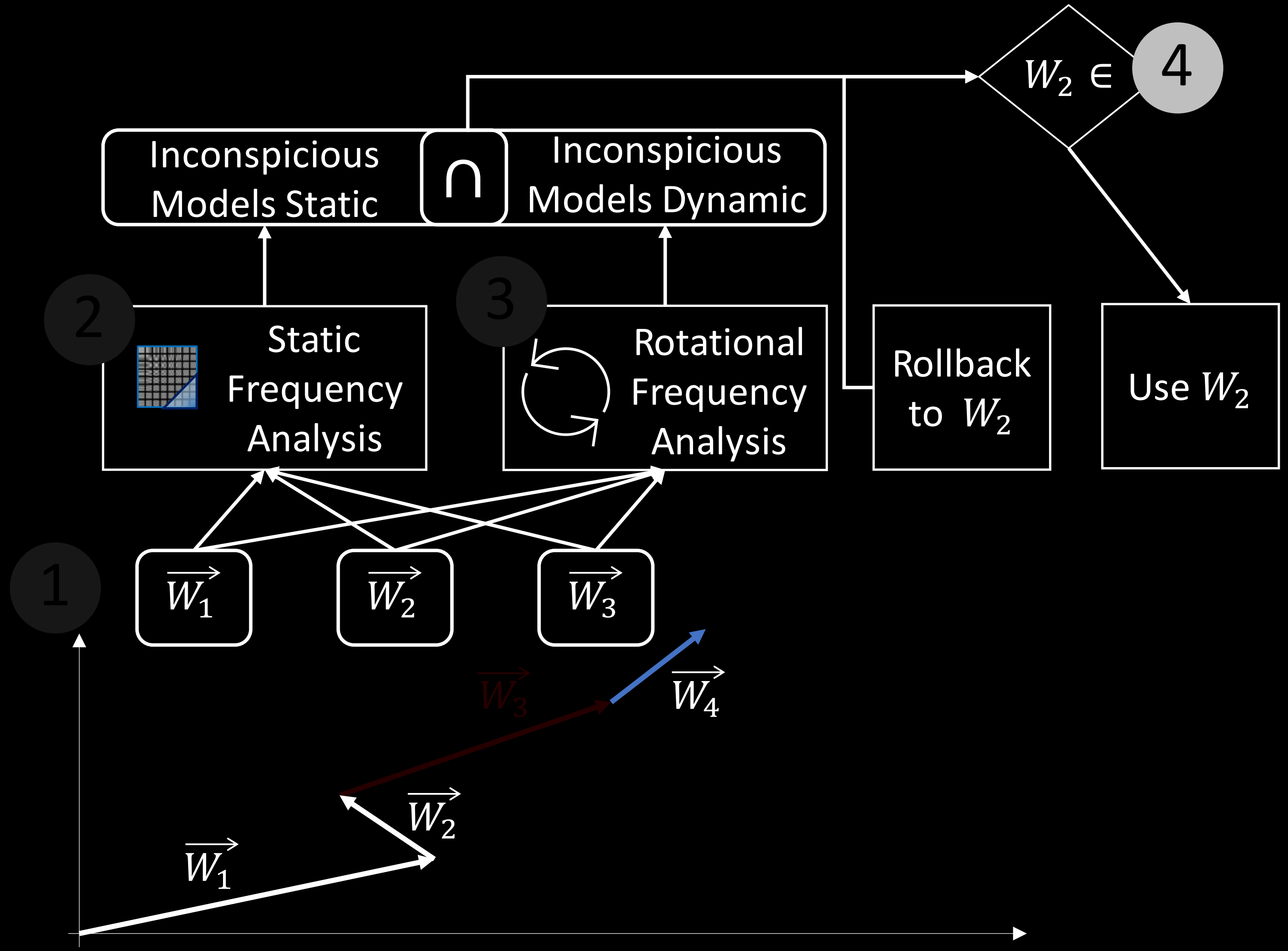

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



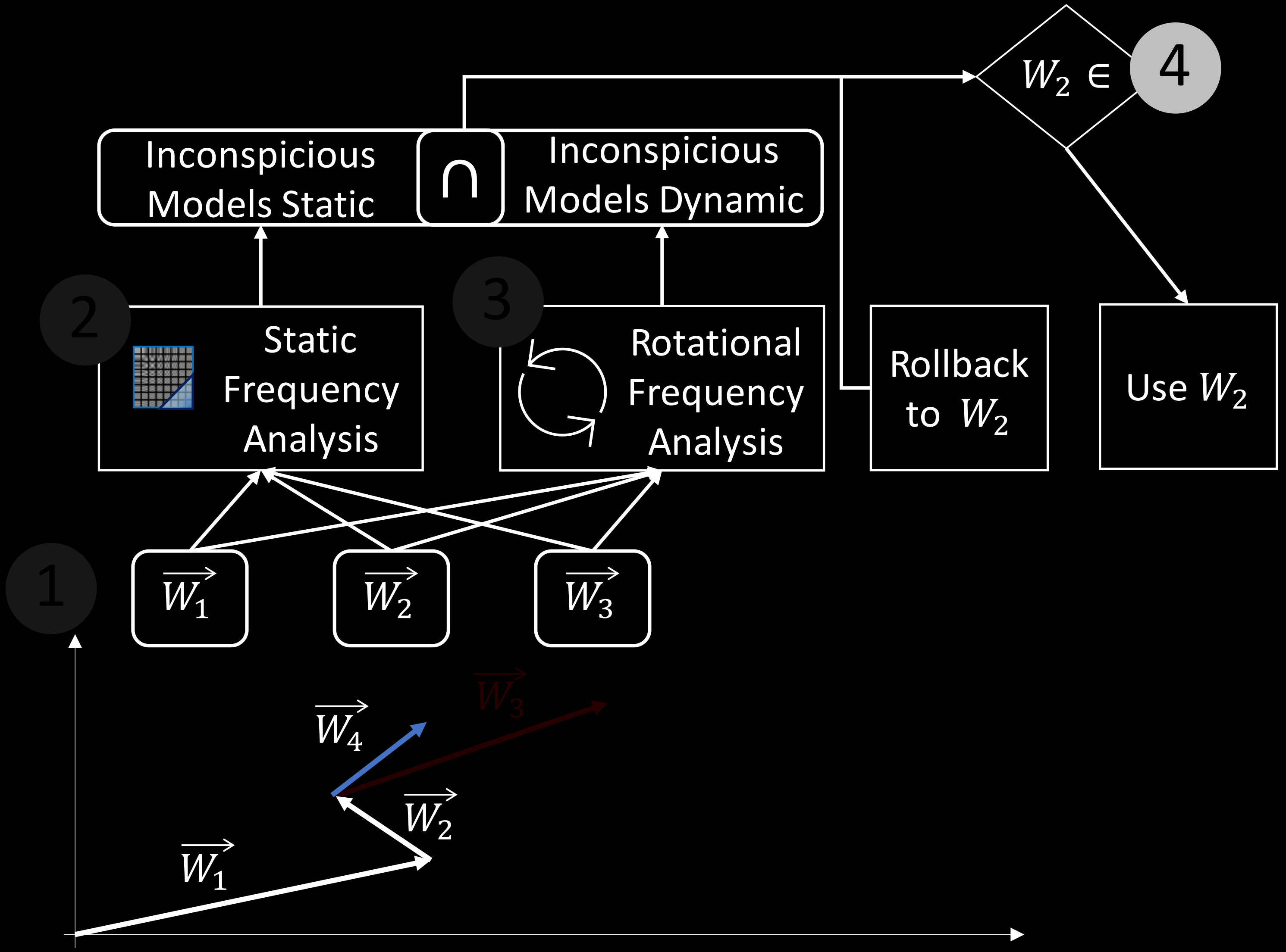

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

4) Rollback Check

# Circular Analysis



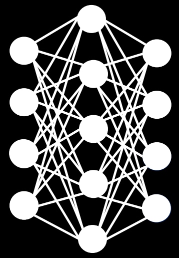

1) Determine Backbone Updates

2) Static Frequency Analysis

3) Rotational Distance Analysis

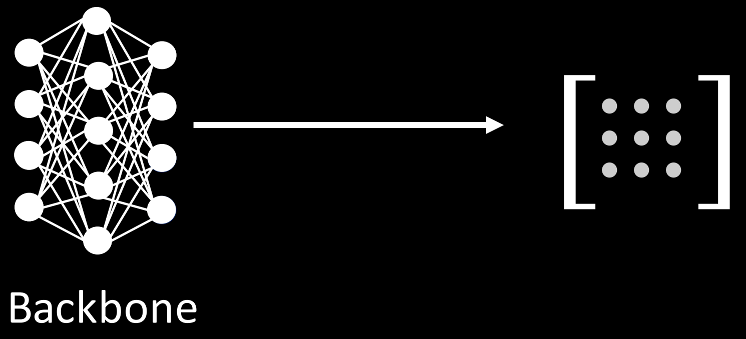
4) Rollback Check

# Dynamic Analysis – Rotational Distance

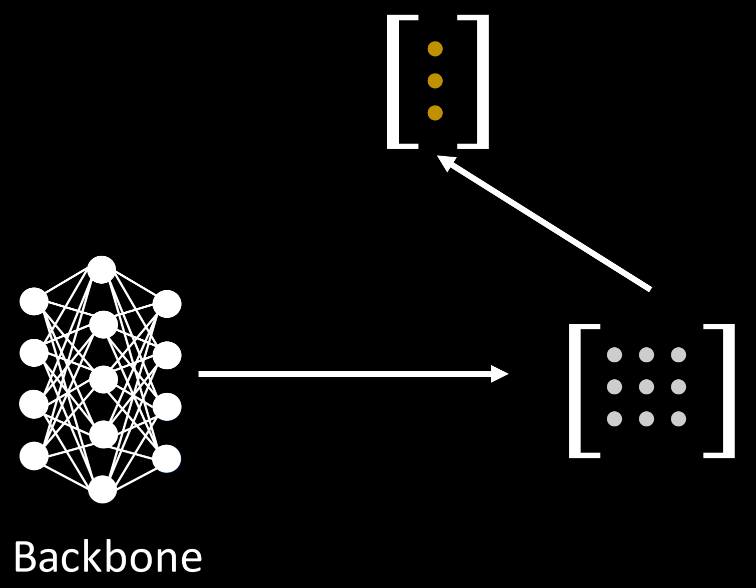


Backbone

# Dynamic Analysis – Rotational Distance

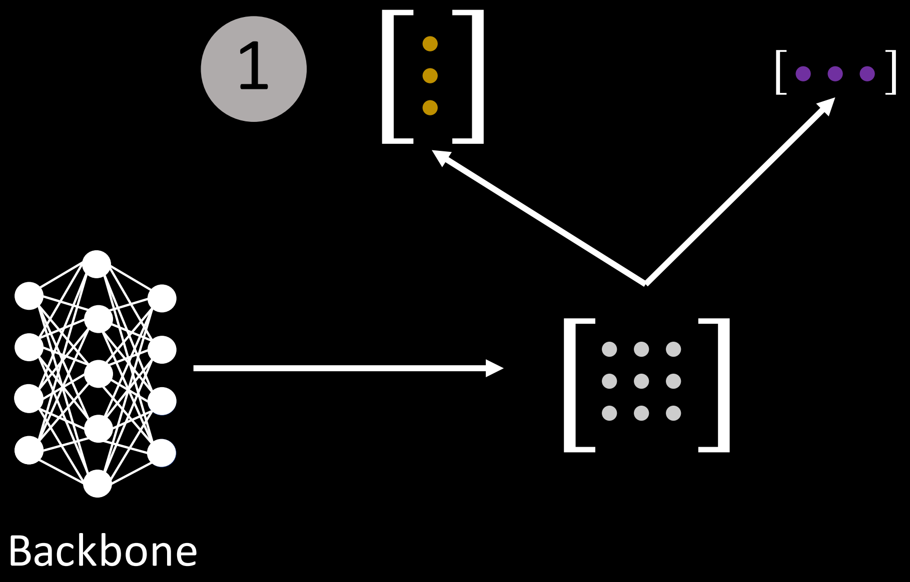


# Dynamic Analysis – Rotational Distance



# Dynamic Analysis – Rotational Distance

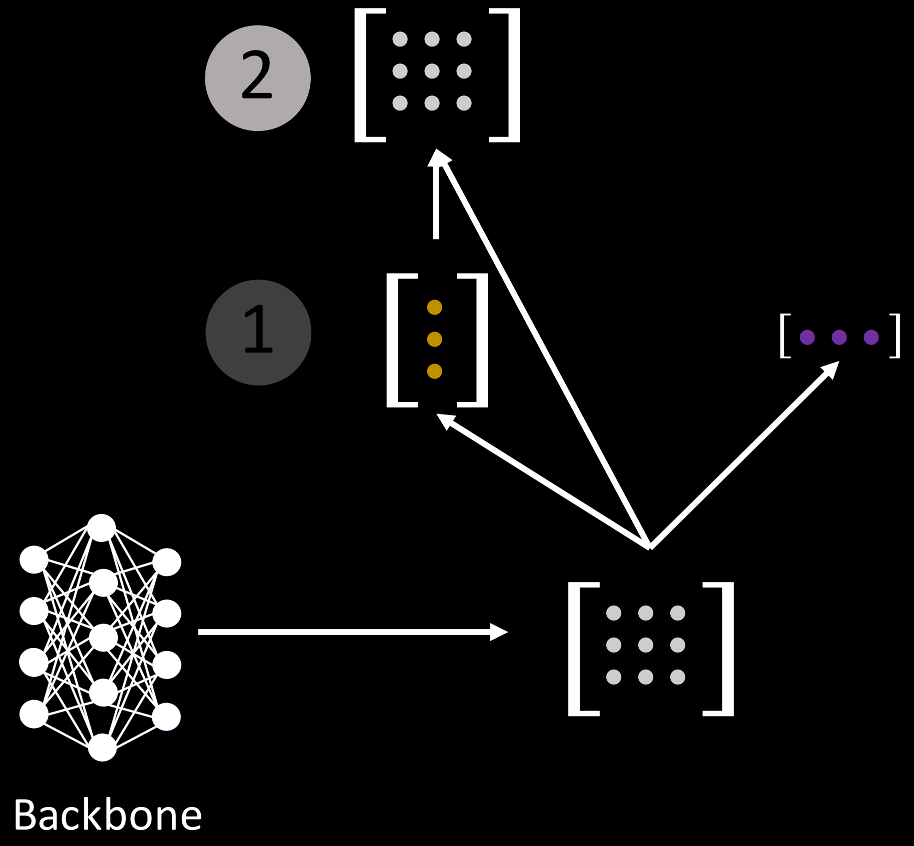
1) Calculate row- and column-mean



# Dynamic Analysis – Rotational Distance

1) Calculate row- and column-mean

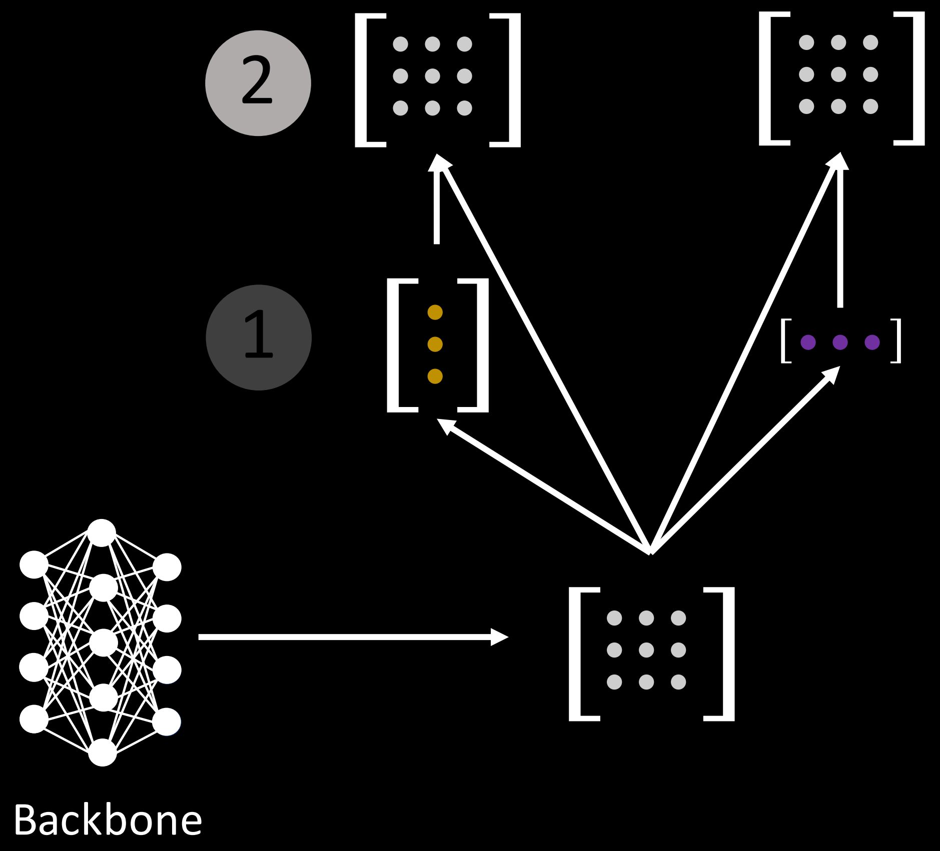
2) Multiply weights with means to obtain construction values



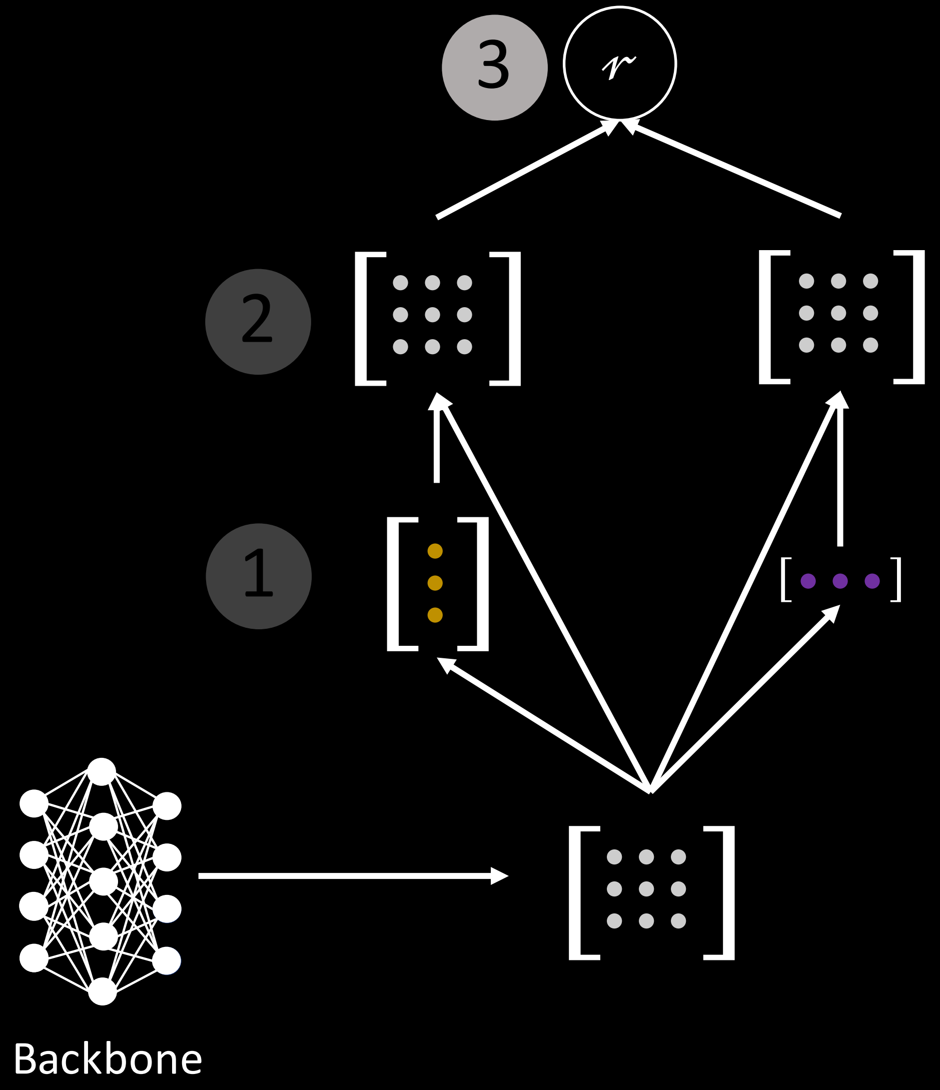
# Dynamic Analysis – Rotational Distance

1) Calculate row- and column-mean

2) Multiply weights with means to obtain construction values



# Dynamic Analysis – Rotational Distance

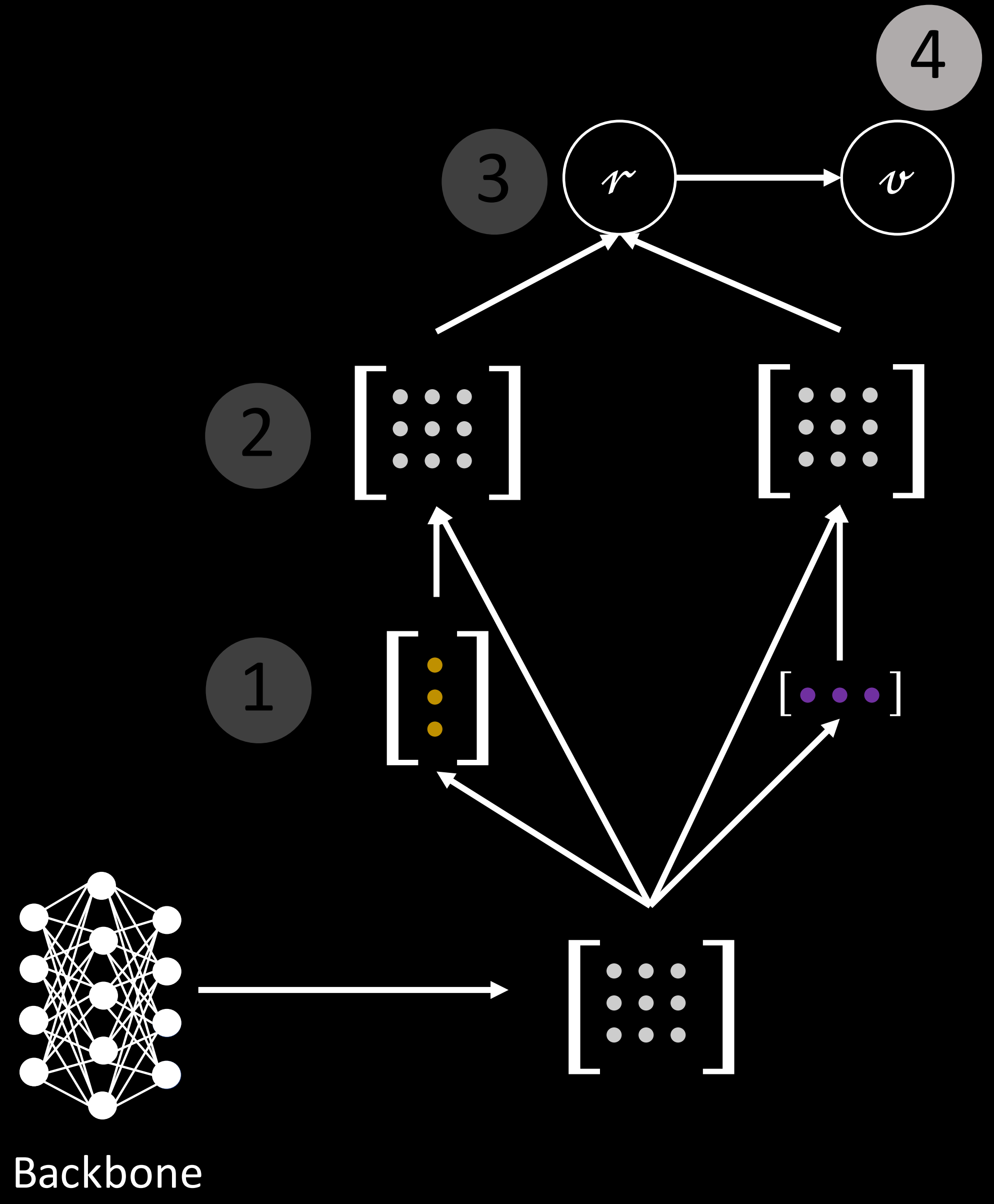



1) Calculate row- and column-mean

2) Multiply weights with means to obtain construction values

3) Calculate angular displacement $r$

# Dynamic Analysis – Rotational Distance



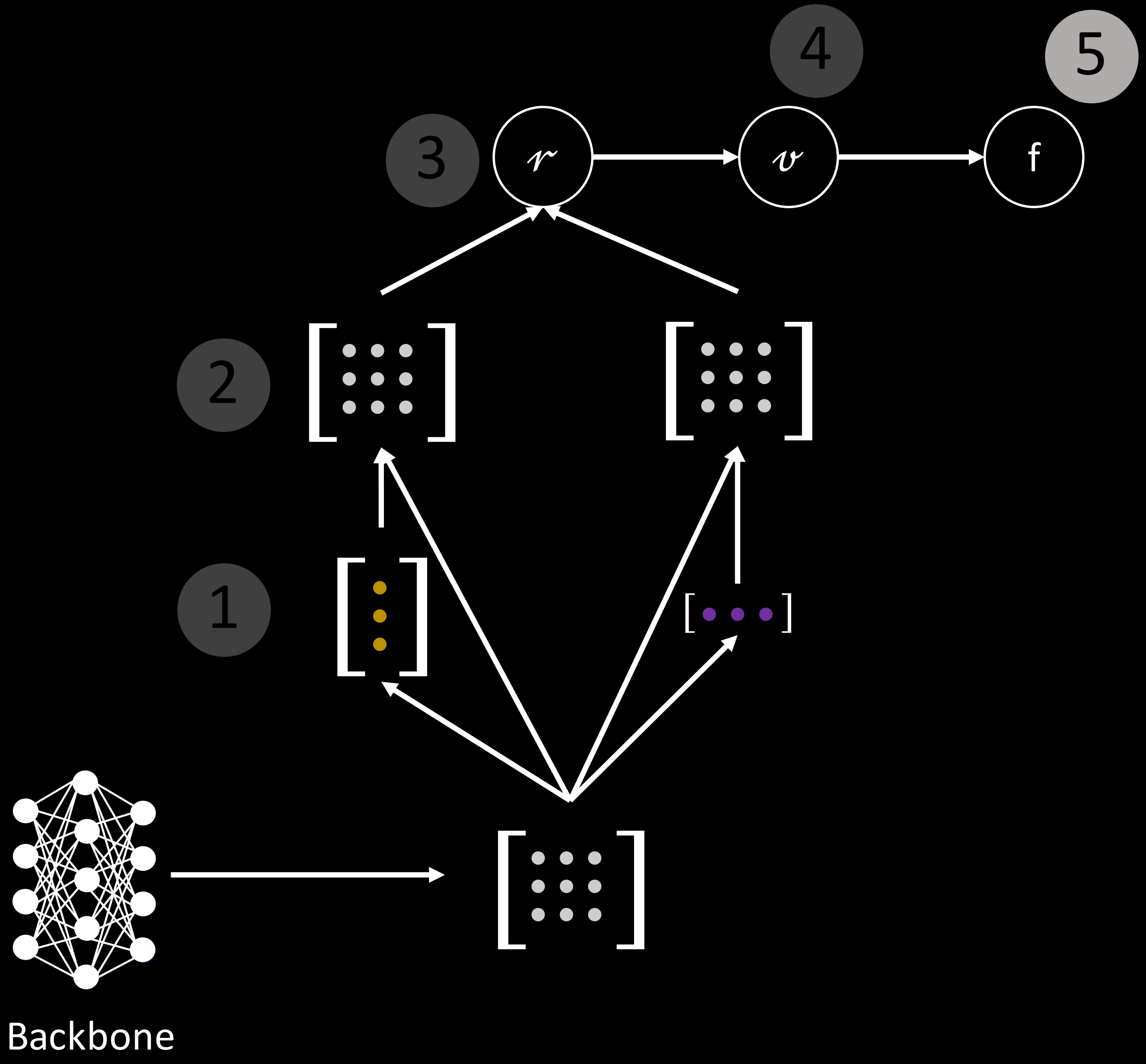

1) Calculate row- and column-mean

2) Multiply weights with means to obtain construction values

3) Calculate angular displacement $r$

4) Divide by time-skew to calculate angular velocity $v$

# Dynamic Analysis – Rotational Distance




1) Calculate row- and column-mean

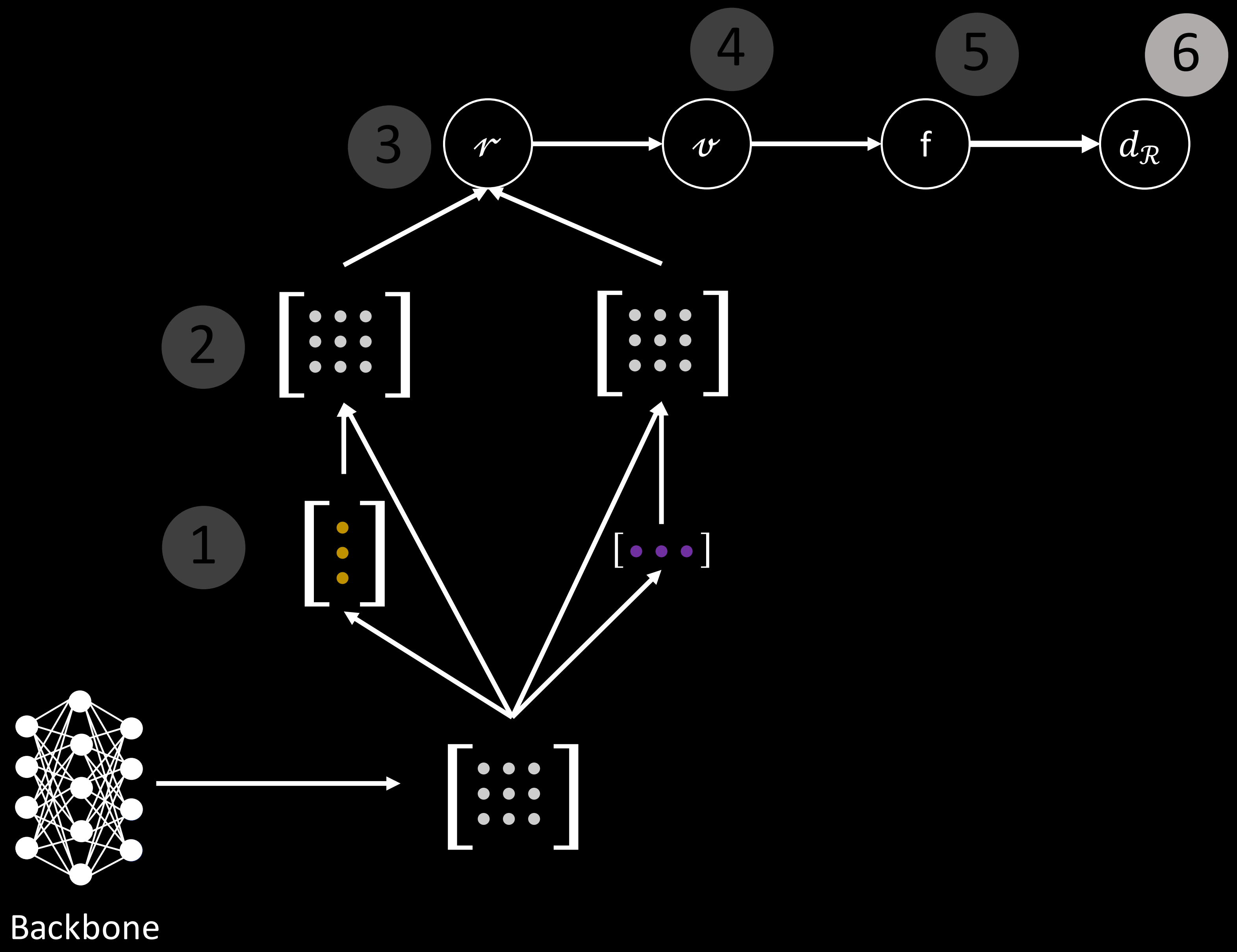
2) Multiply weights with means to obtain construction values

3) Calculate angular displacement $r$

4) Divide by time-skew to calculate angular velocity $v$

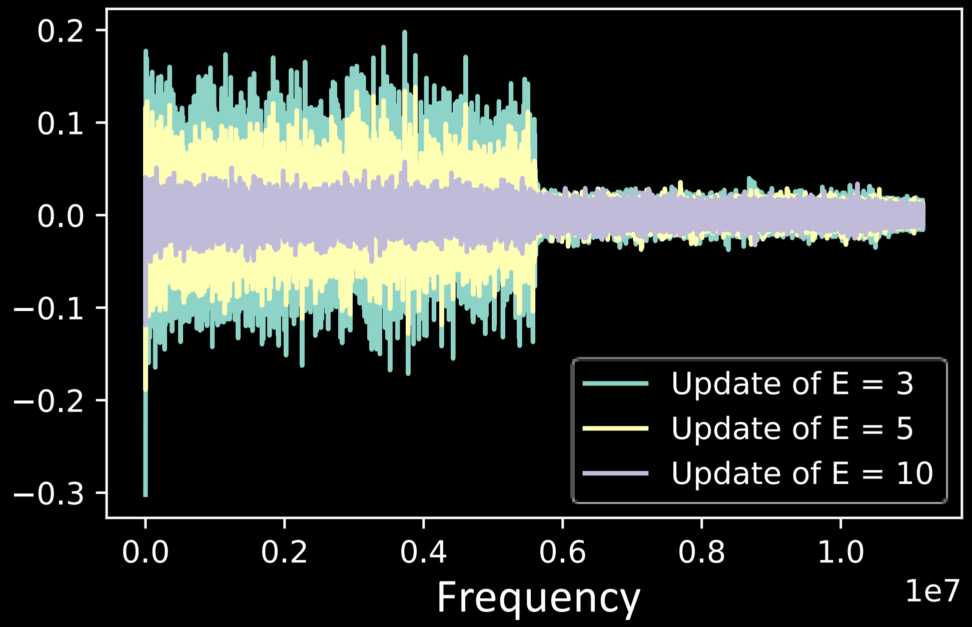
5) Determine rotational frequency

# Dynamic Analysis – Rotational Distance



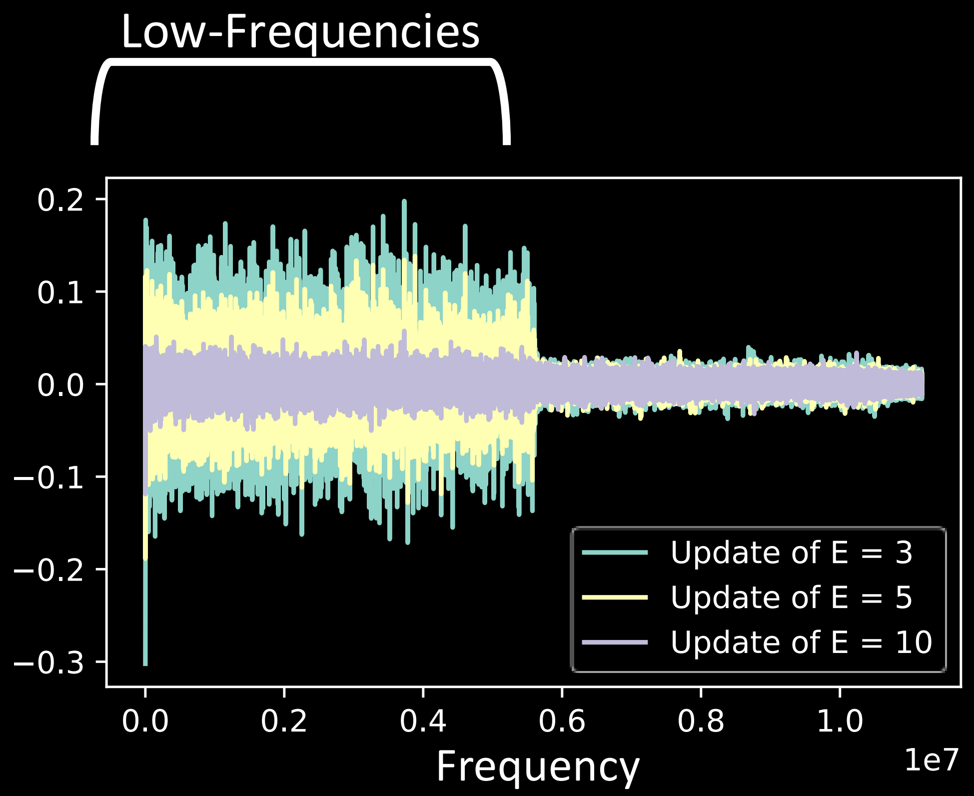

1) Calculate row- and column-mean
2) Multiply weights with means to obtain construction values
3) Calculate angular displacement $r$
4) Divide by time-skew to calculate angular velocity $v$
5) Determine rotational frequency
6) Calculate Distance $d_{\mathcal{R}}$ to Neighborhood

# Static Analysis – Frequency Domain



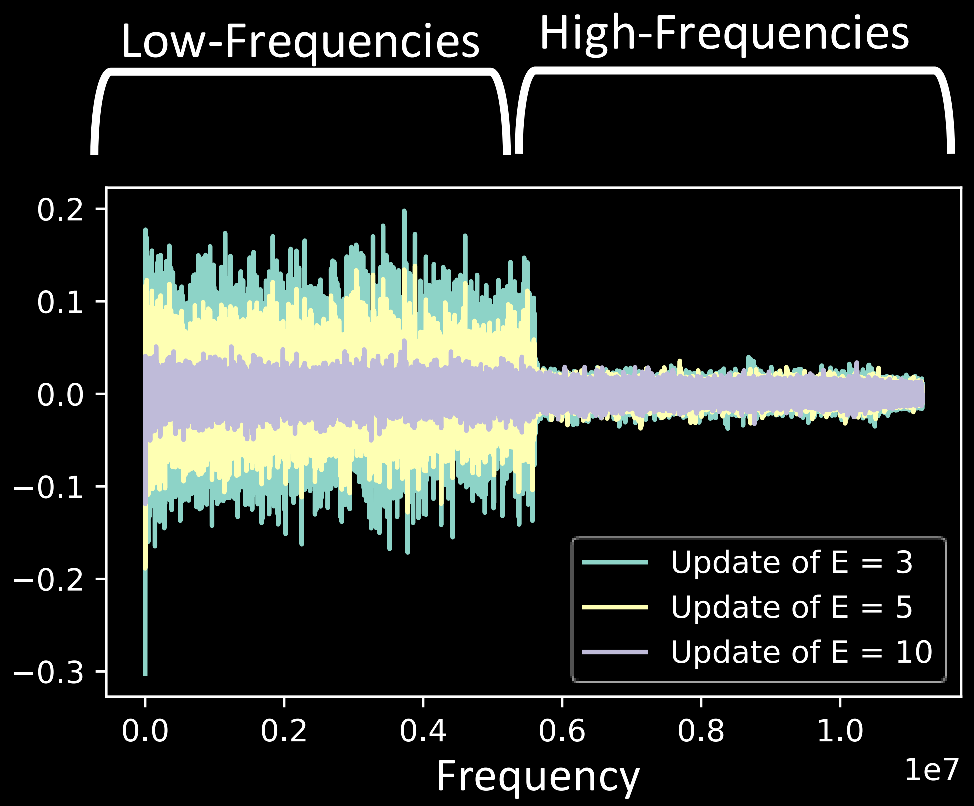
# Static Analysis – Frequency Domain

- In early training stages, low frequencies change significantly
  - Low frequencies represent main behavior
    [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]



# Static Analysis – Frequency Domain

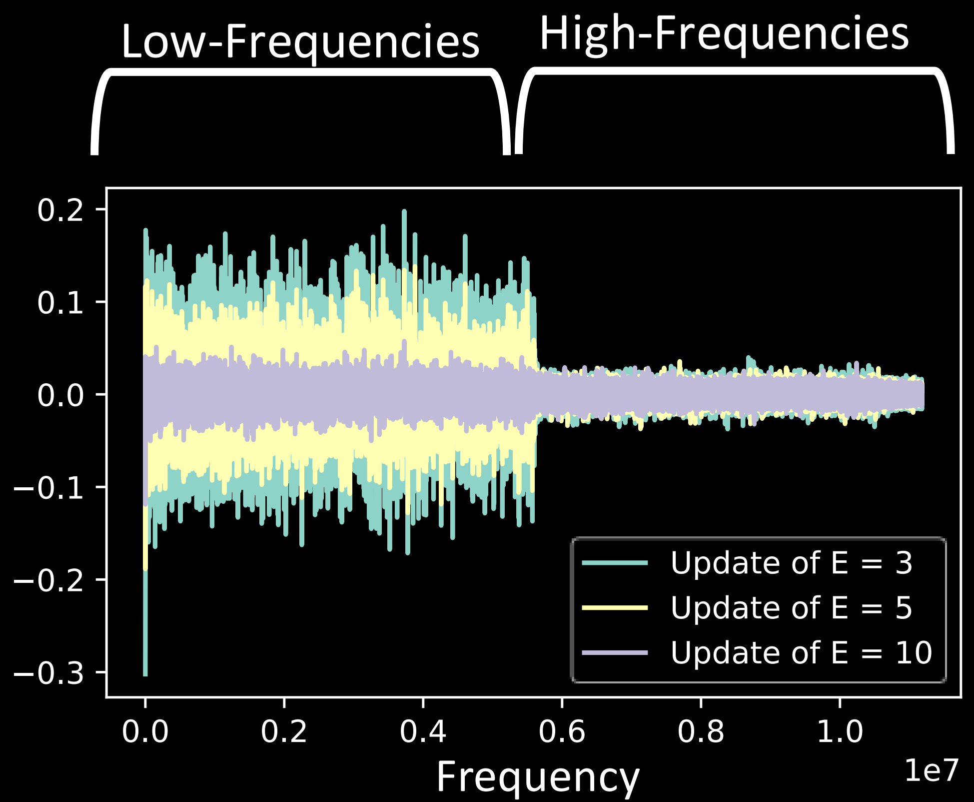
- In early training stages, low frequencies change significantly
  - Low frequencies represent main behavior
    [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]
- In later epoch ratio to high-frequencies changes
  [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]

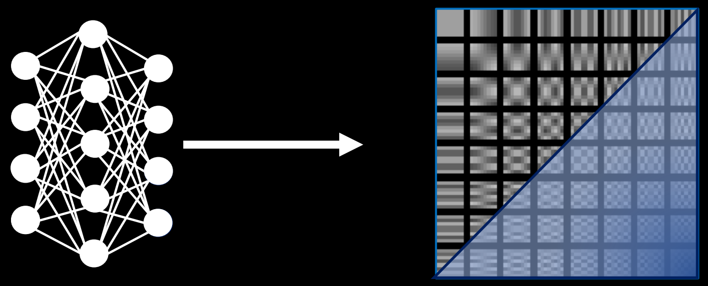


# Static Analysis – Frequency Domain

- In early training stages, low frequencies change significantly
  - Low frequencies represent main behavior
    [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]

- In later epoch ratio to high-frequencies changes
  [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]



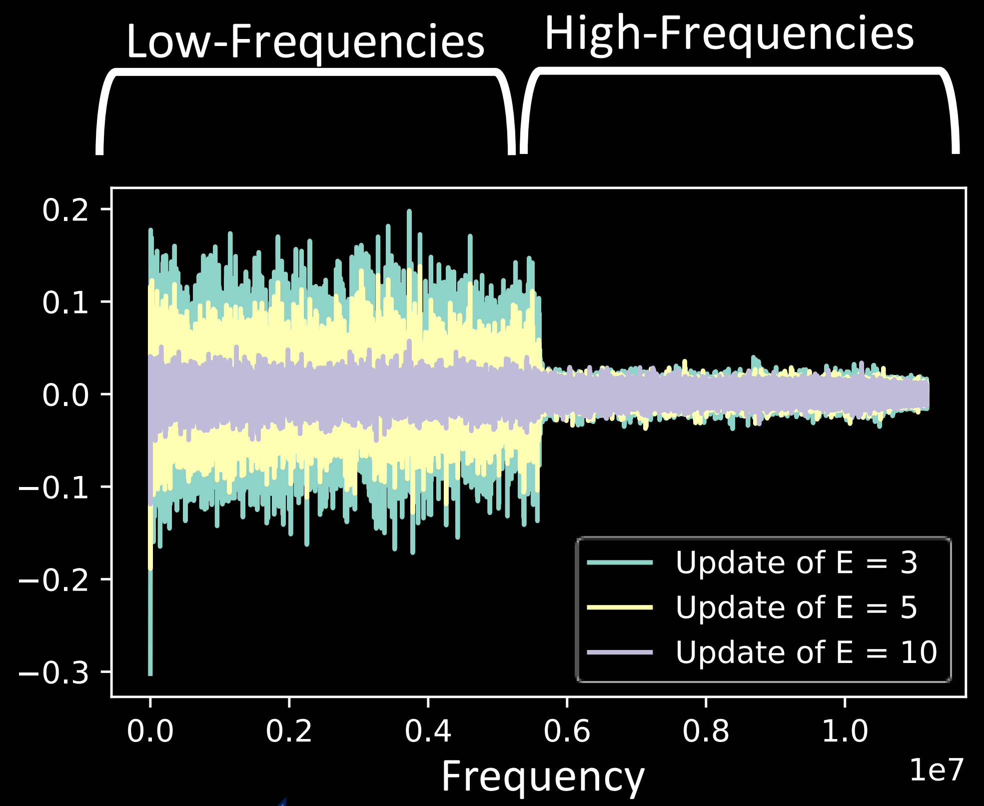



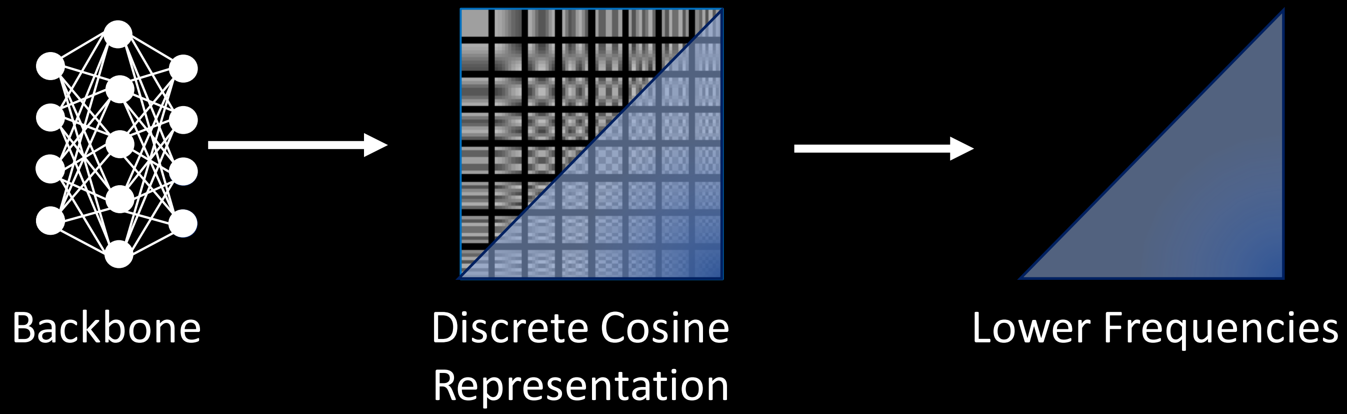
Backbone

Discrete Cosine Representation

# Static Analysis – Frequency Domain

- In early training stages, low frequencies change significantly
  - Low frequencies represent main behavior
    [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]
- In later epoch ratio to high-frequencies changes
  [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]

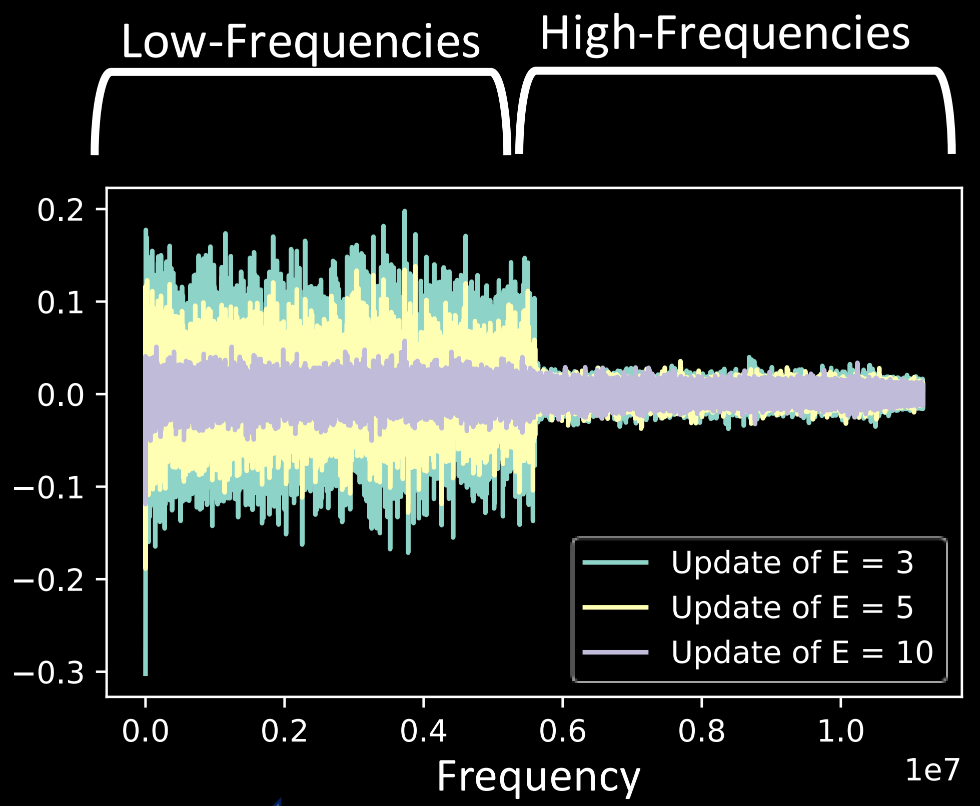


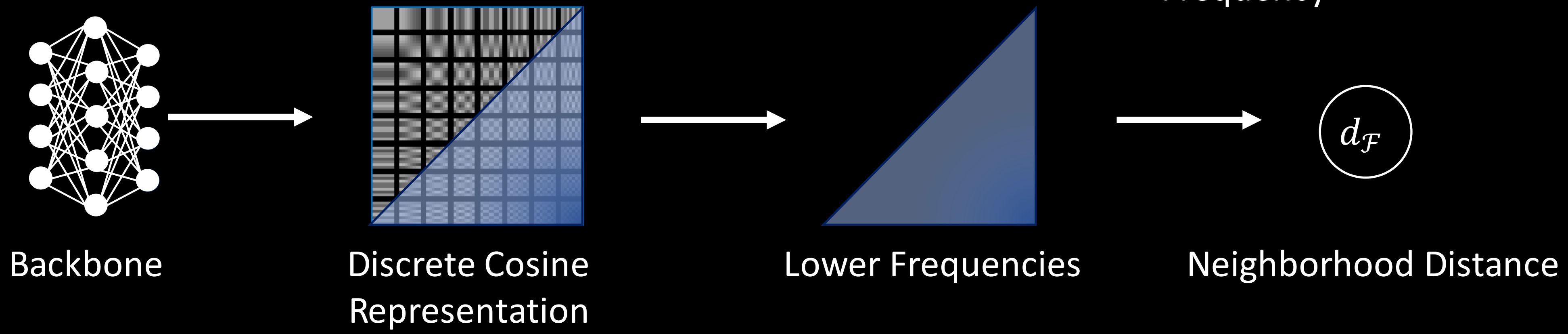



# Static Analysis – Frequency Domain

- In early training stages, low frequencies change significantly
  - Low frequencies represent main behavior
    [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]

- In later epoch ratio to high-frequencies changes
  [Rahamanet al. ICML 2019], [Xu et al. ICONIP 2019]



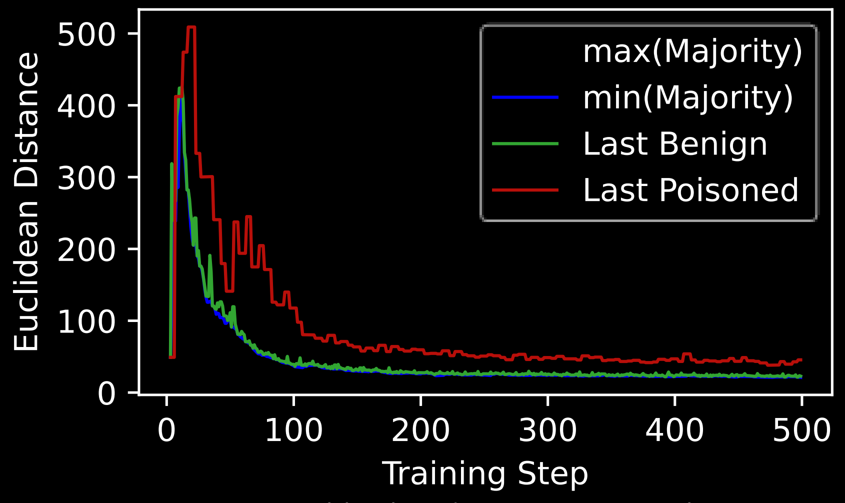
# Evaluation Results

BA: Backdoor Accuracy
MA: Main Task Accuracy

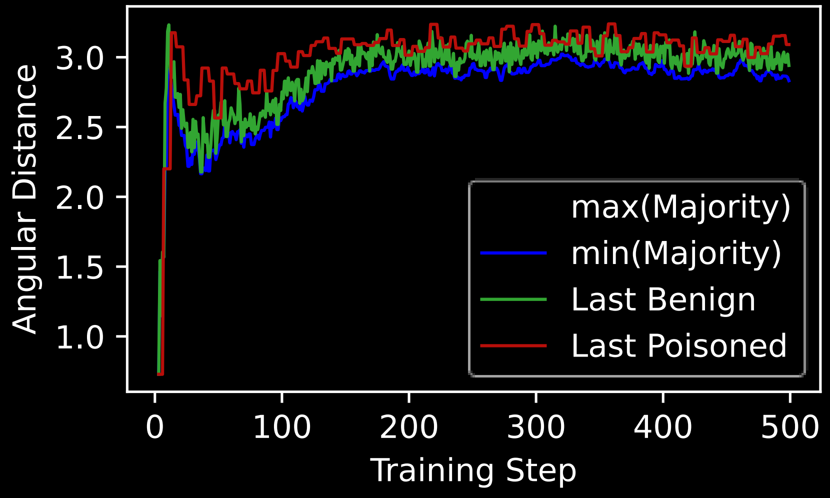
| | No Defense | | Safe Split | |
|---|---|---|---|---|
| | BA | MA | BA | MA |
| Cifar-10 | 59.3% | 66.6% | 0.0% | 62.7% |
| CIFAR-100 | 93.3% | 76.8% | 0.1% | 76.5% |
| MNIST | 86.2% | 98.7% | 0.0% | 98.8% |
| FMNIST | 79.8% | 83.0% | 3.4% | 84.6% |
| GTSRB | 30.0% | 58.0% | 0.6% | 63.7% |

| | No Defense | | Safe Split | |
|---|---|---|---|---|
| | BA | MA | BA | MA |
| ResNet-18 | 59.3% | 66.6% | 0.0% | 62.7% |
| CNN | 78.0% | 62.7% | 0.0% | 60.4% |
| GoogLeNet | 16.7% | 57.9% | 0.0% | 60.2% |
| VGG11 | 76.7% | 49.7% | 0.0% | 43.0% |

Different DNN Architectures for Cifar-10

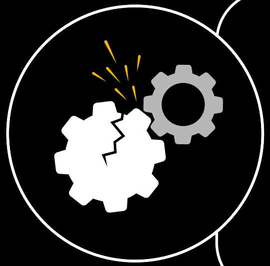


Neighborhood Scores Static Analysis



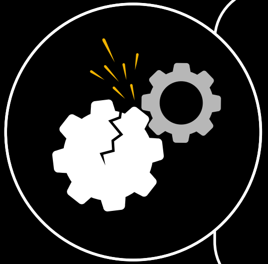
Neighborhood Scores Dynamic Analysis

# Conclusion



- Split learning allows computation expensive training and inference on mobile devices
- However suspectable for backdoor attacks
- Existing defenses are insufficient against due to sequential training

# Conclusion



- Split learning allows computation expensive training and inference on mobile devices
- However suspectable for backdoor attacks
- Existing defenses are insufficient against due to sequential training

- Circle architecture allows mitigating impact of backdoor attacks
- Combines static and dynamic perspective for model analysis
- Rotational distance metric provides detailed insights in training dynamics

# Conclusion

- Split learning allows computation expensive training and inference on mobile devices
- However suspectable for backdoor attacks
- Existing defenses are insufficient against due to sequential training

- Circle architecture allows mitigating impact of backdoor attacks
- Combines static and dynamic perspective for model analysis
- Rotational distance metric provides detailed insights in training dynamics

- Reduce attack success rate below 5%
- Ensemble of different perspectives makes system robust against adaptive attacks
- Circle-wise architecture allows rollback without retraining benign models